

T.C.
BAŐKENT ÜNİVERSİTESİ
EĐİTİM BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR VE ÖĐRETİM TEKNOLOJİLERİ EĐİTİMİ
ANABİLİM DALI
BİLGİSAYAR VE ÖĐRETİM TEKNOLOJİLERİ EĐİTİMİ
TEZLİ YÜKSEK LİSANS PROGRAMI

BLOK TABANLI ORTAMLARDA PROGRAMLAMA
ÖĐRETİMİ SÜRECİNDE FARKLI ÖĐRETİM
STRATEJİLERİNİN ÇEŐİTLİ DEĐİŐKENLER AÇISINDAN
İNCELENMESİ

HAZIRLAYAN
ECMEN ERDEM

TEZ DANIŐMANI
Doç. Dr. FİLİZ KALELİOĐLU

YÜKSEK LİSANS TEZİ
MAYIS 2018

T.C.
BAŐKENT ÜNİVERSİTESİ
EĐİTİM BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR VE ÖĐRETİM TEKNOLOJİLERİ EĐİTİMİ
ANABİLİM DALI
BİLGİSAYAR VE ÖĐRETİM TEKNOLOJİLERİ EĐİTİMİ
TEZLİ YÜKSEK LİSANS PROGRAMI

BLOK TABANLI ORTAMLARDA PROGRAMLAMA
ÖĐRETİMİ SÜRECİNDE FARKLI ÖĐRETİM
STRATEJİLERİNİN ÇEŐİTLİ DEĐİŐKENLER AÇISINDAN
İNCELENMESİ

HAZIRLAYAN

Ecmen ERDEM

TEZ DANIŐMANI

Doç. Dr. Filiz KALELİOĐLU

Yüksek Lisans Tezi

Mayıs 2018

T.C.
BAŞKENT ÜNİVERSİTESİ
EĞİTİM BİLİMLERİ ENSTİTÜSÜ

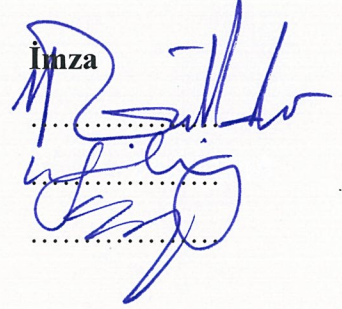
BLOK TABANLI ORTAMLARDA PROGRAMLAMA ÖĞRETİMİ
SÜRECİNDE FARKLI ÖĞRETİM STRATEJİLERİNİN ÇEŞİTLİ
DEĞİŞKENLER AÇISINDAN İNCELENMESİ

Ecmen ERDEM
Yüksek Lisans Tezi

Bu tez 15/05/2018 tarihinde aşağıda üye adları yazılı jüri tarafından kabul edilmiştir.

Ünvan	Adı Soyadı
Prof. Dr.	Yasemin GÜLBAHAR GÜVEN
Doç. Dr.	Filiz KALELİOĞLU
Dr. Öğr. Üyesi	Serpil YALÇINALP

İmza



Onay

/ / 2018

Eğitim Bilimleri Enstitü Müdürü

.....

ETİK

Tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmada,

- Tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- Görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- Başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- Atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- Kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- Ve bu tezin herhangi bir bölümünü bu üniversitede veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

15.05.2018

Ecmen ERDEM

Teşekkürler

Bu süreçte bana her konuda destek olan, hazırladığım tez konusundaki yönlendirmeleri ile benim ufkumu açan, üzerimde büyük emekleri olan değerli tez danışmanım Doç.Dr. Filiz Kalelioğlu'na,

Lisansta ve yüksek lisans dönemimde kendisinden ders aldığım değerli hocam ve tez jürisi olan Yrd. Doç. Dr. Serpil Yalçınalp'e,

Lisans döneminde kendisinden ders alma şerefine nail olup 10 yıl sonra tekrar kendisine sunum yapma fırsatı bulduğum alanımızın gelişmesinde büyük emekleri olan değerli hocam ve tez jüri başkanı olan Prof. Dr. Yasemin Gülbahar'a,

Hayatımda verdiği fikirler ile birçok konuda bana yol gösteren, gerçek bir yol arkadaşı olan değerli dostum Turgay Öntaş'a,

Araştırmamı sahasında yapmama olanak sağlayan okuluma ve değerli yöneticilerine,

Verdikleri motivasyon ile bu tezi bitirmemi sağlayan anneme, babama, kardeşime, sevgili eşime ve biricik oğluma,

Sevgi, saygı ve teşekkürlerimi sunarım...

ÖZ

BLOK TABANLI ORTAMLARDA PROGRAMLAMA ÖĞRETİMİ SÜRECİNDE FARKLI ÖĞRETİM STRATEJİLERİNİN ÇEŞİTLİ DEĞİŞKENLER AÇISINDAN İNCELENMESİ

ECMEN ERDEM

BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ EĞİTİMİ ANABİLİM DALI

YÜKSEK LİSANS TEZİ

MAYIS 2018

Bu çalışmanın amacı 5. sınıf öğrencilerinin yüz yüze eğitim ile ters yüz sınıf modeli olan teknoloji destekli öğrenme ortamlarında olmak üzere iki farklı öğrenme ve öğretme stratejisi ile Scratch programlamayı öğrenmelerinin ve programlama öğretiminin, öğrencilerin bilgi işlemsel düşünme becerilerine etkisini araştırmaktır. Çalışmaya özel bir okuldan 79 5. sınıf öğrencisi katılmıştır. Hem nicel hem nitel yöntemleri içeren karma araştırma yöntemlerinden olan “Açıklayıcı Sıralı Karma Yöntem” kullanılmıştır. Yarı deneysel ön test-son test eşleştirilmiş kontrol gruplu çalışmada, Bilgi İşlemsel Düşünme Becerilerine Yönelik Öz Yeterlik Algısı Değerlendirme Ölçeği ve Bilge Kunduz Uluslararası Enformatik ve Bilgi İşlemsel Düşünme Soruları ön test ve son test olarak uygulanmıştır. Uygulama sonrası 24 öğrenci ile odak grup görüşmesi gerçekleştirilmiştir. Uygulama süresince her hafta anlatılan kazanımlar ile ilgili haftalık başarı testleri uygulanmıştır. Ayrıca uygulama sırasında öğrencilerin yazdıkları programlar, Dr. Scratch ile analiz edilip değerlendirilmiştir. Araştırma sonuçlarına göre yüz yüze eğitim ve ters yüz sınıf modeli ile eğitimin öğrencilerin Scratch programlama öğrenmelerinde ve bilgi işlemsel düşünme becerilerinde istatistiksel olarak anlamlı olmayan bir fark oluşturduğu görülmüştür. Araştırmada kullanılan Bilgi İşlemsel Düşünme Becerilerine Yönelik Öz Yeterlik Algısı Değerlendirme Ölçeği ön test ve son test puanlarına bakıldığında gruplar arasında bilgi işlemsel düşünme becerileri öz yeterlik algıları arasında anlamlı bir farklılık oluşmaz iken gruplar içi ön-test ve son-test puanları arasında anlamlı bir farklılık ortaya çıkmıştır. Özellikle algoritma geliştirme ve problem çözme yeterlik puanları ön-test ve son-test puanları arasında farkın anlamlı çıkması, programlama öğretiminin bu yeterliklerin artmasında önemli bir etken olduğunu söylenebilir. Ayrıca programlama öğretiminin diğer becerilere ilişkin öz yeterliklerine olumlu etkisi olduğunu söylenebilir. Haftalık başarı testlerinde bazı haftalarda deney grubunun ortalaması bazı haftalarda kontrol grubunun ortalaması daha yüksek çıkmıştır. Aynı şekilde Dr. Scratch ile analiz edilen programların ortalama puanlarına bakıldığında, bazı programlarda deney grubunun ortalaması bazılarında kontrol grubunun daha yüksek çıkmıştır. Odak grup görüşmelerinde öğrenci görüşleri incelendiğinde ters yüz sınıftaki öğrencilerin kendi kendine öğrenmenin daha iyi olduğu, yüz yüze sınıftaki öğrencilerin öğretmen anlatımının daha iyi olduğu, ters yüz sınıftaki öğrencilerin anlık yardım alamamasından, yüz yüze sınıftaki öğrencilerin sınıf içi etkenlerin, dersin hızı ve öğretmen anlatımında konuyu kaçırmaktan dolayı zorluklar yaşadıklarını, her iki grupta keşfederek öğrenmenin

daha iyi olabileceđi, oyun ve animasyon yapabilecek becerilerin geliřtiđi dűřüncelerini ifade etmiřlerdir.

Anahtar Sűzcűkler: Çocuklara Programlama Eđitimi, Scratch, Ters Yűz Sınıf, Ters Yűz Sınıf ile Programlama Eđitimi, Scratch ile Programlama Eđitimi

ABSTRACT

THE INVESTIGATION OF DIFFERENT TEACHING STRATEGIES DURING TEACHING PROGRAMMING PROCESS IN BLOCK BASED ENVIRONMENT IN TERMS OF DIFFERENT FACTORS

ECMEN ERDEM

**DEPARTMENT OF COMPUTER EDUCATION AND INSTRUCTIONAL
TECHNOLOGY**

FACULTY OF EDUCATION

MAY 2018

This study aims to investigate how 5th Grade students learn Scratch programming through two different learning and teaching strategies which are face-to-face education and flipped learning based on technology supported environment and the effects of these programming methods on the computational thinking skills of the students. 79 students among 5th graders from a private school participated in this study. The Exploratory Sequential Design which is one of the combined research method including both quantitative and qualitative methods was used. In the quasi-experimental study, which pre-test and post-test were compared with a control group, Self-Sufficiency Perception Evaluation Scale for Computational Thinking Skills and Bilge Kunduz International Informatics and Computational Thinking Questions were conducted as pre-test and post-test. Focus group interviews with 24 students were carried out after the implementation. Weekly achievement tests related to the identified objectives were conducted and the programs written by the students during the implementation were analysed and evaluated through Dr. Scratch during the implementation. The study has revealed that face-to-face education and flipped learning has no statistically effect on learning Scratch programming and computational thinking skills. When the scores of the pre-test and post-test of Self-Sufficiency Perception Evaluation Scale for Computational Thinking Skill were analysed, there was no significant difference between the groups in terms of computational thinking skills and self-sufficiency perceptions while a significant difference occurs on the scores of pre-test and post-test within the groups. Because of the significant difference between the scores of pre-test and post-test and the scores of developing algorithm and problem solving competence, it can be said that the programming method has an important role on the increase of this competence. It can also be stated that programming method has a positive effect on self-sufficiency related to other abilities. Weekly achievement tests have revealed that the average of the control group was high in some weeks while the average of the experimental group was high in other weeks. Similarly, the average of the control group was high in some weeks while the average of the experimental group was high in other weeks in some programmes according to the average scores of the programmes analysed by Dr. Scratch. When the students' views through the focus group interviews has been analysed, the students in the flipped classroom has expressed that it is better to learn by yourself although they cannot get immediate help while the students in the face-to-face classroom has stated that it is better to learn from a teacher although

they have some difficulties because of classroom factors, the speed of the lesson and missing the subject taught by the teacher. However, both groups have expressed that discovery learning is better and their abilities to create game and animations develop during the lessons.

Key Words: Programming Education with Children, Scratch, Flipped Learning, Programming Education through Flipped Learning, Programming Education through Scratch

İÇİNDEKİLER

1. GİRİŞ	1
2. İLGİLİ LİTERATÜR	5
2.1. Programlama Öğretimi ve Bilgi İşlemsel Düşünme	5
2.1.1. Programlama Öğretim Yöntem ve Yaklaşımları	7
2.1.2. Programlama Öğretiminde Blok Tabanlı Öğretim Araçları	9
2.1.2.1. Scratch Programlama Ortamı ve Özellikleri	10
2.1.3. Teknoloji Destekli Programlama Öğretimi	17
2.1.3.1. Araştırmada Kullanılan ÖYS Hakkında Bilgiler	23
2.2. Programlama Öğretiminde Değerlendirme	25
2.2.1. Otomatik Değerlendirme Ortamları	26
2.2.2. Diğer Değerlendirme Araçları	28
3. YÖNTEM	31
3.1. Araştırma Modeli	31
3.2. Çalışma Grubu	31
3.3. Veri Toplama Araçları	32
3.3.1. Bilgi İşlemsel Düşünme Becerilerine Yönelik Öz Yeterlik Algısı Değerlendirme Ölçeği (BİDBÖA)	33
3.3.2. Bilge Kunduz Uluslararası Enformatik ve Bilgi İşlemsel Düşünme Soruları	33
3.3.3. Haftalık Başarı Testi	37
3.3.4. Haftalık Öğrenci Çalışmaları	39
3.4. Uygulama Süreci	40
3.4.1. Hazırlık Çalışması	42
3.4.2. Ön Testlerin Uygulanışı	42
3.4.3. Scratch'a Kayıt ve ÖYS Kullanımı	42
3.4.4. Sahne ve Karakter Özellikleri, Karakteri Boyutlandırma	42
3.4.5. Karakter Zıplama Hareketi, Karakter Yürüyüş Animasyonu	43
3.4.6. Fare ile Etkileşim ve Klavye ile Etkileşim	43
3.4.7. Penaltı – Gol	44
3.4.8. Havai Fişek	44
3.4.9. Biraz da Geometri	45
3.4.10. Analog Saat	45
3.5. Verilerin Analizi	46
4. BULGULAR	48

4.1. Ters Yüz Sınıf Modeli ve Yüz Yüze Eğitim ile Programlama Öğretiminin Bilgi İşlemsel Düşünme Becerilerine Etkisine İlişkin Bulgular	48
4.1.1. Uygulama Öncesinde Deney ve Kontrol Gruplarına Ön Test Olarak Uygulanan Bilge Kunduz Uluslararası Enformatik ve Bilgi İşlemsel Düşünme Sorularına İlişkin Bulgular.....	49
4.1.2. Uygulama Sonrasında Deney ve Kontrol Gruplarına Son Test Olarak Uygulanan Bilge Kunduz Uluslararası Enformatik ve Bilgi İşlemsel Düşünme Sorularına İlişkin Bulgular.....	49
4.2. Programlama Öğretiminin Deney ve Kontrol Grubu Öğrencilerinin Bilgi İşlemsel Düşünme Becerilerine Etkisine İlişkin Bulgular.....	50
4.2.1. Deney Grubuna Uygulama Öncesi ve Sonrası Uygulanan Bilgi İşlemsel Düşünme Becerilerine Etkisine İlişkin Bulgular	50
4.2.2. Kontrol Grubuna Uygulama Öncesi ve Sonrası Uygulanan Bilgi İşlemsel Düşünme Becerilerine Etkisine İlişkin Bulgular	51
4.3. Ters Yüz Sınıf Modeli ve Yüz Yüze Eğitim ile Programlama Öğretiminin Bilgi İşlemsel Düşünme Becerilerine Yönelik Öğrencilerin Öz Yeterlik Algısına İlişkin Bulgular.....	51
4.4. Programlama Öğretiminin Deney ve Kontrol Grubu Öğrencilerinin BİDBÖA Alt Faktörlerine İlişkin Bulguları	55
4.5. Haftalık Değerlendirmelerde Kazanım Bazlı Başarı Puanları	58
4.6. Haftalık Öğrenci Çalışmalarının Değerlendirilmesi	59
4.7. Uygulama Sonrası Öğrencilerin Scratch Programlamayı Farklı Stratejiler ile Öğrenme Konusundaki Görüşler.....	60
4.7.1. Scratch Programlama Dilini Farklı Stratejiler ile Öğrenme Konusundaki Görüşler	61
4.7.2. Scratch Programlama Dilini Farklı Stratejiler ile Öğrenme Konusunda Yaşanılan Zorluklar ile İlgili Görüşler.....	62
4.7.3. Scratch Programlama Dilini Daha İyi Öğrenme Konusunda Görüşler.....	64
4.7.4. Scratch Programlama Ortamında Program Yazmanın Kazandırdıkları Konusunda Görüşler	65
4.7.5. Scratch Programlama Ortamında Programlama ile İlgili Öğrenilen Kavramlar Hakkında Görüşler.....	65
5. TARTIŞMA ve SONUÇ.....	67
6. ÖNERİLER.....	71
KAYNAKLAR	74

TABLolar DİZİNİ

Tablo 1. Scratch Çalışma Sayfasındaki Kod Blokları ve Açıklamaları.....	12
Tablo 2. Ön Test - Son Test Eşleştirilmiş Kontrol Grubu Desen	31
Tablo 3. BİDBÖA Güvenilirlik Katsayıları.....	33
Tablo 4. Araştırmada Uygulanan Bilge Kunduz Uluslararası Enformatik ve Bilgi İşlemsel Düşünme Soruları için Madde Ayırıcılık İndeksi	34
Tablo 5. Ulusal Olarak Uygulanan Bilge Kunduz Uluslararası Enformatik ve Bilgi İşlemsel Düşünme Testi için Madde Ayırıcılık İndeksi	35
Tablo 6. Bilge Kunduz Uluslararası Enformatik ve Bilgi İşlemsel Düşünme Soruları.....	36
Tablo 7. Haftalık Başarı Testleri ve Kazanım İlişkileri.....	37
Tablo 8. Haftalık Başarı Testlerinin Güvenilirlik Katsayıları	38
Tablo 9. Haftalık Başarı Testlerinin Madde Ayırıcılık İndeksi	39
Tablo 10. Haftalık Konu Dağılımları.....	41
Tablo 11. Ön Test için Yapılan Bağımsız t-Testi Sonuçları.....	49
Tablo 12. Son Test için Yapılan Bağımsız Örneklem t-Testi Sonuçları	49
Tablo 13. Deney Grubu Ön Test ve Son Test Bağımlı Örneklem t-Testi Sonuçları	50
Tablo 14. Kontrol Grubu Ön Test ve Son Test Bağımlı Örneklem t-Testi Sonuçları	51
Tablo 15. BİDBÖA Ön Testi Tüm Faktörler için Yapılan Bağımsız t-Testi Sonuçları	53
Tablo 16. BİDBÖA Son Testi Tüm Faktörler için Yapılan Bağımsız t-Testi Sonuçları	54
Tablo 17. Deney Grubu BİDBÖA Ön Test ve Son Test Algoritma Tasarlama Yeterliği için Yapılan Bağımlı Örneklem t-Testi Sonuçları.....	56
Tablo 18. Kontrol Grubu BİDBÖA Ön Test ve Son Test Tüm Alt Faktörler için Yapılan Bağımlı Örneklem t-Testi Sonuçları	57
Tablo 19. Haftalık Değerlendirmeler için Kazanım Bazlı Başarı Sonuçları	59
Tablo 20. Haftalık Öğrenci Çalışmalarının Dr. Scratch ile Elde Edilen Puanların Ortalamaları	60
Tablo 21. Scratch Programlama Dilini Farklı Stratejiler ile Öğrenme Konusundaki Görüşler	61
Tablo 22. Scratch Programlama Dilini Farklı Stratejiler ile Öğrenme Konusunda Yaşanılan Zorluklar ile İlgili Görüşler.....	63
Tablo 23. Scratch Programlama Dilini Daha İyi Öğrenme Konusundaki Görüşler	64

Tablo 24. Scratch Programlama Ortamında Program Yazmanın Katkıları Konusunda Görüşler	65
Tablo 25. Scratch Programlama Ortamında Programlama ile İlgili Öğrenilen Kavramları Hakkında Görüşler.....	66

ŞEKİLLER DİZİNİ

Şekil 1. Scratch Web Sitesinin Arayüzü.....	11
Şekil 2. Scratch Web Sitesinin Giriş Yaptıktan Sonraki Arayüzü.....	11
Şekil 3. Scratch Web Sitesinde Uygulama Geliştirme Ekranı.....	12
Şekil 4. Scratch Web Sitesinde Geliştirilen Örnek Uygulamanın Kodları.....	14
Şekil 5. Ters Yüz Sınıf Modeli.....	18
Şekil 6. Bloom Taksonomisine Göre Geleneksel ve Ters Yüz Sınıf Modeli (Nederveld, 2014).....	19
Şekil 7. Ters Yüz Sınıf Modeli (Waterworth, 2016).....	19
Şekil 8. Araştırmada Kullanılan ÖYS'ye Ait Ekran Görüntüsü.....	23
Şekil 9. Araştırmada Kullanılan ÖYS'ye Ait Ekran Görüntüsü.....	24
Şekil 10. Araştırmada Kullanılan ÖYS'ye Ait Ekran Görüntüsü.....	24
Şekil 11. Araştırmada Kullanılan ÖYS'ye Ait Ekran Görüntüsü.....	25
Şekil 12. Dr. Scratch ile Proje Değerlendirme Raporu.....	27
Şekil 13. Dr. Scratch ile Proje Değerlendirme Raporu.....	27
Şekil 14. Dr. Scratch ile Proje Değerlendirme Raporu.....	28
Şekil 15. Veri Toplama Süreci.....	32
Şekil 16. Yüz Yüze Eğitim Yapan Grup için Kullanılan Öğretim Yöntemleri.....	40
Şekil 17. Ters Yüz Sınıf Model ile Eğitim Yapan Grupta Kullanılan Öğretim Yöntemleri.....	40
Şekil 18. Sahne ve Karakter Özellikleri, Karakteri Boyutlandırma Etkinliği.....	43
Şekil 19. Karakter Zıplama Hareketi, Karakter Yürüyüş Animasyonu Etkinliği....	43
Şekil 20. Fare ve Klavye ile Etkileşim Etkinliği.....	44
Şekil 21. Penaltı – Gol Etkinliği.....	44
Şekil 22. Havai Fişek Etkinliği.....	45
Şekil 23. Biraz da Geometri Etkinliği.....	45
Şekil 24. Analog Saat Etkinliği.....	46

1. GİRİŞ

1.1 Genel Bakış

Teknolojinin kuşkusuz hayatımızda çok önemli bir yeri bulunmaktadır. Hayatın birçok alanında teknolojinin etkili olduğu, bireyleri yönlendirdiği, alışkanlıklarını etkilediği, ihtiyaç alanlarını değiştirdiğini söylemek mümkündür. Bilgisayarlar ilk kullanıldığı dönemlerde bir ihtiyaç olduğu görülmezken, bugün hemen hemen herkesin önemli ihtiyaçları arasında yer almaktadır. Toplum olarak gelişim ise var olan bilgiye ulaşma ve bunu etkili kullanmakla olmaktadır. Bu bilgiye ulaşmanın yolu bilgi ve iletişim teknolojilerinden en etkin bir şekilde yararlanmakla gerçekleşmektedir. Bunun için de bilgi teknolojilerini etkili kullanarak ve üretebilen, bilgi toplumu olmak daha da önemli hale gelmiştir (Seferoğlu, 2015). Küresel bir toplumda bilgi çağındaki gelişmeleri takip edebilmek için Kalelioğlu ve Gülbahar'ın (2014) da belirttiği gibi bilgi ve iletişim teknolojilerini kullanma, medya ve bilgi okuryazarlığı becerilerine ihtiyaç duyulmaktadır. Bu etki alanının genişlemesi ile birlikte bireylerden beklentiler ve beceriler de değişmiştir. Özellikle yaşadığımız çağda teknoloji ve bilgi okuryazarlığı, yaratıcılık, iletişim, eleştirel düşünebilme, yüksek performans gösterme, birçok görevi aynı anda yürütebilme, sorumluluk sahibi olma, işbirlikçi çalışabilme, problem çözme becerileri gibi 21. yy. becerileri öğrencilerin kazanmaları gereken beceriler olarak belirtilmektedir (Yüzer ve Kılınç, 2015).

Bireylerin sahip olması gereken bu becerileri Koenig (2011) bilişsel beceriler, bireylerarası beceriler ve bireysel beceriler olarak sınıflandırmıştır. Buna göre, bilişsel beceriler, sıra dışı problem çözme, sistematik düşünme, eleştirel düşünme becerileri; bireylerarası beceriler ise farklı bireyler ile iletişim, takım olarak çalışabilme, kültürel duyarlık gösterebilme, sosyal ilişkiler kurabilme, farklılıklarla ilgilenme becerileri iken son olarak bireysel beceriler de kendini yönetme, zamanı kullanabilme, kişisel gelişim, öz denetim, uyum sağlama becerileri şeklinde ifade etmiştir. Bunlara ek olarak 21. yy. bireylerinde bulunması gereken; yaratıcılık, problem çözme, algoritmik düşünme gibi temel beceriler, programlama becerilerinin programlama ve bilgisayar bilimi öğretimi ile kazandırılabilmesi gerçeği ile hareketle (Shin ve Park, 2014) son yıllarda erken yaşta programlama öğretimi ile bu becerilerin desteklenmesi tüm dünyada önem kazanmıştır.

Ülkemizde de bu doğrultuda Millî Eğitim Bakanlığı Talim Terbiye Kurulunun 2018 yılı başında yayınladığı Bilişim Teknolojileri ve Yazılım Dersi Öğretim Programları amaçlarında bireylerin problem çözme, bilgi-işlemsel düşünme, algoritma tasarlama becerileri gibi birçok kavramın kazandırılması hedeflenmiştir. Bu gelişmeler ve bilgiler ışığında programlama eğitiminin erken yaşta eğlenceli ve öğrencileri motive edecek şekilde öğretilmesi için farklı yöntemlerin, araçların, gereçlerin kullanılmasının gerekliliği doğmuştur (Gülmez, 2009). Bu nedenle küçük yaşta öğrencilerin programlama becerilerini kazanması için sınıf ortamlarının farklı yöntemler ile desteklenerek pedagojik açıdan uygunluğunu tartışmak önemli olmuştur.

1.2 Çalışmanın Amacı

Bu çalışmanın amacı ortaokul öğrencilerinin yüz yüze öğretim ve teknoloji destekli öğretim ortamlarında olmak üzere iki farklı yöntem ile blok programlama araçlarından olan Scratch programlamayı öğrenmelerinin, öğrencilerin bilgi işlemsel düşünme becerilerine etkisini araştırmaktır.

1. Blok tabanlı programlama öğretimini ters yüz sınıf modeli ve yüz yüze eğitim ile alan öğrencilerin uygulama öncesi ve sonrası bilgi işlemsel düşünme becerileri arasında anlamlı bir fark var mıdır?
2. Blok tabanlı programlama ortamında programlama kavramlarını öğrenmede ters yüz sınıf modeli ile yüz yüze eğitim ile öğrenen öğrencilerde uygulama öncesi ve sonrasında anlamlı bir fark var mıdır?
3. Blok tabanlı programlama öğretimini ters yüz sınıf modeli ve yüz yüze eğitim ile alan öğrencilerin uygulama öncesi ve sonrası bilgi işlemsel düşünme becerilerine yönelik öz yeterlik algıları arasında anlamlı bir fark var mıdır?
4. Blok programlama öğretimini ters yüz sınıf modeli ve yüz yüze eğitim ile alan öğrencilerin bu derse ilişkin görüşleri nelerdir?

1.3 Çalışmanın Önemi

Koenig (2011) bilişsel becerileri, sıra dışı problem çözme becerisi, eleştirel düşünme, sistematik düşünme şeklinde ifade etmiştir. 21. yy. becerilerinden; problem çözme, yaratıcılık, algoritmik düşünme gibi temel beceriler, programlama ve bilgisayar biliminin öğretilmesi ile sağlanmaktadır (Shin ve Park, 2014).

Bu beceriler, bir problem karşısında hızlı ve doğru bir analiz yapıp gerekli adımları uygulamayı bu da problemi en hızlı şekilde çözmeyi sağlar. Wing (2008) bu becerilerden bilgi işlemsel düşünmenin analitik düşünme becerisi olduğunu belirtmektedir. Bu beceriyi bilişim teknolojileri dersinde geliştirmek, öğrencilerin teknolojiyi daha etkin kullanmalarını ve çeşitli disiplinler ile birlikte çalışarak üreten bir topluluk olacağı düşünülmektedir. Yüz yüze öğrenme ile teknoloji destekli öğrenme modellerinin birleştirmesiyle farklı bir model olarak ortaya çıkan harmanlanmış öğrenme modeli son yıllarda sıkça kullanılan bir öğrenme ve öğretme modeli olarak literatürdeki yerini almıştır.

Araştırmanın önemi farklı öğretim yöntemleri ile yapılan derslerin öğrencilerin bilgi işlemsel düşünme becerilerinde oluşan farkları tespit edip, bu becerinin verilmesinde hangi yöntemin daha uygun olacağı ve öğretim yöntemlerinin öğrenciler tarafından nasıl algılandığını belirlemektir. Araştırmanın sonuçlarının bu alandaki yeni araştırma ve uygulamalara katkı sağlaması beklenmektedir. Bununla birlikte öğrencilerin her iki yöntemle ilişkin düşüncelerini tespit edip, öğrencilerden gelen öneriler ile ileride bu alanda yapılan çalışmalara katkı sağlayacağı düşünülmektedir. Ayrıca, bu çalışma bilişim teknolojileri öğretmenlerine sınıf ortamlarını programlama öğretimi için nasıl tasarlaması gerektiğine yönelik ipuçları sunması açısından da ışık tutacağı düşünülmektedir.

1.4 Sayıtlar

- Araştırmaya katılan öğrencilerin bilişsel özellikleri bakımından birbirlerine yakın oldukları,
- Araştırma sonucunu etkileyecek dış etkenlerin, her iki grupta yer alan öğrencileri eşit düzeyde etkilediği,
- Kontrol ve deney gruplarında yer alan öğrencilerin araştırmanın sonucunu etkileyecek bir şekilde iletişime ve etkileşime girmedikleri,
- Araştırmanın veri toplama sürecinde, öğrencilere uygulanan “Bilgi İşlemsel Düşünme Becerilerine Yönelik Öz Değerlendirme Ölçeği”, “Bilgi İşlemsel Düşünme Etkinliği” ve yapılan odak grup görüşmelerinde sorulan sorulara samimi şekilde yanıt verdikleri kabul edilmiştir.

1.5 Sınırlılıklar

1. Çalışma Ankara’da özel bir okulun Bilişim Teknolojileri ve Yazılım Dersini alan 5. sınıf öğrencileri ile sınırlıdır.
2. Bilgi işlemsel düşünme becerisi öz yeterlik algısının değerlendirilmesi “Bilgi İşlemsel Düşünme Becerilerine Yönelik Öz Yeterlik Algısı Değerlendirme Ölçeği” ile sınırlıdır.
3. Bilgi işlemsel düşünme becerilerin ölçülmesi 2014-2015 yıllarına ait “Bilge Kunduz Uluslararası Enformatik ve Bilgi İşlemsel Düşünme Soruları” ile sınırlıdır.
4. Bilgi işlemsel düşünme becerilerin ölçülmesi her ders sonunda yapılan sınavlar ile sınırlıdır.

1.6 Kısaltmalar

Kısaltma	Açıklama
UETB	: Uluslararası Eğitimde Teknolojiler Birliği
BBÖD	: Bilgisayar Bilimi Öğretmenleri Derneği
BİDBÖA	: Bilgi İşlemsel Düşünme Becerilerine Yönelik Öz Yeterlik Algısı Değerlendirme Ölçeği
BİD	: Bilgi İşlemsel Düşünme
ÖYS	: Öğrenme Yönetim Sistemi

1.7 Tanımlar

Bilgi İşlemsel Düşünme: Problemleri formüle ederek bilgisayar ve benzeri araçlarla çözüm üretme, elde olan veriyi analiz etme, düzenleme, modelleme algoritmik düşünme ile çözümler de kullanılabilir kaynakları en verimli şekilde birleştirme ve analiz etme, bu çözümleri genelleyerek diğer problemlerde kullanabilmektir. (UETB ve BBÖD, 2011)

Scratch: MIT tarafından geliştirilen blok şeklinde programlanan grafik, ses ve video destekli zengin bir dijital programlama aracıdır.

Ters Yüz Sınıf: Öğrencinin evde bilgiyi kendi başına öğrenip, okulda öğrendiklerini uygulama fırsatı sunan bir yöntemdir (Zownorega, 2013).

2. İLGİLİ LİTERATÜR

2.1. Programlama Öğretimi ve Bilgi İşlemsel Düşünme

Programlamayı öğrenmek, kod yazmanın ötesinde bireylerin yaratıcı düşünme, işbirlikçi öğrenme, problem çözme gibi becerileri kazanmalarında önemli rol oynamaktadır. Lau ve Yuen (2011), programlama öğrenmenin; sistematik düşünme, iletişim, takım haline iş birliği içerisinde çalışabilme, problemi analiz etme, yöntem geliştirme ve çözme becerilerini geliştirdiğini şeklinde ifade etmişlerdir. Bundan dolayı programlama becerisi 21. yy. becerilerinden biri haline dönüşmüş olduğu görülmektedir (Durak, Karaođlan Yılmaz, Yılmaz ve Seferođlu, 2017). Bu düşüncelere benzer bir şekilde Fessakis, Gouli ve Mavroudi (2013) programlama öğrenmenin ve bu becerilere erken yaşta sahip olma, üst düzey düşünmeyi, algoritmik düşünmeyi, problem çözme becerilerinin gelişmesinde önemli bir rol oynadığını ifade etmiştir.

Akpınar ve Altun (2014), programlama öğretimi öğrencilerin uzamsal düşünme becerilerini, teknolojik okuryazarlıklarını, işbirlikli çalışma becerilerini, bilgisayar ile öğrenme becerilerini geliştirdiğini belirtmiştir. Bu görüşlere ek olarak Shin ve Park (2014) son yıllarda, bütün dünyadaki uzmanlar öğrencilerin erken yaşlarda programlamayı öğrenmesi gerektiğini belirtmektedir. Çünkü, 21.yy. becerileri olarak belirtilen; problem çözme, yaratıcılık, bilgi işlemsel düşünme gibi becerilerin programlama öğrenilmesi ile geliştiđi belirtilmektedir. Benzer bir görüşü Kaleliođlu (2015), çocuklarda programlama öğretiminin algoritmik düşünme, eleştirel ve yaratıcı düşünme, problem çözme gibi üst düzey düşünme becerilerine olumlu etkilerinin bulunduđunu ifade etmiştir. Bu ifadelerde yer alan görüşlerle birlikte küçük yaştaki öğrencilere özel görsel özelliklere sahip programlama araçları kullanılarak programlama öğretiminin ilköğretim kurumlarında yaygınlaştırılması amaçlanmıştır (Çatlak, Tekdal ve Baz, 2015).

Programlama öğretiminin bireye kazandırdığı becerilerden birisi olan problem çözme becerileri çocuk yaştan itibaren öğrenilmekle ve okul yıllarında da bu beceriler çeşitli etkinlikler ile geliştirilmektedir. Bu becerilerin geliştirilmesine yönelik problemi anlama, çözüm planı arama, çözüm planını uygulama ve sonuçları değerlendirme ana başlıkları altında sıralanmıştır. Problem çözebilen bireylerin yaratıcı düşünebildikleri, sosyal ilişkilerde daha aktif olduđu, kendilerine güvenen ve problemleri rahatlıkla aşabilen bireyler olduđu belirtilmektedir (Serin, Bulut Serin ve Saygılı, 2010).

Problem çözme becerileri geliştirmede en çok kullanılan yöntemlerden birisi olan Probleme dayalı öğrenme, öğrencilerin bir konuyu birden fazla probleme maruz bırakarak öğrenmelerine ve konuyla ilgili öğrendiklerini bu problemlerle ilişki kurmalarını sağlamaya yönelik yapıcı bir yöntemdir. Bu tür bir öğrenmede öğrenciler problemleri çok farklı şekillerde çözmeye çalışır, zihinlerini canlandırır (Hmelo-Silver ve Barrows, 2006). Bilgi işlemsel düşünme ve problem çözme becerilerinin arasında büyük benzerlikler bulunduđunu ve bundan dolayı problem çözme becerilerinin aynı zamanda bilgi işlemsel düşünme becerileri olduđunu söylenebilir.

Bilgi işlemsel düşünme birçok araştırmacı tarafından farklı zamanlarda tanımlanmıştır. Aho (2012), bilgi işlemsel düşünmeyi, problemleri formüle etmede kullanılan, çözümleri algoritma olarak gösterilebilir düşünce süreçleri olarak ifade etmektedir. Araştırmacı bu düşünme becerisinin önemli bir parçasını problemi formüle etmek ve çözümlerini üretmek için uygun problem çözme modelleri bulmaktır diye açıklamaktadır. Mannila ve diğerleri (2014) bilgi işlemsel düşünmeyi, problemleri farklı disiplinlerde çözümlerini formüle etmeye yardımcı olan bilgisayar bilimlerinden bir dizi kavram ve düşünce sürecini kapsayan bir terim olduğu şeklinde tanımlamıştır. Wing (2006, 2011) bilgi işlemsel düşünmeyi, bilginin etkin bir biçimde işlenmesi için gerekli problemlerin çözümlerini üretmeyi içeren düşünce süreçleri olarak ifade etmiştir.

Futschek'e (2006) göre bilgi işlemsel düşünme, anlama ve analiz etme gibi birçok alt becerilerden oluşan bir kavramdır. Var olan bir problemi analiz etme, ifade etme, problemin çözümü için yöntemler bulma, bu yöntemler ile problemin çözümüne uygun algoritma oluşturma alt becerilerinin tamamını kapsamaktadır. Pulimood, Pearson ve Bates (2016) bilgi işlemsel düşünmeyi soyut işlemler için mantık yürütme ve çözümler üretme olarak tanımlarken, BCS (2014) bilgi işlemsel düşünmeyi, kimseden yardım almadan bir bilgisayarın üstesinden gelebilecek çözümü üreten zihinsel bir beceri olarak tanımlamıştır.

UETB ve BBÖD (2011), bilgi işlemsel düşünme becerisini problemleri formüle ederek bilgisayar ve benzeri araçlarla çözüm üretme, elde olan veriyi analiz etme, düzenleme, modelleme algoritmik düşünme ile çözümler de kullanılabilir kaynakları en verimli şekilde birleştirme ve analiz etme, bu çözümleri genelleyerek diğer problemlerde kullanabilmek olarak tanımlamıştır. Hu (2011) bilgi işlemsel düşünmeyi; bir bilgisayarın çalışma prensibindeki gibi bilgi işleme araçlarını kullanarak problemi çözmek, çözümü genellemek ve diğer problemlere uyarlamak, çözümleri analiz ederek somutlayarak veya soyutlayarak yeni sistemler tasarlayabilme becerileri şeklinde tanımlamıştır.

UETB (2016) öğrencilerin sahip olması gereken standartları güçlendirilmiş öğrenci, dijital vatandaşlık, bilgi oluşturucu, yenilikçi tasarımcı, bilgi işlemsel düşünme, yaratıcı düşünme ve iş birliği becerisi olarak belirtmiştir. UETB (2016) standartlarından bilgi işlemsel düşünme becerilerinde öne çıkan davranışları şu şekilde tanımlanmıştır:

- Öğrenciler, veri analizi, soyut modeller ve çözüm arama ve bulmada algoritmik düşünme gibi teknoloji destekli yöntemler için uygun olan problem tanımlarını formüle ederler.
- Öğrenciler, veri toplamakta, ilgili veri setlerini belirlemekte, bunları analiz etmek için dijital araçları kullanırlar. Problem çözme ve karar vermeyi kolaylaştırmak için verileri çeşitli yollarla analiz ederler.
- Öğrenciler, problemleri bileşen parçalarına bölerek önemli bilgileri çıkarırlar ve karmaşık sistemleri anlamak ve problem çözmeyi kolaylaştırmak için tanımlayıcı modeller geliştirirler.
- Öğrenciler otomasyonun nasıl çalıştığını anlarlar, otomatik çözümler üretmek ve test etmek için bir dizi adım geliştirmek için algoritmik düşünceyi kullanırlar.

Bilgi işlemsel düşünme ile bir problem daha küçük, daha yönetilebilir bir dizi (parçalama) oluşturmayı içerir. Bu daha küçük problemlerin her biri, daha önce benzer problemlerin nasıl çözüldüğü (benzerlik arama) ve sadece önemli ayrıntılara odaklanırken, ilgisiz bilgileri (soyutlama) göz ardı ederek incelenebilir. Daha sonra, küçük problemlerin her birini çözmek için basit adımlar veya kurallar (algoritmalar) tasarlanabilir (BBC, 2018).

Bilgi işlemsel düşünme becerilerinin programlama öğretimi ile geliştirilmesi üzerine yapılan çalışmalara bakıldığında Howland ve Good (2015) yaşları 12 ile 13 arası değişen ortaokul öğrencileri ile yaptıkları çalışmada oyun tasarım sürecinin bilgi işlemsel becerilerine etkisini incelemiştir. Çalışmada bilgi işlemsel kavramları ölçen soruların yer aldığı ön test- son test uygulanmış ve geliştirilen uygulamalar incelenmiştir. Araştırma sonucuna göre oyun tasarım sürecinin öğrencilerin programlama becerilerini ve bilgi işlemsel becerilerini geliştirdiği ortaya çıkmıştır.

Şanal ve Erdem (2017), yaptıkları çalışmada kodlama ve robotik üzerine çalışma yapan ve yapmayan öğrencilerin problem çözme becerileri arasındaki farklılıkları incelemiştir. Araştırmada öğrencilere teknik ve sosyal problemler verilip bu problem durumlarına uygun çözüm bulmaları istenmiştir. Sonuç olarak kodlama yapan öğrencilerin teknik problemlerde problem çözme süreçleri kodlama yapmayan öğrencilerin çözüm süreçlerinden daha verimli olduğunu belirtmişlerdir.

Alan yazın ve araştırmalar incelendiğinde programlama öğretiminde çeşitli yöntem ve yaklaşımların kullanıldığı görülmektedir. Bu yöntem ve yaklaşımların kullanılmasının temel felsefesi programlamayı farklı yaş gruplarına uygun olarak anlatmak ve öğrencilerin bu yöntemler ile programlamayı öğrenmelerini desteklemektir.

2.1.1. Programlama Öğretim Yöntem ve Yaklaşımları

Programlama öğretimi tüm dünyada çeşitli yöntem ve yaklaşımlar ile yapılabilmektedir. Yapılan çalışmalarda kimi araştırmacılar var olan öğretim yöntemlerini programlama öğretimine uyarlarken, bazıları ise kendi programlama dersi müfredatlarını oluşturup buna uygun yöntemler geliştirmişlerdir.

Lye ve Koh (2014) ilk ve ortaokul öğrencilerinin etkili bir şekilde programlama öğrenmeleri için sesli düşünme tekniğini kullanarak kodlamayı ilk olarak sözel olarak ifade edebilmeleri gerektiğini daha sonrasında programlama çalışmalarına başlamaları gerektiğini ifade etmişlerdir. Grover, Pea ve Cooper (2015), ortaokul öğrencileri için “Bilgi İşlemsel Düşünmeyi Geliştirmenin Temelleri (BİDGT)” adı verilen karma bir müfredat oluşturup test etmişlerdir. 2015 yılında yaptıkları çalışmada, bu müfredatın “derin öğrenme” için geliştirildiği, yanlış öğrenmelerin giderilmesi ve öğrenme sonrası değerlendirme sistemlerini de içinde barındırdığı, bloktan temelli programlamadan metin tabanlı programlamaya geçişi desteklemeye yönelik pedagojik stratejilere odaklandığı belirtilmiştir. Araştırmacılar, BİDGT’in öğrencilerin bilgi işlemsel düşünmede önemli kazanımlar elde ettiklerini, öğrendikleri kazanımları programlama öğrenirken kullanabildiklerini belirtmişlerdir. Bu müfredat tek bir pedagojik yaklaşıma odaklanmayıp, sorgulamayı ve bilişsel öğrenmeyi harmanlayan pedagojik bir yaklaşımdan oluşmaktadır. Bunun yanı sıra müfredat kod okuma ve izleme, yüksek sesle

düşünme, bilgisayar dilini kullanma, açık uçlu projeler, bağımsız ve işbirlikçi çalışmalar gibi yöntemleri içermektedir.

Webb, Repenning ve Koh (2012) ölçeklenebilir oyun tasarımı adı verdikleri projelerinde öğretilme şekillerinin kız ve erkek öğrencilerin farklı etkilendiklerini belirtmişlerdir. Kız öğrencilerin doğrudan talimatlar ile verilen eğitimlerde daha az başarılı olduklarını belirtmişlerdir. Bunun yerine rehber eşliğinde keşfederek öğrenmenin kız öğrencilere daha faydalı olduğunu belirtmişlerdir. Hansen, Hansen, Dwyer, Harlow ve Franklin (2016) 4-6. sınıflarda okuyan öğrencilere programlama öğretmek için kodlama müfredatı geliştirmişlerdir. Bu müfredat iki modül olarak geliştirilmiştir. Her modül 12-16 saat uygulanmaktadır. Her iki modülde çocuklara uygun olarak tasarlanmıştır ve blok tabanlı programlama ortamında öğrenciler hem programlama becerilerini geliştirirken diğer yandan bilgi işlemsel düşünme becerilerini attırmaktadır. Her modül sonunda öğrenciler kendi projelerini kodlayarak, tasarlayarak geliştirmeleri beklenmektedir. Teague ve Lister (2014) programlamayı öğrenmenin sıralı ve artarak ilerlemeli olduğunu, yeni öğrenenlerin, kod parçalarını yazmadan önce bu kodları okuyarak anlamlandırılmalarını, bunu sözel olarak ifade etmelerini bununla beraber kod izleme etkinliklerin yapılmasını ve sorasında küçük görevlerle başlayarak kod yazma etkinliklerinin yapılmasını önermişlerdir.

Gülmez (2009), programlama eğitiminde algoritmayı hikayeleştirerek öğrenmeleri programlama becerilerinin arttırdığını ifade etmiştir. Erümit ve diğerleri (2017) algoritmik düşünme becerisini geliştirmede bilgisayar olmadan ders planları oluşturulmasını önermişlerdir. Alanyazına uygun yöntemleri kullanarak etkinliklerin hazırlanması, planlanması ve öğretimin gerçekleştirilmesi gerektiğini belirtmişlerdir. Bununla beraber programlama eğitimde işbirlikçi etkinlikler mutlaka yer almalı, hazırlanan etkinliklerde oyun tabanlı uygulamalara yer verilmesi gerektiğini ifade etmişlerdir. Benton, Hoyles ve Noss (2017), öğretmen anlatımı ile eş zamanlı kod yazmanın ve önceki matematik bilgilerinin kodlama ile ilişkilendirilmesinin öğrencilerin başarılarına daha fazla etki edeceğini belirtmişlerdir. Kafai ve Burke (2015) oyunların çocuklara teknik beceriler kazandırdığını ve bunları geleneksel dijital oyun kültürüne uydurarak birbirleriyle ilişkilendirdiklerini, bu sebeple programlama öğretiminin oyunlaştırarak verilmesi ile birlikte öğrencilerin başarılarının artacağını belirtmişlerdir.

Programlama öğretimi, yöntemi, programlama dili, hedef kitleye uygun kodlama ortamı gibi birtakım sorunları da içerisinde barındırmaktadır. Özellikle programlamaya yeni başlayan bireylerde en önemli engel metin tabanlı programlama dillerinin karmaşıklığından dolayı öğrenilmelerinin zor olduğu görülmektedir. Bu sorunu gidermek için, öğrenilmesi kolay ve görselliği ön planda tutan blok tabanlı programlama ortamları geliştirilmiştir (Çatlak, Tekdal ve Baz, 2015). Bununla beraber Garneli, Giannakos, ve Chorianopoulos (2015) programlamada etkili, eğlenceli bir öğrenme olması için ve metin tabanlı dilleri daha iyi öğrenmek için görsel programlama dillerinin kullanılmasının faydalı olacağını belirtmişlerdir.

2.1.2. Programlama Öğretiminde Blok Tabanlı Öğretim Araçları

Öğrenciler programlama öğrenmeye başladıklarında programlamayı zor olarak algılamaktadırlar. Bununla beraber algoritmaları oluşturma ve kullanmada zorlanmaktadırlar (Aşkar ve Davenport, 2009). Bu sebeple hem anlamayı hem de uygulamayı kolaylaştırmak için görsel programlama araçlarının kullanılması önerilmektedir (Erümit ve diğerleri 2017). Programlama öğretimi Türkiye’de daha çok olarak lisans öğreniminde ve meslek liselerinde yapılmaktaydı. Günümüzde, özellikle birçok ülkelerde, yazılım geliştirmeyi eğlenceli bir hale getirme ve bu yeterlilikleri çocukların erken yaşta başlamalarını sağlamaya yönelik çalışmalar yapılmaktadır. 2012 yılından itibaren ülkemizde değişen eğitim sistemiyle beraber 5. ve 6. sınıflarda Bilişim Teknolojileri ve Yazılım dersi kazanımları içerisinde yer alan “problem çözme, programlama ve özgün ürün geliştirme” için blok tabanlı araçlar ile çocukların eğlenerek uygulama geliştirebilecekleri programlar kullanılmaktadır. (Berksoy, Sözcü, Armağan ve Arslan, 2016).

Bilişim derslerinin programlama becerilerini, erken yaşlarda öğrencilerde bir yetkinlik olarak kazanılması için yeniden şekillendirdiği açıktır. Birçok eğitimci, öğrencilere programlama becerileri ve mantığını öğretme sürecinde güçlük çekmektedir. Yükseköğretim seviyesinde öğrenciler programlama öğreniminde zorluk çekiyorsa, ilköğretim ve ortaöğretim öğrencileri için en uygun stratejiler belirlenmelidir. Bu sorular göz önüne alındığında, çocuklara programlama öğretme sürecinde, problem çözme becerilerine katkı açısından blok tabanlı araçların kullanmanın etkinliğini araştırmaya çalışılmıştır (Kalelioğlu ve Gülbahar, 2014). Erol (2015), blok tabanlı araçlar ile programlama öğretiminde, programlama mantığının kazandırılması ve temel programlama yapılarının öğretilmesi amaçlamıştır. Araştırmanın nitel verileri incelendiğinde deney grubunda yer alan katılımcılar blok tabanlı araçlar ile oyun tasarımı etkinliklerinin eğlenceli ve kolay olduğunu, ders süresince yapılan etkinliklerin programlama mantığını kazandırmada ve motivasyonu artırmada etkili olduğunu belirtmiştir.

Numanoğlu ve Keser (2017) yaptıkları çalışmada blok tabanlı robot kodlama platformunun programlama öğretiminde kullanılabilirliğini belirlemeyi, programlama yapısını, programla öğretime ilişkin uygulamalar nasıl geliştirileceğini, araştırmayı amaçlamışlardır. Blok tabanlı olarak programlanan yazılımların programlama öğretiminde soyut kavramların kolayca somutlaştırılabileceği ve yazdığı programın etkisini anında gören öğrencilerin problem çözme ve bilgi işlemsel düşünme becerilerini daha kolay ve hızlı bir şekilde geliştirilebileceklerini ifade etmişlerdir.

Papert ve MIT araştırmacıları tarafından geliştirilen ve 1969 yılında öğrencilere mantıksal ve matematiksel düşünme becerileri kazandırmak için geliştirilen Logo programlama dili ile başlayan, son yıllarda dünyanın önde gelen üniversitelerinin ve yazılım firmalarının ortak olarak geliştirdiği çocuklarda problem çözme becerilerini ve bilgi işlemsel düşünme becerilerini arttırmak amaçlı, çocuklarda programlama öğrenmeyi kolaylaştıran ve eğlenceli bir hane getiren çeşitli programlama araçları vardır. Bu araçlara örnek olarak Small Basic, Code Monkey, Code Combat, Kodu, Code.org, Blockly, Appinventor ve Scratch verilebilir.

Small Basic, Microsoft tarafından geliştirilen bir programlama dilidir. Ücretsiz olarak <http://smallbasic.com/> adresinden ulaşılabilir. Small Basic programlamayı daha çok programlamaya yeni başlayan veya başlayacak kişilerin tercih etmektedir. Code Monkey, küçük yaşta çocuklar için oluşturulmuş belirli bir seviyeye kadar ücretsiz olarak programlama yapılabilen bir kodlama platformudur. Burada bir maymun karakterinin kodlar yazarak çeşitli görevleri tamamlaması sağlanmaktadır. Bu uygulamaya <http://www.codemonkey.com/> adresinden ulaşılabilir. Code Combat daha çok aksiyon oyunlarını seven çocuklara hitap eden oyunun başında bir karakter seçerek harita üzerinden sırası ile belirli yerlerde verilen görevleri kodlama yaparak tamamlanması istenilen kodlama oyunudur. Oyunda birçok programlama dili seçeneği bulunmaktadır. Bunlar Python, Javascript, CoffeeScript ve Lua dilleridir. Bu kodlama oyununa <https://codecombat.com> adresinden ulaşılabilir.

Kodu, Microsoft tarafından geliştirilmiştir. Çocukların basit bir görsel programlama dili ile PC ve Xbox'ta oyunlar oluşturmalarına olanak sağlar. Kodu yaratıcılık, problem çözme, hikaye anlatımı ve programlamayı öğretmek için kullanılabilir. Herhangi bir tasarım veya programlama becerisine sahip olmayan yetişkinlerin yanı sıra bir oyun yapmak için herkes kullanabilir. Code.org web tabanlı çalışan <https://code.org/> adresinden erişilebilen Microsoft, Google, Facebook vb. teknoloji firmalarının desteklediği, küçük yaşta itibaren kodlama öğretmeyi hedefleyen bir organizasyondur. Platformda 4 yaş ve üstü, 6 yaş ve üstü, 8 yaş ve üstü, 10 yaş ve üstü gibi çeşitli seviyelerde dersler mevcuttur. İçerisinde blok tabanlı kodlama yaparak verilen görevleri yerine getirilmesi istenilen birçok oyun bulunmaktadır.

Blockly Google tarafından oluşturulmuş erken yaşta çocuklara programlama öğretiminde kullanılan blok tabanlı bir web uygulamasıdır. İçerisinde birçok farklı kodlama etkinlikleri vardır. Aynı zaman da açık kaynak kodlu olmasından dolayı bağımsız kodlama uygulamaları yapılabilir. Appinventor, MIT Media Lab tarafından mobil cihazlarda uygulama yapmak için geliştirilen bir blok kodlama platformudur. Ücretsiz olarak erişilebilen Appinventor ile şu anda sadece Android tabanlı uygulamalar geliştirilebilir. Fakat MIT ilerde IOS ile uygulama da geliştirileceğini belirtmektedir. Bu araştırma için tercih edilen Scratch, MIT tarafından geliştirilen dünyanın en çok kullanılan blok tabanlı her türlü bağımsız olarak uygulama geliştirilebilen bir kodlama ortamıdır.

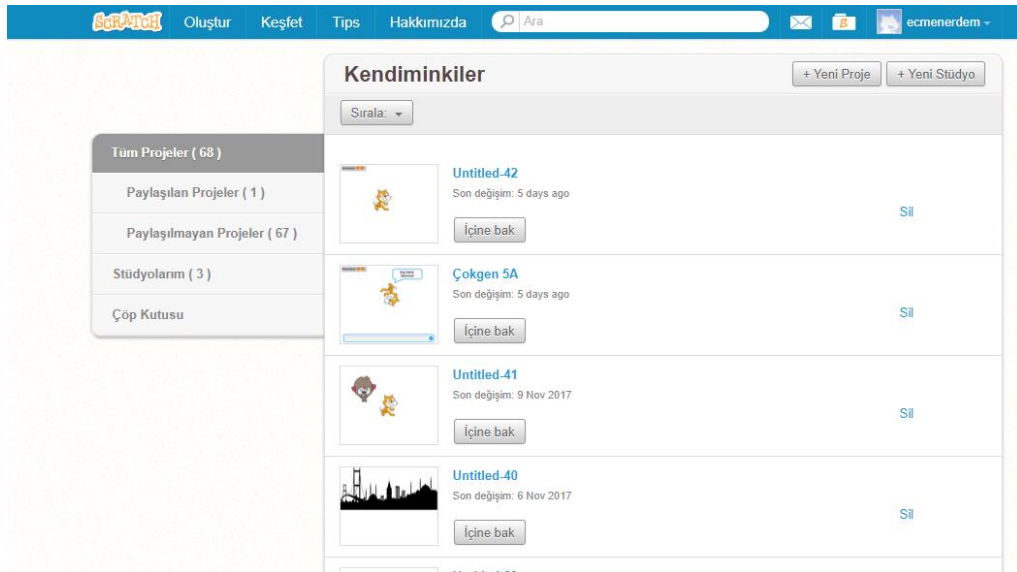
2.1.2.1. Scratch Programlama Ortamı ve Özellikleri

Scratch, öğrencilerin yaratıcı, sistematik biçimde düşüncelerine ve birlikte çalışmaya yardımcı olur. Scratch, MIT Media Lab'da geliştirilmiştir ve ücretsizdir (Lamb ve Johnson, 2011). Scratch blok şeklinde programlanan grafik, ses ve video destekli zengin bir dijital programlama aracıdır. Çocuklar, bu blokları istedikleri hareketleri ve etkileşimi sağlamak için ayrı tutulabilir veya birleştirilebilir. Scratch koordinatlar, açı ve uzunluk ölçümleri gibi geometrik ve ölçüm kavramlarını kullanır. Öğrencilerin bunları öğrenmelerini kolaylaştırır ve yaratıcı problem çözme, mantıksal akıl yürütme ve iş birliğini teşvik eder (Calder, 2010).

Scratch programı çevrimiçi ve çevrimdışı kullanılabilir. Çevrimdışı kullanımı için gerekli program Scratch'ın resmî sitesinden indirilmektedir. Bu araştırmada öğrenciler projelerini çevrimiçi program kullanarak yapmışlardır. Çevrimiçi programa <https://scratch.mit.edu/> adresinden giriş yapılmaktadır.



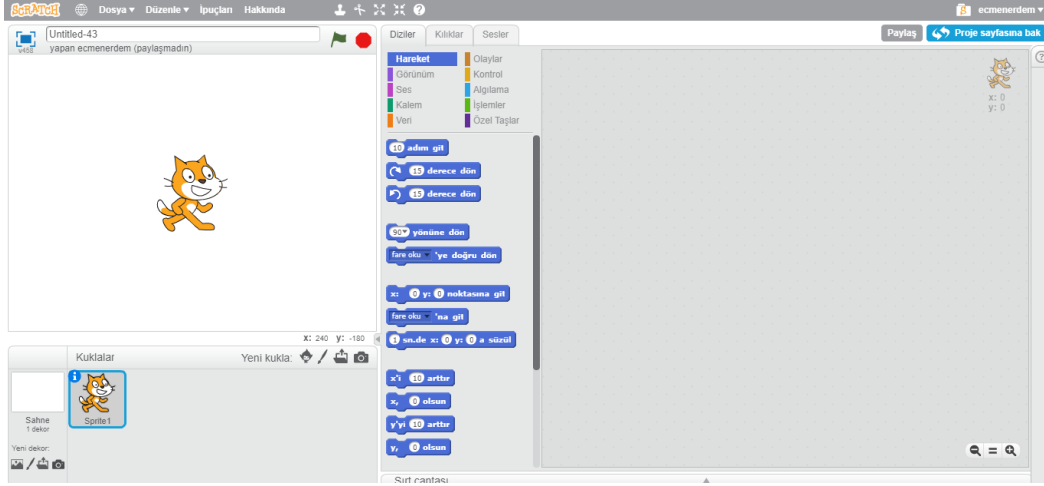
Şekil 1. Scratch Web Sitesinin Arayüzü



Şekil 2. Scratch Web Sitesinin Giriş Yaptıktan Sonraki Arayüzü

Sisteme ilk defa giriş yapıp üye olmak için Scratch'a Katıl linkine tıklayıp kayıt formunu doldurmaları gerekmektedir. Daha sonra sisteme giriş yaptıktan sonra "Kendiminkiler" butonuna basarak daha önce yapmış oldukları çalışmalara erişebilirler.

"Oluştur" butonuna basarak uygulamayı geliştirme ekranına geçerler.



Şekil 3. Scratch Web Sitesinde Uygulama Geliştirme Ekranı

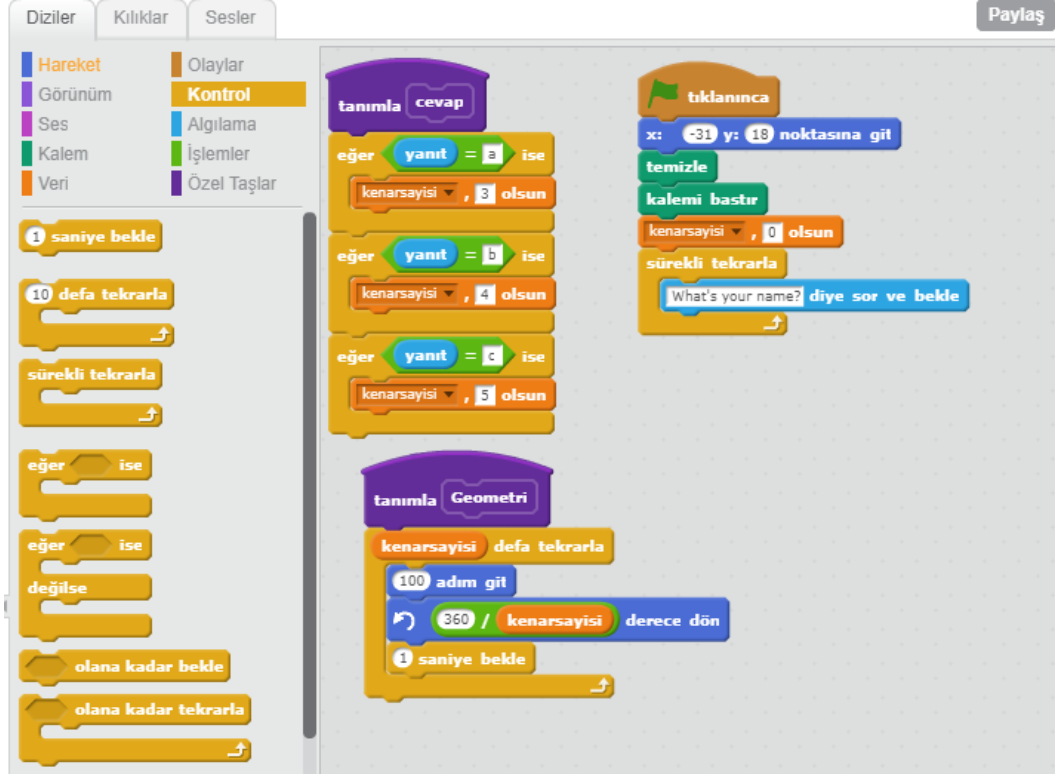
Şekil 3'te görüldüğü üzere uygulama geliştirme ekranında sol üst kısmında uygulamanın çalışır hali görülebilir. Sol alt kısımda uygulamada kullanılan karakterler ve sahneler bulunmaktadır. Bu nesnelerin ayarlamaları buradan yapılmaktadır. Orta bölümde uygulamayı geliştirecek kodlar bölümü bulunmaktadır. Sağ tarafta bulunan gri kısım ortadaki kodların sürüklenerek bırakıldığı, uygulamanın kod geliştirme ekranıdır.

Tablo 1. Scratch Çalışma Sayfasındaki Kod Blokları ve Açıklamaları

Diziler	Kılıklar	Sesler	
Hareket	Olaylar		Scratch'te uygulama geliştirirken kullanılan kodların, karakterin farklı görünümünün ve programda kullanılan seslerin bulunduğu bölümdür. Bu sekmelerden Diziler bölümünde uygulamayı geliştiren kullanılacak tüm kodlar yer almaktadır.
Görünüm	Kontrol		
Ses	Algılama		
Kalem	İşlemler		
Veri	Özel Taşlar		
Görünüm			Karakterin uygulamadaki görüntüsü ile ilgili işlemleri yapabilmek için gerekli kodların bulunduğu bölümdür.
Ses			Uygulama içerisinde kullanılan sesleri kontrol edebilmek için kullanılan kodların bulunduğu bölümdür.
Kalem			Uygulama içerisinde çizim animasyonu veya çizim hareketlerini kontrol edebilmek için kullanılan kodların bulunduğu bölümdür.

	Uygulamada kullanılan deęişkenlerin tanımlanıp deęerlerinin verildięi kodların bulunduęu bölümdür.
	Uygulamanın nasıl başlayacağı veya uygulama içerisinde hangi tuşların ne işe yarayacağının belirlendięi, fonksiyonların tanımlandığı kodların bulunduęu kısımdır.
	Uygulama içerisinde kullanılan döngüler, koşul ifadelerinin yazıldığı kodlar bu bölümde bulunmaktadır.
	Uygulamada karakterlerin birbirleri ile olan etkileşimlerini veya karakterin başka nesnelere olan etkileşimlerini kontrol etmek için kullanılan kodlar bulunmaktadır.
	Uygulama içerisinde kullanılan mantıksal ve matematiksel kontrolleri sağlayan kodların bulunduęu bölümdür.
	Uygulama içerisinde aynı kod bloklarını birden fazla kullanılması gerektiğinde bunları bir metot içerisinde alınmasını sağlayan kodların bulunduęu bölümdür.

Sahneye eklenen karakterin uygulama içerisindeki hareket etmesi, konuşması, koşula veya döngüye baęlı olarak herhangi bir işi gerçekleştirmek için orta kısımda bulunan kodları sürükleyerek ekranın saę tarafında bulunan bölüme sürükleyerek bırakmak gerekmektedir. Bu kod bloklarını birbirine baęlayarak program oluşturulmaktadır.



Şekil 4. Scratch Web Sitesinde Geliştirilen Örnek Uygulamanın Kodları

Şekil 4' te görüldüğü üzere ekranın sol tarafından bulunan kodlar sağ bölümde bulunan kod alanına sürüklenerek bırakılmaktadır. Bu uygulama kenar sayısı verilen bir geometrik şekli çizmek için oluşturulmuştur.

Araştırmalara göre; programlama eğitiminde öğrencilere herhangi bir programlama dilini öğretmek yerine programlama mantığını öğretebilecek örnekler verilmelidir (Gülmez, 2009). Yazılımda kullanılan görselliğin ve etkileşimin öğrenci başarısı üzerindeki etkisini olumlu olarak belirlenmiştir (Arabacıoğlu, Bülbül ve Filiz, 2007). Programlama eğitiminde görselliğin önemini vurgulamak amacı ile yapılan araştırma sonucunda, görsel özelliklerin öğrenci başarı ve motivasyonun da önemli etkisi olduğunu belirlemişlerdir. Konu ile ilgili yapılan diğer araştırmalar, yazılımlarda grafik ve animasyon kullanımının öğrencilerin derslere olan ilgisini artırmada daha etkili olduklarını göstermektedir (Gülmez, 2009).

Özcan, Ergün, Köse, Emir ve Gezgün (2017) yapmış oldukları çalışmada Scratch programının kullanımına yönelik öğrenci görüşlerini almışlardır. Öğrenci görüşleri incelendiğinde programlama eğitimine faydalı olduğu, programlama mantığının geliştiği, eğlenceli ve iyi olması, yaratıcılığı geliştirdiği, algoritmayı daha iyi anlamaya katkı sağladığı gibi görüşlerin bu çalışmadaki öğrenci görüşleri ile tutarlılık sağladığı görülmektedir. Çatlak, Tekdal ve Baz (2015) yapmış oldukları Scratch programlama öğretimi doküman inceleme çalışmasında Scratch'in kolay, ilginç, merak uyandırıcı ve keyifli bir uygulama olduğunu, algoritma ve programlama temellerini öğretiminde etkili

olduğunu yaratıcı düşünebilmeyi sağladığını, motivasyonu arttırdığı yönünde görüşler elde etmişlerdir.

Baytak ve Land (2011), 5. sınıf öğrencileri ile Scratch'te yapmış oldukları çalışmada programlama becerilerini oyun tasarımı ile geliştirmeyi amaçlamışlardır. Sonuçlar, öğrencilerin planlama, tasarım, test etme ve paylaşma gibi fonksiyonel oyunlar tasarlayabildiklerini göstermiştir. Öğrencilerin grafiksel programlama yazılımı kullanırken programlama kavramlarını daha iyi anladıklarını ve bilgisayar oyunları oluşturabileceğini belirtmiştir.

Genç ve Karakuş (2011), yapmış oldukları çalışmada eğitimde bilgisayar oyunları tasarımı dersinde öğrencilerin uygulama süresince daha aktif olmalarını, matematiksel ve bilgi işlemsel düşünme becerilerinin gelişimini sağlamak amacıyla görsel bir oyun, animasyon geliştirme programı olan Scratch kullanılmıştır. Öğrencilerin oyun tasarım sürecinde oyun geliştirme ve Scratch programını kullanmaları hakkındaki görüşleri alınmıştır. 109 öğrenci ile yapılan çalışmada öğrencilerin %79' u Scratch programını kolay bulduklarını, eğlenceli ve keyifli bir ortam sunduğunu, kullanımından hoşlandıklarını ifade etmişlerdir. Öğrencilerin %20' si Scratch programlamadan hoşlanmadıklarını, %73' lük bir kesim Scratch' in programlamanın temel yapılarının öğrenilmesinin birçok programlama dillerine göre daha kolay olduğunu ve Scratch'ın eğitimde kullanılabileceğini belirtmişlerdir.

Burke (2012), ortaokul öğrencileri ile yaptığı çalışmada öğrencilerin programlama becerilerini geliştirmek, genel okuryazarlık çalışmalarını dijital hikâye tasarımı ile farklılaştırmak ve öğrencilerin teknoloji okuryazarlığını artırmak için Scratch ile dijital hikâye ve oyun tasarımı etkinlikleri gerçekleştirmiştir. Araştırma sonuçlarına göre öğrencilerin Scratch etkinliklerinde döngü, karar verme gibi programlama yapılarını öğrendikleri, hikâye oluşturma ve eş zamanlama gibi dijital hikâye becerilerinin geliştiği sonuçlarına ulaşılmıştır. Fadjo (2012), ortaokul öğrencilerinde Scratch kullanımının bilgi işlemsel düşünme becerilerine etkisini incelemek için iki farklı çalışma yapmıştır. Birinci çalışmada 56 ortaokul öğrencisi Scratch ile dijital hikâye tasarımı yapmışlardır. İkinci çalışmada 78 ortaokul öğrencisi Scratch ile oyun tasarımı etkinlikleri yapmışlardır. Çalışma sonunda koşullu ifadeler, olaylar gibi programlama kavramlarını içeren bilgi işlemsel becerilerinin geliştiği görülmüştür. Scratch'in bilgi işlemsel düşünme becerisi geliştirmede etkili olduğunu vurgulanmıştır.

Begosso ve Da Silva (2013), Scratch programlama ortamında problem çözme becerileri ve mantıksal matematiksel düşünmeyi geliştirmeye odaklanma üzerine çalışma yapmışlardır. Yaşları 11-13 arası değişen öğrenciler ile yapılan bu çalışmada öğrenciler haftalık 300 saat olmak üzere 3 ay süre ile programlama ve problem çözme becerilerini geliştirmeye yönelik Scratch'te durum (if -else), döngüler (for, foreach, while), karşılaştırma operatörlerini kullanarak etkinlikler yapmışlardır. Ön test ve son test kullanılarak analiz edilen çalışmanın sonuçlarına göre öğrencilerde bu becerilerin geliştiğini belirtmişlerdir. Kukul ve Gökçearslan (2014), Scratch ile programlama eğitimi alan öğrencilerin problem çözme becerilerinin incelenmesi üzerine çalışma yapmışlardır. Çalışmanın amacı, ilk defa programlama dersi alan ortaokul öğrencilerinde verilen programlama eğitimi sonrasında öğrencilerin problem çözme becerileri incelenmişlerdir.

Tarama modelinin kullanıldığı çalışmaya, ortaokul 5. sınıf ve 6. sınıfta öğrenim gören 154'ü kız, 150 erkek olmak üzere toplam 304 öğrenci katılmıştır. Çalışmada öğrencilerin problem çözme becerilerini belirlemek amacıyla Serin, Bulut Serin ve Saygılı (2010) tarafından geliştirilen “Çocuklar için Problem Çözme Envanteri” kullanılmıştır. Araştırma sonuçlarına göre, cinsiyet, sınıf düzeyi ve bilgisayar sahipliği bakımından problem çözme becerilerinin anlamlı biçimde değişmediği görülmüştür.

Kalelioğlu ve Gülbahar (2014), 49 5.sınıf öğrenci ile Scratch'in problem çözme becerilerine olan etkisini ve 5. sınıf öğrencilerinin programlama hakkındaki düşünceleri incelemiştir. 5 hafta süren çalışmada öğrenciler Scratch ile çeşitli uygulamalar geliştirmişlerdir. Çalışmada öğrencilerin problem çözme becerilerini belirlemek amacıyla Serin, Bulut Serin ve Saygılı (2010) tarafından geliştirilen “Çocuklar için Problem Çözme Envanteri” kullanılmıştır. Nicel sonuçlara göre, Scratch platformundaki programlama, ilköğretim öğrencilerinin problem çözme becerilerinde önemli bir farklılığa neden olmadığını, öğrenci görüşlerinde Scratch ile program geliştirmeyi çok sevdiğini ve mutlu olduklarını belirtmişlerdir.

Tekerek ve Altan (2014), 6. sınıfta eğitim gören 60 öğrenci ile yaptığı çalışmasında Scratch' in algoritma öğretimine etkisini incelemiştir. Araştırmada öğrenciler deney ve kontrol grubu olmak üzere ikiye ayrılmıştır. Bir gruba oyun tasarımı ile algoritma öğretimi gerçekleştirilirken, diğer gruba ise düz anlatım yöntemiyle gerçekleştirmiştir. Çalışma sonunda her iki grupta bulunan öğrencilerin başarılarının arttığı görülmüştür. Son testler karşılaştırıldığında ise deney ve kontrol grubu arasında bir farklılık bulunmamıştır. Ayrıca cinsiyet değişkeni açısından kontrol ve deney gruplarının son test puanları arasında anlamlı bir farklılık bulunmamıştır.

Hsu (2014), ilköğretim öğrencileri ile yaptığı çalışmada Scratch ile oyun tasarımı yapan öğrencilerin programlama mantığını kavrama ve oyunlarında kullanma durumunu cinsiyete göre incelemiştir. 46 öğrenci ile yapılan çalışmanın sonunda programlama yapılarını kapsayan test uygulanmış ayrıca tasarladıkları oyunlar incelenmiştir. Analiz sonucunda kız ve erkek öğrencilerin programlama yapılarını anlamada farklılık olmadığı, döngü yapılarının kız öğrencilerin tarafından daha iyi anlaşıldığı ortaya çıkmıştır. Ayrıca oyun analizi sonucunda kız ve erkek öğrencilerin tasarladıkları oyunlarda blokları benzer oranda kullandıkları, oyunların içerdiği kostüm, sahne, karakter vb. yapıları kullanma açısından da cinsiyetin belirleyici bir özellik olmadığını sonucuna ulaşılmıştır.

Chiu (2014), Scratch ile programlama öğretiminin öğretmenlerin programlama becerisine etkisini incelemiştir. Bu çalışmada öğretmenler haftada üç saat olmak üzere Scratch ile oyun ve animasyon tasarımı yapmışlardır. Çalışma sonunda öğretmenlerin programlama becerileri ön test-son test puanları arasında bir farklılık bulunmamıştır. Öğretmenlerin görüşleri incelendiğinde Scratch ile programlama süreci programlama öğretimi için uygun bir araç olduğunu, eğlenceli ve sürükle bırak bloklar ile kolay öğrenilebilir olduğu belirtmişlerdir.

Nikou ve Economides (2014), lise öğrencileri ile yaptığı çalışmasında MIT tarafından geliştirilen Scratch ve mobil uygulama geliştirme platformu App Inventor programlama

araçlarının öğrencilerin programlama öğrenmelerinin etkisini incelemiştir. Çalışma sonunda her iki araçta programlamaya olan ilgilerinin artırdığı görülmüştür.

Shin ve Park (2014), yaptıkları çalışmada Scratch ile öğrencilerin problem çözme derecelerini ölçmüşlerdir. 6. sınıflardan 46 öğrenci ile yapılan çalışmada matematiksel mantık problemlerini Scratch ile çözme etkinlikleri yapılmıştır. Öğrencilerinin problem çözme becerilerini ölçmek için SoonsikSeo (2008) tarafından düzenlenen 34 soruyu test aracı olarak kullanmıştır. Araştırma sonuçlarına göre Scratch'in problem çözme becerine doğrudan etki ettiğini belirtmişlerdir.

Programlama öğretim yöntemleri alan yazında incelendiğinde yüz yüze eğitim, teknoloji destekli eğitim gibi farklı yöntemlerin kullanıldığı görülmüştür. Kullanılan bu yöntemlerde içeriğin verilme biçimi, sınıf motivasyonunu yüksek tutma, öğrencilerin yapılan derslere ilişkin görüşleri kullanılan yöntemlere göre değişkenlik göstermektedir. Son yıllarda teknolojinin eğitimde kullanılması ile birlikte teknoloji destekli öğretim yöntemi sık kullanılan yöntemlerden birisi olmuştur. Bu yöntem özellikle bireysel olarak çalışmak isteyen, kendi öğrenme hızına göre ve kendi kendine öğrenmeyi isteyen bireyler tarafından kullanılan bir yöntemdir.

2.1.3. Teknoloji Destekli Programlama Öğretimi

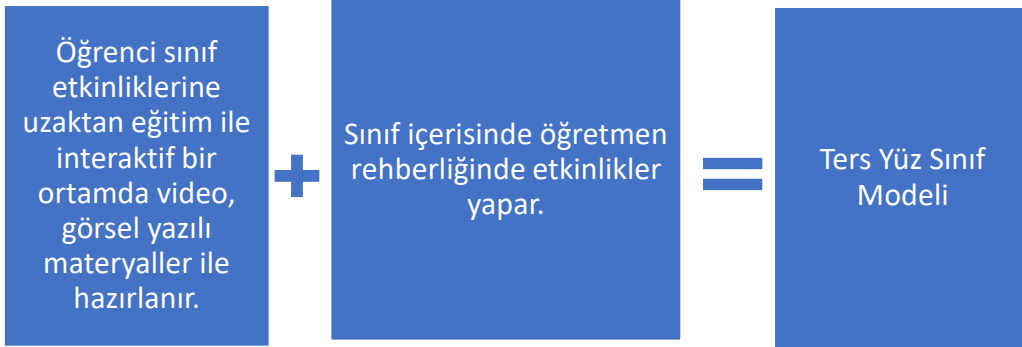
Günümüz teknolojilerinin hızla gelişmesi ve bilgiye erişimin daha hızlı olması ile birlikte eğitimde de teknolojik dönüşümler başlamıştır. Özellikle yapılandırmacı yaklaşımla beraber öğrenci merkezli eğitim modellerinin teknoloji destekli olarak yapılandırılması üzerine çeşitli çalışmalar ve araştırmalar yapılmaktadır. Bu yapılandırmada öğrencinin bilgiye kendi kendine ulaşım, kendi hızında öğrenmesi, bu bilgiye sınıf ortamı dışında erişebilmesi ve tüm bunların sistematik olarak bir öğretim programı olarak uygulanması gerekmektedir.

21. yy. da hazırlanacak eğitim programlarının öğrencilere yaratıcı, eleştirel çok yönlü ve bağımsız düşünmeyi, problem çözme becerilerini arttıracak, iletişim becerilerini daha etkin kullanmalarını sağlayacak, bilgiyi üretme, kendi kendine öğrenme yani öğrenmeyi öğrenme, özgüven, kendi farklılıklarını keşfedip yapabileceklerini öğrenme gibi birçok konuda rehberlik etmesi gerekmektedir (Özden, 2005). Bu becerilerden öğrenmeyi öğrenme Özer (2002) tarafından etkili öğrenmenin en önemli yöntem olduğunu belirtmiştir. Öğrenmeyi öğrenme, öğrencinin kendi öğrenmesinde yararlanabileceği farklı öğrenme stratejilerini kapsamaktadır. Etkili öğrenme becerisinin kazandırılmasında dört ana öğeden yararlanır. Bunlar etkin öğrenme, iş birliği yaparak öğrenme, öğrenme sorumluluğu edinme ve öğrenmeyi öğrenmedir. Öğrenme yöntemleri, öğrencinin kendi kendine öğrenmesini kolaylaştıran yöntemlerdir. Öğrenme yöntemleri ile bir öğrenci, öğrenmeyi kendisi planlayıp uygulayabilir ve öğrenmeyi kalıcı bir hale getirebilir (Özer, 2002).

Bu doğrultuda sınıf ortamı dışında bilgiyi öğrenip, derse hazır gelme, sınıf içinde yapılan etkinlikleri artırma ve daha çok uygulamaya dayalı bir eğitim gerçekleştirmek açısından yapılandırmacı yaklaşım çerçevesinde teknolojinin çok aktif olarak kullanıldığı 'Ters Yüz Sınıf Modeli' "uygulanması son yıllarda yaygınlaşmıştır (Kocabatmaz, 2016).

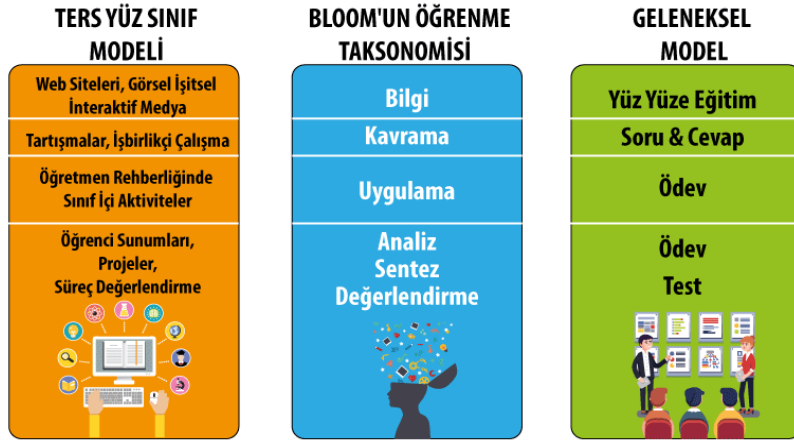
Öğrenmeyi öğrenmede kullanılabilir stratejilerden birisi harmanlanmış öğretim yaklaşımı olan ters yüz edilmiş sınıf (Flipped Classroom) modelidir. Harmanlanmış öğretim geleneksel anlayışla teknolojik fırsatların bir arada kullanıldığı yapılara denmektedir (Demiralay ve Karataş, 2014). Ters yüz sınıf modeli temel olarak evde ders okulda ödev imkanı sağlayan bir öğretim yaklaşımıdır.

Bu sistem geleneksel öğrenmenin aksine öğrencinin evde bilgiyi kendi başına öğrenip, okulda öğrendiklerini uygulama fırsatı sunan bir yöntem olarak tanımlanmaktadır (Zownorega, 2013). Ters yüz edilmiş sınıf modeli, önceden hazırlanmış video ve ders içerikleri ile bu derslere ait uygulamalar kullanılarak öğrencilerin aktif çalışmalarına olanak sağlar. Diğer yandan bu model, öğrenciye istediği bilgiye kendi planladığı yer ve zamanda ulaşmayı sağlayarak bireysel öğrenmeyi desteklemektedir (Talbert, 2012). Ters yüz sınıf modeli geleneksel öğretim yöntemlerinden farklı olarak, öğrenciler dersin kuramsal bölümünü çevrimiçi ortamlarda bulunan videolar, sunumla ve çoklu-ortam araçları ile fiziksel olarak bulunabileceği herhangi bir yerden öğrenmektedirler. Sınıf ortamında, konu ile ilgili hazırlanan uygulamalar ile edindikleri bilgileri paylaşma ve pekiştirme imkânı bulmaktadırlar (Gençer, Gürbulak ve Adıgüzel, 2014). Şekil 5’ te bu tanıma uygun grafiksel gösterim mevcuttur.



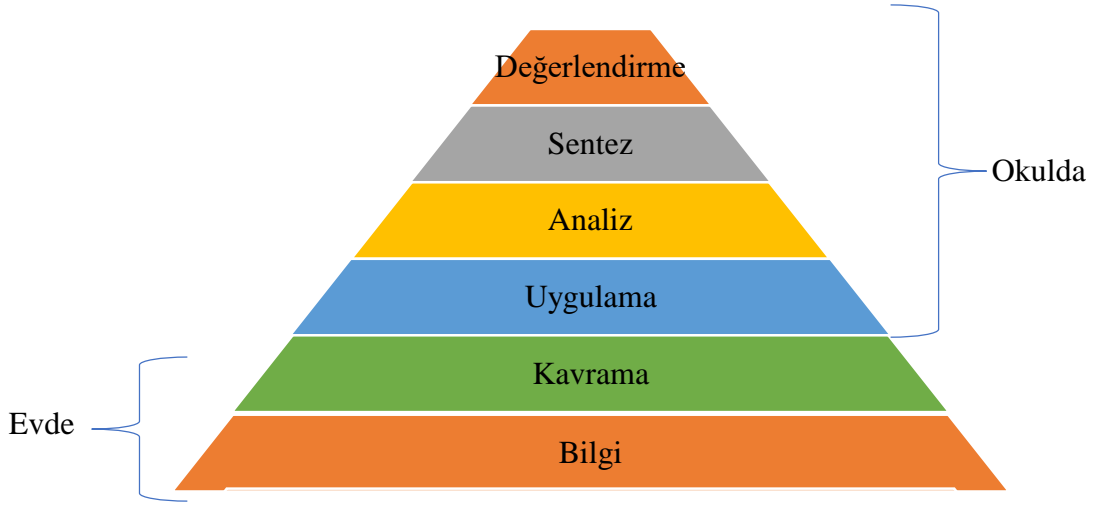
Şekil 5. Ters Yüz Sınıf Modeli

Öğrencilerin dersten önce bilgi edinme, anlama gibi bilişsel becerileri gerçekleştirmeleri ve ders sırasında akranları ve öğretmenlerin desteği ile uygulama, analiz, sentez ve değerlendirme gibi daha üst düzeydeki bilişsel işlere odaklanmalarını sağlayan ters yüz edilmiş sınıf modeli, Bloom Sınıflandırması’na uyum sağlamaktadır (Şekil 6).



Şekil 6. Bloom Taksonomisine Göre Geleneksel ve Ters Yüz Sınıf Modeli (Nederveld, 2014)

Verleger ve Bishop (2013) ters yüz sınıflar için sınıf içinde etkileşimli işbirlikçi aktiviteler, sınıf dışında bilgisayar temelli bireysel öğrenme aktivitelerden oluştuğunu ifade etmişlerdir. Aşağıdaki şekil Verleger ve Bishop' in (2013) ters yüz sınıf modelinin gösterimidir (Şekil 7).



Şekil 7. Ters Yüz Sınıf Modeli (Waterworth, 2016)

Ters yüz sınıf sisteminde avantajları ve dezavantajları üzerine yapılan tartışmalarda çeşitli görüşler ortaya atılmıştır. Bu yönetimin öğrenciler ve öğretmenler için bir çok avantajı mevcuttur. Öğrenciler açısından bu avantajları, öğrenci bireysel durumuna göre farklı sürelerde öğrenmeyi sağlaması, yorum yapma becerisini geliştirmesi, bilgiye önceden erişim ve öğrenmesine olanak sağlaması, bağımsız zamanlarda erişim sağlanması, ebevenlerin öğrencileri takip etmesi ve gerektiği yerlerde destek verebilmesi. Öğretmenler açısından avantajları sınıf içinde rehberlik yapmasını sağlama, sınıf içi etkinliklerde öğrencilere daha fazla yardımcı olabilme, öğrencilerin etkileşimli eğitim yapmasından dolayı sınıf yönetimi sorunlarının azalması, çeşitli büyüklükte gruplar ile

çalışılabilmesi, konu anlatımında oluşabilecek zaman kaybının en aza inmesi, öğrenci ile iletişimin gelişmesine olanak sağlaması, ders materyali hazırlarken işbirlikçi çalışmaya olanak salamasıdır. (Miller, 2012; Gencer, Gürbulak ve Adıgüzel, 2014).

Miller (2012) etkili bir ters yüz sınıf yaratmak için bazı süreçlere dikkat edilmesi gerektiğini ifade etmiştir. Bunlar;

- **Doğru İçerik:** Hazırlanacak içeriğin ihtiyaçlarının nasıl yapıldığının belirlenmesi gerektiğinin belirlenmesi gerekmektedir. Doğru analiz edilmeden hazırlanan içeriğin öğrencinin ilgisini çekmeyebilir.
- **Etkileyici Modeller:** Hazırlanan içeriklerin derslerde en iyi şekilde kullanılabilmesi için bazı pedagojik modeller kullanılmalıdır. Bunlar;
 - Proje tabanlı öğrenme
 - Oyun tabanlı öğrenme
 - Problem tabanlı öğrenme
- **Teknoloji:** Öğretmenin bu hazırlanan içerikleri hangi teknolojiler ile anlatacağı önemli bir durumdur. Öğretmen bu içerikleri hangi teknoloji ile anlatılırsa daha etkili olacağını belirlemesi gerekmektedir. Burada teknik olarak kullanılacak cihazlar, dijital olarak tasarlanan öğrenme nesnelerinin formatları ve cihazlarla uyumluluğu gibi birçok konuda şartların sağlanmış olması gerekmektedir.
- **Yansıtma:** Öğrenciler öğrendikleri bilgilerin farkında olmaları ve bilgiyi transfer edebilmeleri gerekmektedir. Örneğin öğrenciler bir video izlediğinde, tıpkı herhangi bir öğretim faaliyetinde olduğu gibi öğrencilerin öğrendiklerini, onlara nasıl yardımcı olacağını, alaka düzeyi ve daha fazlasını düşünmesini sağlamak için yansıtıcı etkinlikler yapılmalıdır.
- **Zaman ve Mekân:** Öğrencilerin hepsi ödevi olarak verilen dijital materyalleri ev ortamında yapmalarını beklemek zor olabilir. Videolardaki öğrenmeyi öğrenciler için yönetilebilir olmalıdır. Bu, öğrenmeyi sağlamak için biçimsel olarak değerlendirilmesine yardımcı olur ve öğrenciler için uygulanabilir olur.

Bu araştırmada ters yüz sınıf modeli ile eğitim seçilmesinin temel sebebi geleneksel öğretim modelinin aksine bireyin kendi kendine, teknolojiyi kullanarak öğrenmeyi gerçekleştirip, bireyin bu yöntemi kullanarak bilgi ve becerilerde geleneksel yöntemle göre farklılık olup olmadığını değerlendirip, bu süreci gözlemlemektir. Bu alanda yapılan çalışmalar incelendiğinde bu yöntemin öğrenci başarısına olumlu etkisi olduğu görülmüştür. Zappe, Leicht, Messner, Litzinger ve Lee (2009) yaptıkları çalışmada inşaat mühendisliğinde okuyan öğrencilerle ters yüz sınıf modelini kullanmıştır. Aktif öğrenmeyi daha etkin kılmak için ders içerikleri sınıfta değil, çevrim içi video ile gönderilmiştir. Bu yöntem ile öğrencilerin öğrenmeleri daha aktif ve problem çözme becerilerinin daha iyileştiğini belirtmiştir.

Dill (2012), çalışmasında Master of Education programında bir proje kapsamında Ashland Ortaokulu'nda (MA) iki yedinci sınıfın performansını karşılaştırmıştır. Dill, iki hafta boyunca bir sınıfı geleneksel yöntemlerle diğer sınıfta ise ters yüz sınıf yöntemi uygulamıştır. Ters yüz modeli uygulanan sınıfta öğrenciler ders dışında videolar izlemişler, sınıfta ise proje tabanlı çalışmalar ve çalışma kitabı üzerinden etkinlikler yapmışlardır. Ters yüz sınıf modeli uygulanan sınıfın bu çalışmaya başlamadan önceki ödev tamamlama yüzdesi %79 iken iki hafta sonra bu yöntem ile ödev tamamlama yüzdesinin %98.7 olduğunu belirtmiştir. Geleneksel sınıf modeli ile ders yapan öğrencilerin ödev tamamlama yüzdesinin %81.4 olduğunu belirtmiştir. Dill öğrencilerin Fransızca öğretimi performanslarını ters yüz sınıf modeli uygulamadan önce ve sonra olmak üzere dilbilgisi sınavları, yazılı değerlendirme ile yapmıştır. Geleneksel sınıf modeli uygulanan sınıfta uygulama öncesi yazılı değerlendirme sonuçlarının %89.7 olan başarısının uygulama sonrası %87 olduğunu, dil bilgisi uygulamalarında uygulama öncesi %75 olan başarının uygulama sonrası %76 olduğunu belirtmiştir. Ters yüz modeli uygulanan sınıfta uygulama öncesi dil bilgisi uygulamalardaki başarı %78 iken uygulama sonrası %88 olduğunu, yazılı uygulamalarda ise uygulama öncesi %87.3 olan başarının uygulama sonrası %92 olduğunu belirtmiştir.

Ekmekçi (2014), İngiliz Dili Eğitimi hazırlık sınıfı öğrencileriyle 2013-2014 akademik yılı güz döneminde yaptığı çalışmada bir gruba ters yüz sınıf modeli uygulanmış diğer gruba geleneksel öğretim yöntemi uygulamıştır. 23 öğrenciden oluşan deney grubundaki öğrencilere; öğretmenin paylaştığı linkleri görebilmeleri ve ders gereksinimlerini takip edebilmeleri amacıyla Edmodo kullanılmış ve bir dönem boyunca, uygulanmıştır. 20 öğrenciden oluşan kontrol grubuna geleneksel ders yöntemi kullanılmıştır. Her iki grupta da daha önce seçilmiş ders kitabına uygun olarak araştırmacı tarafından hazırlanmış aynı izlenice takip edilmiştir. Veri toplama araçları olarak araştırmada; ön ve son testler, tartışmacı paragraf puanlama rubriği, bilgi ve iletişim teknolojileri okuryazarlığı anketi ve tersten yapılandırılmış yazma sınıfı tutum ölçeği kullanılmıştır. Araştırma sonuçlarına göre, Deney grubu öğrencileri uygulamadan sonra kontrol grubundaki öğrencilerden daha iyi performans sergilemişlerdir.

Albert ve Beatty (2014), yaptıkları çalışmalarında derste ters yüz sınıf yönteminin sınıf içi etkisini değerlendirmişlerdir. 2012 güz döneminde geleneksel olarak yapılan ders 2013 güz döneminde ters yüz sınıf metodu ile yapılmıştır. Derslerde aynı içerikler aynı değerlendirmeler, aynı kitaplar, aynı öğretim elemanı tarafından verilmiştir. Sonuç olarak ters yüz sınıf metodu ile ders yapılan sınıfın başarı seviyesi bir önceki dönem aynı dersi geleneksel yöntem ile alan sınıfa göre daha başarılı çıkmıştır.

Walsh (2015), yapmış olduğu araştırmasında, 2014 yılında, bu öğretim metodolojisini değerlendirmek amacıyla, bir üniversitede 6 ters yüz sınıf öğretim teknikleri kullanılmıştır. Nitel değerlendirme için bir anket aracı kullanılmıştır ve öğrencilerin büyük bir çoğunluğu olumlu yanıt vermiştir. Öğrencilerin %85'i bu yaklaşımı sevdiğini ve %76'sı bu yaklaşımın kendilerinin öğrenmesine daha iyi yardımcı olduğunu belirtmiştir.

Zhang, Du, Yuan ve Zhang (2016), A ve B sınıflarının her birinde 32 öğrenci olma üzere toplam 64 öğrenci ile ters yüz sınıf yöntemi ile İngilizce telaffuz derslerinin etkinliğini

araştırmıştır. Bir sınıfa geleneksel yöntemler ile diğer sınıfa ters yüz sınıf yöntemi ile ders işlemiştir. Bu çalışmada öğrencilerin final sınav puanları geleneksel öğretim yöntemi ile öğretim yapılan öğrencilerin puanlarıyla karşılaştırmıştır ve öğrencilerin öğretim metotlarına yönelik tutumlarını araştırmak için bire bir, yüz yüze görüşmeler yapılmıştır. Sonuç olarak, ters yüz sınıf metodunun telaffuz öğretiminde geleneksel metottan daha etkili olduğu sonucuna varılmıştır.

Programlama ve bilgi işlemsel düşünme, teknoloji kullanma becerilerinin ters yüz sınıf modeli ile kazandırılması üzerine çalışmalarda mevcuttur. Karaca (2016), programlama ve algoritma dersini bir gruba ters yüz sınıf modeli, diğer gruba geleneksel öğretim yöntemini kullanarak hangi yöntem ile programlama ve algoritma öğretiminin daha iyi olacağını belirlemeyi amaçlamıştır. Araştırmada 80'i kontrol, 80'i deney grubu olmak üzere toplam 160 öğrenci ile yapılmıştır. Çalışma sonucunda elde edilen bulgulara göre, ters yüz sınıf modeli ile öğrenim gören öğrencilerin daha başarılı olduğunu ve ters yüz sınıf modelinin programlama ve algoritma öğretiminde kullanılabilir bir yöntem olduğunu belirtmiştir.

Çakır (2017), ters yüz sınıf uygulamalarının fen bilimleri 7. sınıf öğrencilerinin akademik başarı, zihinsel risk alma ve bilgisayarca düşünme becerilerine etkisi üzerine yapmış olduğu çalışmada ön test, son test yarı deneysel desen kullanılmış; bu kapsamda deney ve kontrol gruplarıyla çalışılmıştır. Araştırmanın çalışma grubunu ortaokul seviyesinde öğrenim gören 26 deney grubu, 27 kontrol grubu olmak üzere toplam 7. sınıf öğrencisi oluşturmuştur. Deney grubu öğrencileriyle ters yüz sınıf uygulamaları ile ders işlenirken; kontrol grubu öğrencileriyle ise okullarda kullanılan mevcut programın yürütüldüğü biçimde ders işlenmiştir. Veriler, akademik başarı testi, Bilgi İşlemsel Düşünme ölçeği ve Zihinsel Risk Alma ve Yordayıcılarına Yönelik Algı Ölçeği kullanılarak toplanmıştır. Araştırma sonuçlarına göre ters yüz sınıf uygulamalarının öğrencilerin akademik başarılarına olumlu etkisi olduğunu göstermiştir. Öğrencilerin öğrenmelerinin kalıcılığı bakımından ters yüz modeli ile ders yapılan grubun lehine bir farklılık çıkmıştır. Ters yüz sınıf uygulamalarının öğrencilerin bilgi işlemsel düşünme becerileri üzerine etkisi incelendiğinde bir artış olduğu görülmekle birlikte kontrol grubunda bir farklılık olmadığını belirtmiştir.

Ters yüz sınıf modeli ile yapılan eğitimlerde verilmek istenen kazanım ve buna bağlı olarak öğrenci başarı puanlarında artış olduğunu belirten birçok araştırma mevcuttur. Programlama öğretiminde kullanılan yöntemlere ilişkin çalışmalar genellikle yükseköğretim de okuyan öğrenciler üzerine olmuştur. Scratch ile çocukların örneklem olarak kullanıldığı birçok çalışma olsa da çocuklarda programlama eğitiminin farklı yöntemler ile verilmesinde öğrencilerin programlama ve bilgi işlemsel düşünme becerilerine etkisi üzerine çok az sayıda çalışma olduğu görülmüştür. Diğer bir ifade ile bu araştırmanın temel parametreleri olan yöntem ve programlama kavramlarının kazandırılmasının bir arada kullanıldığı çok fazla çalışma bulunmamaktadır. Bu çalışma yüz yüze eğitim modeli ve ters yüz sınıf modeli ile verilen blok tabanlı programlama eğitiminde hangi yöntemin 5. sınıf öğrencilerine uygun olduğu, bilgi ve beceri kazandırmada hangisinin daha iyi olacağı, kullanılan yöntemlerin avantajları ve

dezavantajlarını görmek adına araştırılacak olmasından dolayı, bu alanda yapılacak arařtırmalara ileride ışık tutacağı düşünölmektedir.

2.1.3.1. Arařtırmada Kullanılan ÖYS Hakkında Bilgiler

Ters yüz sınıf modelini desteklemek için bu arařtırmada Biliřim Teknolojileri ve Yazılım dersinde kullanılan bir özel bir ÖYS olan Biliřim Garaj Akademisi sistemi kullanılmıřtır. Sistem anasınıfından lise son sınıfa kadar birçok biliřim konusunun ders içerikleri mevcuttur. Bu konular genel olarak kodlama, 3 boyutlu tasarım, Arduino ile robotik tasarım gibi konuları içermektedir. Sistem genel olarak řu özelliklere sahiptir.

- Öğrenci kayıt
- Öğrenci profil düzenleme
- Öğrenciyi gruba ve řubeye atama
- Ders Açma ve Kapatma
- Öğrencinin kayıtlı olduđu gruba ders atama
- Öğrencinin sistemi kullanım istatistikleri
- Duyurular
- Ajanda
- Not işlemleri
- Kütüphane ve kaynak ekleme
- Konu anlatımı ve bağımsız etkinlikler



Őekil 8. Arařtırmada Kullanılan ÖYS'ye Ait Ekran Görüntüsü

Őekil 8'de göröldüđu üzere Havai Fiřek konulu eğitim ekranı yer almaktadır. Öğrenci Bařlamak için tııklayın butonuna basarak bir sonraki sayfa olan kazanımlar sayfasına yönlendirilmektedir. Öğrenci kendi öğrenme sürecini bu yazılım aracılıđı ile yönetmektedir.



Şekil 9. Araştırmada Kullanılan ÖYS'ye Ait Ekran Görüntüsü

Şekil 9’da ilgili konuya ait kazanımlar belirtilmiştir. Öğrenci bu etkinliği yapmadan önce ilgili kazanımları görüp konu bitiminde hangi kazanımları öğreneceği konusunda bilgi sahibi olmaktadır. Sahnede ileri ve geri butonu ayrıca bu etkinlikte kullanılacak kaynaklara erişim için bir buton bulunmaktadır. Ayrıca sahnede kullanıcının kaçınıcı sayfada olduğunu gösteren bir bölüm bulunmaktadır.



Şekil 10. Araştırmada Kullanılan ÖYS'ye Ait Ekran Görüntüsü

Şekil 10’da etkinliğin bitiminde öğrencilerden yapmaları beklenen ürünün bir örneği bulunmaktadır. Burada öğrenciler temel olarak bu ekrandaki gördükleri animasyonu tamamlayıp, daha sonra kendi isteklerine göre biçimlendirip ve daha fazla özellik ekleyebilecekleri belirtilmiştir.



Şekil 11. Araştırmada Kullanılan ÖYS'ye Ait Ekran Görüntüsü

Şekil 11’ de yapmaları beklenen oyun veya animasyonun nasıl geliştireceklerine dair adım adım kodlar verilmiştir. Öğrenciler kullanılan kodların hangi anlamlara geldiğini ve geliştirme esnasında dikkat etmeleri gereken noktalar kırmızı numaralı butonlara basarak öğrenebilmektedirler.

2.2. Programlama Öğretiminde Değerlendirme

Programlama dili öğretiminde, öğrencilerin programladığı projelerin ve algoritmaların değerlendirilmesinde eğitimcilerin işini kolaylaştıran birçok yöntem ve araç bulunmaktadır. Bilgi işlemsel düşünme, problem çözme, soyutlama becerilerini değerlendirmede performans değerlendirme, sorun çözme etkinlikleri, görev tabanlı değerlendirme, problem kurma tabanlı değerlendirme, tasarım senaryoları gibi birçok değerlendirme yöntemi bulunmaktadır (Yeni, 2017).

Wilson, Hainey ve Connolly (2012) görsel programlar için 3 aşamalı değerlendirme yapılması gerektiğini belirtmişlerdir. Bunlardan ilki programlama kavramları (iş parçacıkları, koşullar, değişkenler vb.), ikinci değerlendirme kod organizasyonu (değişkenlerin isimlendirilmesi, kullanılan nesnelerin isimlendirilmesi) son değerlendirme ise ara yüz ve kullanılabilirliktir (işlevsellik, özgünlük vb.) Tseng ve Weng (2009) öğrenciler tarafından geliştirilen programların değerlendirilmesinde 3 farklı

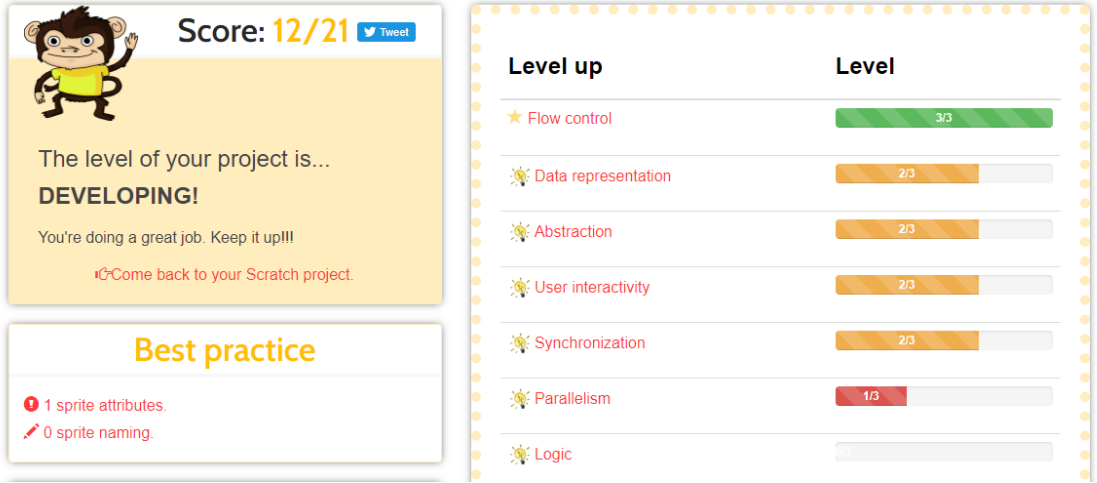
değerlendirme yapılması gerektiğini belirtmişlerdir. Bunlar Mantık (Programın senaryosu), kod (sahne, karakter, olaylar, metotlar, programın akışı), kaynaklar (kullanılan nesnelere, grafikler, sesler). Bu kriterler kullanılarak bir sistem yardımı ile otomatik olarak programın içerisindeki kodlar analiz edilip, çeşitli geri bildirimler verilebilmektedir. Bu tip araçlar otomatik değerlendirme araçları olarak isimlendirilmektedir.

2.2.1. Otomatik Değerlendirme Ortamları

Otomatik değerlendirme ortamları, eğitimcilerin daha hızlı puanlama yapmasına yardımcı olmak için oluşturulmuş sistemlerdir. Dreher, Reiners ve Dreher (2011) eğitimcilerin el yordamı ile yaptığı değerlendirmelerin kendilerine bağlı olduğunu, geniş içerikli programları değerlendirirken aldıkları sorumluluklardan dolayı ve baskı hissedecekleri, bununda öğrencilere verdikleri geribildirim yetersiz olabileceğini ve bu ortamlar değerlendirme sürecini otomatikleştirerek tüm paydaşlara fayda sağlayacağını belirtmiştir. Bu görüşe katkı olarak Sant (2009), otomatik programlama araçlarını programlama görevlerinin otomatik değerlendiren ve herhangi bir özel altyapı gerektirmeyen, ayrıntılı geri bildirim veren, kalabalık gruplarda öğrencilerin puanlarını daha kısa sürede alabilmelerini sağlayan bir sistem olarak belirtmiştir.

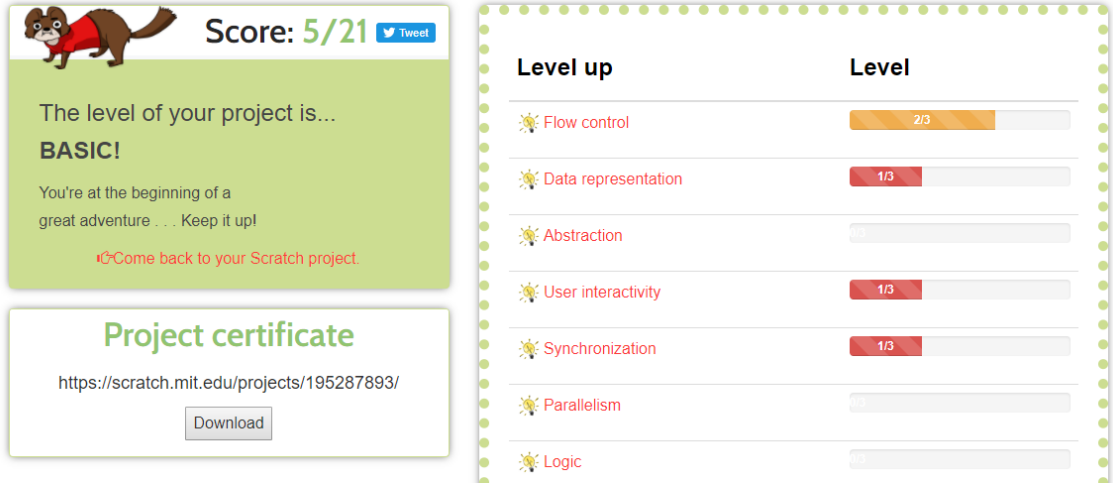
Vihavainen, Vikberg, Luukkainen, ve Pärtel (2013) otomatik bir değerlendirme aracı olan TMC'nin (Test My Code) bir üniversitesinde kullanılmasının faydalarını da bildirmiştir. Bu araç yazılan programın kodlarını inceleyerek öğretmen ve öğrenciye çift yönlü bir geri bildirimde bulunmaktadır. Koh ve diğerleri (2014), ABD' de öğretmenlerin gerçek zamanlı olarak öğrencilerin bilgi işlemsel düşünme becerilerini değerlendirmek ve öğrencilerden hangilerinin yardıma ihtiyacı olanları gözlemlenmelerine yardımcı olmak için otomatik bir araç (REACT) geliştirmiştir. Bu çalışmada da kullanılan web tabanlı diğer bir otomatik değerlendirme aracı olan Dr. Scratch, Scratch ile oluşturulan projeler otomatik olarak, belirlenen kriterler üzerinden değerlendirilebilir.

Dr. Scratch herkesin erişimine açık herhangi bir ücretli üyelik gerektirmeyen Scratch projelerini çeşitli alt boyutlar ile inceleyip kullanıcıya geliştirmiş olduğu uygulama hakkında puan ve bilgi veren bir platformdur. Bu sisteme <http://www.drscratch.org> adresi üzerinden ulaşılabilir. Proje değerlendirmede alt boyutlar mantık, veri gösterimi, uygulamada koşul sunma, uygulamada zaman uyumu, kullanıcı etkileşimi, akış denetimi, soyutlama, yinelenmiş komutlar, yanlış isimlendirme, kullanılmayan kodlardır. Bu kriterlere göre bir proje 21 puan üzerinden değerlendirilir. Toplam puan ne kadar yüksek ise projenin iyi ve belirtilen kriterlere göre hazırlanmış olduğu anlamına gelmektedir. Öğrenciler tarafından hazırlanan çalışmaların Dr. Scratch tarafından değerlendirilmesi sonucu ortaya çıkan raporlardan bazı örnekler aşağıda yer almaktadır.



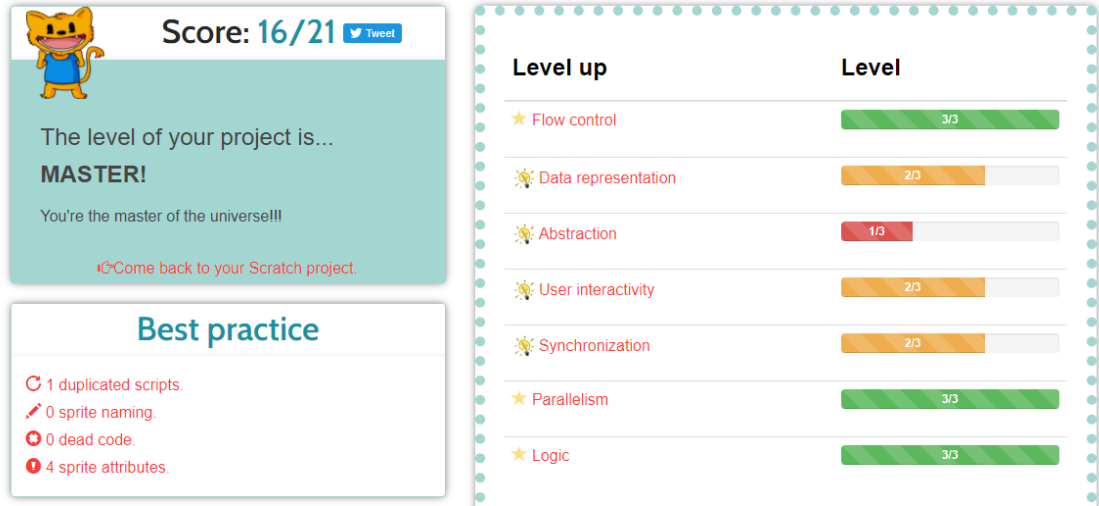
Şekil 12. Dr. Scratch ile Proje Değerlendirme Raporu

Şekil 12’de Scratch ile geliştirilmiş örnek bir projenin bilgi işlemsel düşünme becerileri açısından kodlarının Dr. Scratch tarafından değerlendirme raporu görülmektedir. Bu rapora göre öğrenci tarafından yapılan proje 21 puan üzerinden 12 puan almıştır. Proje geliştirilmeli kategorisinde yer almaktadır. Bu projede akış kontrolü kriterinden tam puan almıştır. Aynı buton veya nesneye tıklanıldığında birden fazla komutu çalıştırma kriterinden 1 puan almıştır. Mantık kriterinden hiç puan alamamıştır. Diğer kriterlerden 2 puan almıştır.



Şekil 13. Dr. Scratch ile Proje Değerlendirme Raporu

Şekil 13’de Scratch ile geliştirilmiş örnek bir projenin bilgi işlemsel düşünme becerileri açısından kodlarının Dr. Scratch tarafından değerlendirme raporu görülmektedir. Öğrenci tarafından hazırlanmış proje 21 puan üzerinden 5 puan almıştır. Proje basit kategorisinde yer almaktadır. Bu projede akış kontrolü kriterinden 2 almıştır. Aynı buton veya nesneye tıklanıldığında birden fazla komutu çalıştırma, mantık, soyutlama kriterlerinden hiç puan alamamıştır. Diğer kriterlerden 2 puan almıştır.



Şekil 14. Dr. Scratch ile Proje Değerlendirme Raporu

Şekil 14’de Scratch ile geliştirilmiş örnek bir projenin bilgi işlemsel düşünme becerileri açısından kodlarının Dr. Scratch tarafından değerlendirme raporu görülmektedir. Öğrenci tarafından hazırlanmış proje 21 puan üzerinden 16 puan almıştır. Proje uzman kategorisinde yer almaktadır. Bu projede akış kontrolü, aynı buton veya nesneye tıkladığında birden fazla komutu çalıştırma, mantık, kriterlerinden 3 puan almıştır. Veri gösterimi, kullanıcı etkileşimi, zaman uyumu kriterlerinden 2 puan almıştır. Soyutlama kriterinden 1 puan almıştır.

Dr. Scratch’in kullanımına yönelik alan yazında birkaç çalışmaya ulaşılmıştır. Bu çalışmalarda temel olarak Scratch ile geliştirilen projeler analiz edilip sonuçlar üzerinden bulgular elde edilmiştir. Moreno, Robles, ve Román (2015) yaptıkları çalışmada Dr. Scratch’in Scratch programlamada yanlış kodlamayı veya önemli olmayan nesne isimlerini, kodun tekrarı, hiçbir zaman çalıştırılmamış kod ve nesne özniteliklerinin yanlış başlatılması gibi olası hataları algıladığını belirtmişlerdir. Oluk, Korkmaz ve Oluk (2018), yapmış oldukları çalışmada Scratch kullanımının algoritma geliştirme ve bilgi-ışlemsel düşünme becerilerini geliştirmede etkisini incelemişlerdir. Öğrencilere bilgi-ışlemsel düşünme ölçeği ve algoritma geliştirme başarı testi uygulanmıştır. Öğrenci projelerini Dr. Scratch ile analiz ettiklerini belirtmişlerdir.

2.2.2. Diğer Değerlendirme Araçları

Programları değerlendirmede otomatik olmayan farklı yöntemler ile projelerin ve programların içeriğini ve algoritmasını değerlendirmek mümkündür. Burada kullanılan yöntem ve araçlar kodların kullanıp kullanılmadığını, hata ayıklama gibi programın teknik ayrıntılarından ziyade geliştirilen ürünün kalitesi ve bu ürün geliştirilirken bireylerin kazanmış oldukları bilgi işlemsel düşünme, algoritma oluşturma gibi bireysel becerileri değerlendirmeyi amaçlamaktadır. Bu amaç doğrultusunda dereceleme ölçüt listeleri, testler, öz değerlendirme ve gözlem formları gibi araçlar ile öğrenci projeleri ve çalışma süreçleri değerlendirilebilir. Yükseltürk ve Altıok (2016) yapmış oldukları çalışmada katılımcılara yüz yüze eğitim ile Scratch dili öğretilmiş ve Scratch ortamında eğitsel oyun projeleri geliştirmelerini istemişlerdir. Bu projeleri konusu, hedef kitlesi ve

uygulamanın sahip olması gereken özelliklerine göre belirlenen değerlendirme puanları açısından sınıflandırmışlardır. Ayrıca 30 dakikalık bir uygulama sınavı yapmışlardır. Katılımcılar uygulamanın analizi ve tasarımı ile ilgili rapor hazırlamaları istenmiştir. Dersteki başarı durumlarına göre katılımcıların içerisinde bir grup seçilerek Scratch programlamaya ilişkin deneyimleri, geliştirdikleri uygulamalar hakkındaki düşüncelerini öğrenmek adına odak grup görüşmesi yapmışlardır.

Diğer bir değerlendirme yaklaşımı olarak Bilge Kunduz etkinliği kapsamındaki sorular tercih edilebilir. Bu etkinlik bilgisayar bilimini ve bilgi işlemsel düşünmeyi öğrencilere öğretmek amacı ile oluşturulmuş, aynı zamanda eğlendirmeyi de önemseyen uluslararası bir etkinliktir. Bu etkinlik, birçok ülke aynı dönemde öğrencilerin enformatik becerilerini test eden çevrimiçi bir etkinliktir (Kalelioğlu ve Gülbahar, 2016). İlk olarak 2004 yılında Litvanya’da 779 öğrenci ile düzenlenen etkinlik yıllar içerisinde birçok ülkede uygulanmaya başlanmıştır. 2014 verilerine göre 33 ülkeden 900.000 ‘den fazla öğrenci bu etkinliğe katılmıştır. Günümüzde 50 ülkenin bilgisayar uzmanları ile sorular belirlenip etkinlik yapılmaktadır.

Etkinliğin uluslararası uygulaması şu şekildedir.

- Karışık zorluk dereceli problemlerden oluşturulan 18 soruluk etkinliği çözmek için öğrencilere 45 dakika verilir.
- Zorluk derecelerine göre puanlamalar;
 - Kolay derecede 6 soru her soru 6 puan,
 - Orta derecede 6 soru her soru 9 puan,
 - Zor derecede 6 soru her soru 12 puan.
- Yanlış olan her soru için zorluk derecesine göre o sorunun doğru puanının 3’te biri alınarak toplam puandan düşülür. Örneğin Kolay derecedeki herhangi bir soruyu yanlış yapan öğrencinin toplam puanından 2 puan düşülür.

Gülbahar ve Kalelioğlu (2014), yapmış oldukları çalışmada Bilge Kunduz Uluslararası Enformatik Yarışması 2014 yılında ilk kez pilot olarak Türkiye’de düzenlendiğini belirtmiştir. 12 ilden 57 okul katılmıştır. Başvuran öğrencilerden 1788 kişi yarışmayı başarıyla tamamlamıştır. Yarışmaya katılan toplam 1788 5. ve 6. sınıf öğrencilerin aldıkları puanlar 0 ve 129 arasında değişmektedir. Puanların ortalaması 65,01 bulunmuştur. Tüm soruları doğru yapan öğrenci olmamıştır. Öğrencilerin yarışmada kalma süreleri 2 dakika ile 45 dakika arasında değişmekte olduğunu belirtmişlerdir. Yarışmaya ilişkin görüşler incelendiğinde son derece yararlı bir yarışma olduğunu, Bilişim Teknolojileri dersinin tanıtımı ve bu kapsamda algoritma eğitiminin gerekliliğini göstermesi açısından önemli olduğunu belirtilmiştir.

Gülbahar ve Kalelioğlu (2015), hazırladıkları Bilge Kunduz Uluslararası Enformatik Yarışması 2015 yılı raporunda 257 okulun ve 13.784 öğrencinin katıldığını belirtmişlerdir. Öğrencilerin aldıkları puanlar -45 ve 135 arasında değişmekte olduğunu, puanların ortalaması 33,49 olarak bulunduğunu ve 162 öğrenci tüm soruları doğru yanıtladığını belirtmişlerdir. Yarışmaya ilişkin görüşler incelendiğinde düşünme becerilerini harekete geçiren bir etkinlik olduğunu, enformatik konusunda bilinçlendiren, eğlenceli, yönlendirici ve farklı bir etkinlik olduğunu ifade etmişlerdir.

Yukarıda belirtilen otomatik deęerlendirme aracı olan Dr. Scratch ve dięer bir deęerlendirme aracı olan Bilge Kunduz Uluslararası Enformatik Soruları bu arařtırmada deęerlendirme aracı olarak kullanılıp, öęrencilerin hem geliřtirdikleri programların analizi yapılmıř, aynı zamanda bilgi iřlemsel dūřünme, algoritma geliřtirme ve problem çözmeye becerileri ölçülmüřtür.

3. YÖNTEM

Bu bölümde araştırma modeli, çalışma grubu, veri toplama araçları, uygulama süreci, verilerin çözümlenmesi ve yorumlanmasına ilişkin bilgilere yer verilmiştir.

3.1. Araştırma Modeli

Bu çalışmada hem nicel hem nitel yöntemleri içeren karma araştırma yöntemlerinden olan “Açıklayıcı Sıralı Karma Yöntem” kullanılmıştır. Creswell (2012), bu yöntemi aynı anda veri toplayıp sonuçlarını birleştirmek yerine, nicel ve nitel bilgileri sırayla iki aşamada toplayabilir ve bir veri toplama formuyla takip edebilir şeklinde ifade etmiştir. Bu araştırma modeli nitelikli bulgular elde etmek için birinci aşamada nitel ve nicel verilerin en iyilerini elde eder ve ikinci aşamada bulgular kısmında ayrıntılı olarak detaylandırır (Creswell, 2012).

Bu çalışmada yarı deneysel ön test-son test eşleştirilmiş kontrol gruplu desen uygulanmıştır. Bu model şematik olarak Tablo 2’de verilmektedir.

Tablo 2. Ön Test - Son Test Eşleştirilmiş Kontrol Gruplu Desen

Grup	Ön Test	İşlem	Son Test
Deney	O1	X	O3
Kontrol	O2	Y	O4

X: Ters Yüz Sınıf Modeli ile Eğitim

Y: Yüz Yüze Eğitim

O1-O3: Deney Grubu ön test ve son test ölçümleri

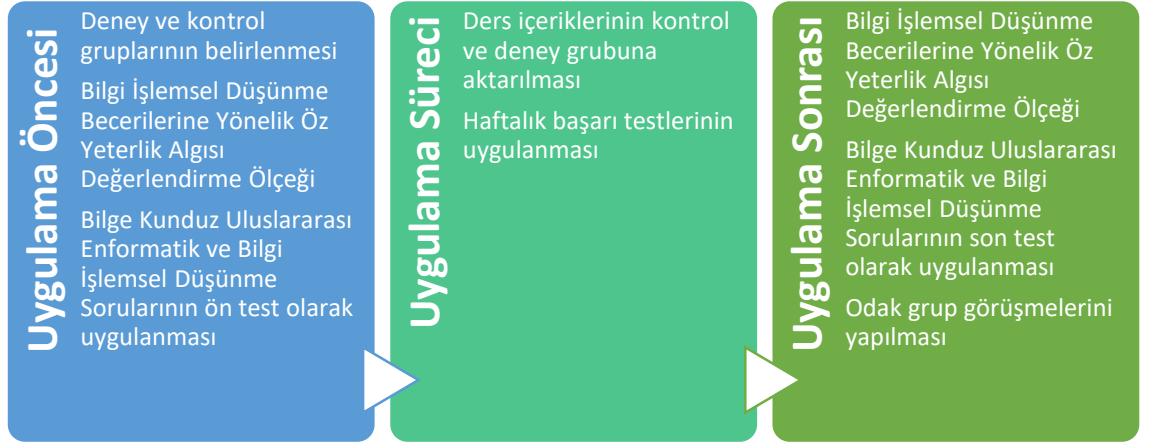
O2-O4: Kontrol Grubu ön test ve son test ölçümleri

3.2. Çalışma Grubu

Araştırmanın çalışma grubunu Ankara ilinin Çankaya ilçesinde özel bir okulun 5. sınıfında okuyan (5A, 5B, 5C ve 5D) toplam 79 öğrenci oluşturmaktadır. Araştırmacı görev yaptığı okulun öğrencileriyle Bilişim Teknolojileri dersinde bu araştırmayı gerçekleştirmiştir. Çalışmaya katılan öğrenciler 11 yaşında olup, 30’u (% 38) kız, 49’ü (% 62) erkektir. Öğrencilerin hiçbiri daha önceden Scratch aracını kullanmamışlardır. Seçilen grupların akademik başarıları araştırmacı tarafından incelenmiş, aynı zamanda okul politikası olarak sınıfların akademik anlamda homojen oluşturulmasından dolayı grupların eşit düzeyde oldukları varsayılarak rasgele gruplara ayrılmıştır.

3.3. Veri Toplama Araçları

Uygulama öncesi Bilgi İşlemsel Düşünme Becerilerine Yönelik Öz Değerlendirme Ölçeği ön test olarak uygulanmıştır. Bilge Kunduz Uluslararası Enformatik Yarışmasında kullanılan sorulardan uzman görüşü alınarak seçilmiş 15 soru seçilmiştir. Bu sorulardan oluşturulan sınav öğrencilere ön test olarak uygulanmıştır. Uygulama süresince her ders sonunda ilgili haftanın kazanımlarını ölçmek amacıyla uzman görüşü alınarak araştırmacı tarafından hazırlanan sınavlar uygulanmıştır. Uygulama sonrası ön test olarak uygulanan Bilgi İşlemsel Düşünme Becerilerine Yönelik Öz Değerlendirme Ölçeği ve Bilge Kunduz Uluslararası Enformatik Yarışmasında kullanılan sorulardan uzman görüşü alınarak seçilmiş 15 soru son test olarak uygulanmıştır. Ayrıca öğrencilerle Scratch ile programlama öğretimi hakkında odak grup görüşmeleri ile toplanmıştır. Yapılacak çalışmaya, öğrencilerin yaş seviyelerine uygunluğu gibi birçok faktör dikkate alınarak ve birçok araştırmacıdan uzman görüşü alınarak testler uygulanmaya karar verilmiştir. Uygulama öncesi, uygulama ve uygulama sonrası veri toplama süreci Şekil 15’ te yer almaktadır.



Şekil 15. Veri Toplama Süreci

Bu araştırmada deney grubu ters yüz sınıf yöntemi ile programlama eğitimi alan, kontrol grubu yüz yüze eğitim ile programlama eğitimi alan öğrenciler olarak seçilmiştir.

3.3.1. Bilgi İşlemsel Düşünme Becerilerine Yönelik Öz Yeterlik Algısı Değerlendirme Ölçeği (BİDBÖA)

Bilgi İşlemsel Düşünme Becerilerine Yönelik Öz Değerlendirme Yeterlik Algısı Ölçeği Gülbahar, Kert ve Kalelioğlu (baskıda) tarafından geliştirilmiştir. Ölçek, 17 farklı ortaokuldan 916 5. ve 6. sınıf öğrencisinin katılımı ile oluşturulmuştur.

Tablo 3. BİDBÖA Güvenilirlik Katsayıları

Alt Faktör	Güvenilirlik Katsayısı
Algoritma Tasarlama Yeterliği	0.93
Problem Çözme Yeterliği	0.86
Veri İşleme Yeterliği	0.86
Temel Programlama Yeterliği	0.84
Özgüven Yeterliği	0.79

Tablo 3’de verildiği üzere üçlü likert tipinde, 39 sorudan oluşan ölçekte Algoritma Tasarlama Yeterliği, Problem Çözme Yeterliği, Veri İşleme Yeterliği, Temel Programlama Yeterliği ve Özgüven Yeterliği olmak üzere beş faktör bulunmaktadır. Ölçeğin toplam güvenilirlik katsayısı 0.945 olarak ortaya çıkarken, alt faktörlere ilişkin güvenilirlik katsayıları ise Algoritma Tasarlama Yeterliği için 0.93, Problem Çözme Yeterliği için 0.86, Veri İşleme Yeterliği için 0.86, Temel Programlama Yeterliği için 0.84 ve son faktör olan Özgüven Yeterliği için 0.79 bulunmuştur.

3.3.2. Bilge Kunduz Uluslararası Enformatik ve Bilgi İşlemsel Düşünme Soruları

Araştırma süresince öğrencilere verilecek kazanımlara benzer kazanımları içeren sorular içerik ve kavramsal düzeyleri konusunda uzman görüşü alınarak çeşitli yıllara ait soru havuzundan seçilmiştir. Seçilen soruların farklı kavramsal düzeyleri ölçmesine dikkat edilmiştir. Bu şekilde öğrencilerin aynı kazanım için uygulama öncesi ve sonrası bilgi düzeyinde ne gibi farklılıkları olduğu tespit edilmiştir. Sorular seçilirken mümkün olduğunca öğrencilerin düşüncelerini zorlayacak ve gerçek bir programlamada yazılan algoritma tasarlama becerilerini kullanabilecekleri sorular seçilmiştir. Ayrıca birçok soru tek bir kazanımı içerirken, bazı sorular birden fazla kazanımı içererek aynı anda birden fazla algoritma tasarlama becerilerini de ölçülmesine dikkat edilmiştir.

Araştırmada 2014 ve 2015 yılında sorulan sorulardan seçilmiştir (Gülbahar & Kalelioğlu, 2014; Gülbahar & Kalelioğlu, 2015). 2014 yılı Bilge Kunduz Uluslararası Enformatik ve Bilgi İşlemsel Düşünme Testi güvenilirlik katsayısı 0,65 olarak bulunmuştur. 2015 yılı Bilge Kunduz Uluslararası Enformatik ve Bilgi İşlemsel Düşünme Testi için ise güvenilirlik katsayısı 0,63 olarak bulunmuştur. Araştırmada her zorluk derecesinden 5 soru olmak üzere toplam 15 soru sorulmuştur.

Bu teste ait güvenilirlik katsayısı 0.65 olarak bulunmuştur. Can (2017) güvenilirlik katsayısı .60-.90 arasında kalan testlerin oldukça güvenilir olduğunu belirtmiştir. Buna

göre bu arařtırmada kullanılan testlerin oldukça güvenilir olduđu hesaplanmıřtır. Sorulara ait madde ayırt edicilik indeksleri %27'lik alt – üst grup formülü ile hesaplanmıřtır. Madde toplam korelasyonu .30 ve daha yüksek olan maddelerin daha iyi ayırt ettiđi, .20-.30 arasındaki maddelerin zorunlu olması durumunda teste alınması, .20 nin altında kalan maddelerin teste alınmaması gerekmektedir. Elde edilen sonuçlar madde toplam korelasyonuna bakıldıđında 1, 2, 4, 5, 6, 9, 10, 11, 12, 13, 14, 15 numaralı maddelerin çok iyi, 3, 7, 8 numaralı maddelerin oldukça iyi maddeler olduđu görölmektedir. Her maddenin ayırıcılık indeksi Tablo 4'te yer almaktadır.

Tablo 4. Arařtırmada Uygulanan Bilge Kunduz Uluslararası Enformatik ve Bilgi İşlemsel Düşünme Soruları için Madde Ayırıcılık İndeksi

Soru No	Madde Ayırıcılık İndeksi	N
1	0.48	21
2	0.42	21
3	0.38	21
4	0.57	21
5	0.68	21
6	0.48	21
7	0.33	21
8	0.33	21
9	0.57	21
10	0.58	21
11	0.85	21
12	0.51	21
13	0.76	21
14	0.41	21
15	0.57	21

Seçilen soruların sorulduđu yıllara ait yapılan ulusal yarışmalarda elde edilen sonuçlara bakıldıđında 1, 2, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15 numaralı maddelerin çok iyi;3, 7 numaralı maddelerin oldukça iyi maddeler olduđu görölmektedir. Her maddenin ayırıcılık indeksleri Tablo 5'te verilmiştir.

Tablo 5. Ulusal Olarak Uygulanan Bilge Kunduz Uluslararası Enformatik ve Bilgi İşlemsel Düşünme Testi için Madde Ayırıcılık İndeksi

Soru No	Yılı	Madde Ayırıcılık İndeksi	N
1	2014	0.51	480
2	2014	0.52	480
3	2014	0.38	480
4	2014	0.57	480
5	2015	0.59	3721
6	2014	0.41	480
7	2014	0.39	480
8	2014	0.61	480
9	2015	0.64	3721
10	2015	0.57	3721
11	2014	0.70	480
12	2014	0.54	480
13	2014	0.66	480
14	2015	0.65	3721
15	2015	0.55	3721

Seçilen soruların hangi kavramsal düzeyi ölçtüğünü ve hangi yıla ait olduğuna dair veriler Tablo 6' da verilmektedir.

Tablo 6. Bilge Kunduz Uluslararası Enformatik ve Bilgi İşlemsel Düşünme Soruları

Soru No	Zorluk Düzeyi	Yılı	Kazanım
1	Kolay	2014	Verileri özelliklerine göre sıralama. Algoritma oluşturma.
2	Kolay	2014	Belirtilen aralıkta döngü kullanımı. Algoritma oluşturma.
3	Kolay	2014	Dışarıdan alınan verileri kullanarak problem çözme.
4	Kolay	2014	Dışarıdan alınan verileri kullanarak problem çözme. Karakteri verilen yönergeler doğrultusunda hareket ettirme.
5	Kolay	2015	Verileri Sıralama, algoritma oluşturma.
6	Orta	2014	Programlamada rastgele kavramı
7	Orta	2014	Algoritma oluşturma. Programlamada hata ayıklama
8	Orta	2014	Verilen veriler ile algoritma oluşturma. Problem çözümü için hangi verilere ihtiyaç olduğunu saptayabilme
9	Orta	2015	Bir nesne grubunun belirli koşul ve gereksinimlere göre doğru ya da yanlış olduğunun belirlenmesi. Algoritma oluşturma.
10	Orta	2015	Verilen koşulların ilgili durumlara uyup uymadığı test edilir. Algoritma oluşturma.
11	Zor	2014	Algoritma kurma. Programlamada aynı anda başlayan ve farklı zaman süreleri içerisinde biten işlemlerin sonuçlarını saptayabilme
12	Zor	2014	Problem çözümünde en az çaba veya program geliştirmede en az kod ile verilen görevi yerine getirebilme. Algoritma oluşturma.
13	Zor	2014	Problemi verilen veriler ile çözebilme. Algoritma oluşturma..
14	Zor	2015	Basit hesaplama işlemi yapan bir makinenin, sonuca ulaşmak için bilginin oluşturma sürecine ilişkin sürecin hangi aşamalardan geçtiği saptamak. Algoritma oluşturma.
15	Zor	2015	Çizge Kuramı, yönlendirilmiş çizge, düğümler ve kenarlar, sulama sistemi. Algoritma oluşturma.

Tablo 6'ya bakıldığında seçilen 15 sorunun tümünde algoritma oluşturma kavramsal düzeyi bulunmaktadır. Bu etkinlikte temel kazandırılması istenilen davranış algoritma kurma becerisi olduğundan, tüm soruların kazanımlarında yer almıştır. Diğer kazanımlara

bakıldığında döngü kullanımı, verileri sıralama, karakteri hareket ettirme, rasgele kavramı, hata ayıklama, koşullar, programlamada zaman kavramı, çizge kuramı olduğu görülmektedir. Tüm bu kazanımlar araştırmada belirlenen kazanımlar ile benzerlik göstermektedir.

3.3.3. Haftalık Başarı Testi

Haftalık başarı testleri, her hafta yapılan derslerden sonra işlenen kazanımları içeren 2 ve 3 soruluk 4 seçenekten oluşan çoktan seçmeli sınavlardır. Sınav soruları her hafta en az 3 uzman tarafından kontrol edilip içerik ve görünüm geçerliliği sağlanmıştır. Sorular Google Formlar üzerinden hazırlanıp her dersin sonunda, o hafta yapılacak olan sınavın web adresi her öğrencinin bilgisayarında açılıp çözmeleri sağlanmıştır. Soruların hangi kavramsal düzeyi ölçtüğüne dair veriler Tablo 7’de verilmektedir.

Tablo 7. Haftalık Başarı Testleri ve Kazanım İlişkileri

Sınav No	Soru No	Kazanım
1	1	Dışarıdan alınan verileri kullanarak problem çözme. Algoritma kurma
	2	Karakterli verilen yönergeler doğrultusunda hareket ettirme. Algoritma kurma
	3	Dışarıdan alınan verileri kullanarak problem çözme. Algoritma kurma
2	1	Verileri Sıralama. Algoritma kurma.
	2	Verileri Sıralama. Algoritma kurma. Döngüleri kullanma
3	1	Algoritma kurma. Eğer-Değilse koşul kullanımı
	2	Algoritma kurma. Eğer-Değilse koşul kullanımı. Çizim araçlarını kullanma. Algoritma kurma
	3	Verilen veriler ile algoritma kurabilme. Programlamada hata ayıklama. Algoritma kurma
4	1	Algoritma kurma Değişken kullanımı
	2	Algoritma kurma. Eğer-Değilse koşul kullanımı Döngü kullanımı
	3	Algoritma kurma. Problem çözümü için hangi verilere ihtiyaç olduğunu saptayabilme
5	1	Dışarıdan alınan verileri kullanarak problem çözme. Problem çözümü için hangi verilere ihtiyaç olduğunu saptayabilme. Algoritma kurma
	2	Verilen veriler ile algoritma kurabilme. Programlamada rastgele kavramı
	3	Programlamada fonksiyon kullanımı (Haber Sal). Algoritma kurma

6	1	Matematiksel işlem. Algoritma kurma. Kullanıcıdan yanıt alma
	2	Koşula bağlı döngü Programlamada rasgele komutu kullanımı. Algoritma kurma
	3	Programlamada fonksiyon kullanımı. Algoritma kurma Kullanıcıdan yanıt alma Değişken kullanımı

Tablo 7’de belirtildiği üzere haftalık başarı testlerinin tüm sorularında temel programlama becerisi olan algoritma kurma becerisi kazanımı vardır. Bununla birlikte programlamada kullanılan değişken kullanımı, fonksiyonlar, kullanıcıdan yanıt alma, döngüler, koşullar, rasgele kavramı, çizim araçlarını kullanma, verileri sıralama gibi kavramlar bulunmaktadır. Başka bir şekilde ifade edilirse haftalık başarı testleri temel programlamada kullanılan hemen hemen tüm kavramları içermektedir.

Haftalık başarı testlerinin güvenilirlik katsayıları hesaplamalarının sonucunda ortaya çıkan sonuçlar yorumlandığında tüm testlerin güvenilirlik katsayılarının .60’dan büyük olduğu görülmektedir. Daha önce bahsedilen .60’dan büyük katsayılı testlerin oldukça güvenilir olduğu ifadesine göre haftalık testlerinin tümü oldukça güvenilir söylenebilir. Güvenilirlik katsayılarına ilişkin veriler Tablo 8’de yer almaktadır.

Tablo 8. Haftalık Başarı Testlerinin Güvenilirlik Katsayıları

Sınav No	Güvenilirlik Katsayısı	N
1	0,637	79
2	0,652	79
3	0,624	79
4	0,615	79
5	0,626	79
6	0,645	79

Tablo 8’ de belirtildiği üzere 6 farklı hafta da uygulanan ve her haftaya ait kazanımları içeren haftalık başarı testlerinin tümü oldukça güvenilir test olduğu sonucuna ulaşılmıştır. Bu testlerin kazanımları ölçmek için yeterli olduğunu göstermektedir. Haftalık başarı testlerinin %27’lik alt – üst grup yöntemi ile madde ayırıcılık indeksi hesaplanmıştır. Soruların tümünün ayırıcılık indeksleri .30’dan büyük olduğundan tüm sorular için oldukça iyi denilebilir. Her maddenin ayırıcılık indeksi Tablo 9’da yer almaktadır.

Tablo 9. Haftalık Başarı Testlerinin Madde Ayırıcılık İndeksi

Sınav No	Soru No	Madde Ayırıcılık İndeksi	N
1	1	0,76	21
	2	1	
	3	0,52	
2	1	0,85	21
	2	0,90	
3	1	0,47	21
	2	0,95	
	3	0,95	
4	1	0,38	21
	2	0,52	
	3	0,95	
5	1	0,76	21
	2	0,80	
	3	0,71	
6	1	0,57	21
	2	0,95	
	3	0,95	

3.3.4. Haftalık Öğrenci Çalışmaları

Haftalık öğrenci çalışmalarında öğrenciler o hafta öğrendikleri kazanımla ilişkili araştırmacı tarafından verilen etkinliği istenilen şekilde yapmışlardır. Yapılan tüm çalışmalar her hafta Dr. Scratch üzerinden hesaplanan puanların ortalamaları alınarak gruplar arası ortalama puan değerlendirilmesi yapılmıştır. Aynı haftadaki etkinliklerde hem deney grubundaki öğrencilerin hem de kontrol grubu öğrencilerinin ortalama puanlarına bakılmıştır. Her öğrenci Dr. Scratch sistemine girip kendi yaptığı uygulamayı sistemde değerlendirip belirtmiştir.

3.3.5. Odak Grup Görüşmesi

Öğrencilerin Bilge Kunduz Uluslararası Enformatik ve Bilgi İşlemsel Düşünme Soruları sonuçlarından aldıkları puanlara göre yüksek ve düşük olarak tespit edilen her şubeden 6 kişi olmak üzere toplam 24 öğrenci ile odak grup görüşmesi yapılmıştır. Öğrencilerle görüşmeler yapılırken onların izinleri ile sesleri kaydedilmiş ve daha sonra veriler nicel araştırmalara destek olmak amacıyla nitel veri çözümleme yöntemlerine göre değerlendirilmiştir.

Sorular ilk olarak araştırmacı tarafından oluşturulmuştur. Daha sonra uzman görüşleri alınarak soruların son hali verilmiştir. Toplam 5 sorudan oluşan görüşmelerde her iki grubun ilk iki sorusu sadece grubun ders işleme yöntemine yönelik olup diğer 3 soru Scratch ortamında programlama öğrenimine yönelik olup her iki gruba ortak olarak sorulmuştur. Her iki gruba sorulan sorular Ek-3 bölümünde yer almaktadır.

3.4. Uygulama Süreci

Uygulama 2017-2018 eğitim öğretim yılı güz döneminde 7 hafta boyunca yapılmıştır. Uygulama öncesi araştırmacı çalıştığı kurumdan bu araştırmayı yapabilmek için izin almıştır. Scratch programlama öğretiminde farklı iki strateji kullanılmıştır. Birincisi öğretmen anlatımı ile doğrudan öğretim stratejisi, bir diğeri ise ters yüz sınıf modeli ile evde ders okulda ödev stratejisidir. Toplamda 4 şube olan 5. sınıf öğrencilerinden iki şube yüz yüze eğitim modeli ile programlama öğretimi yapılmıştır. Konu kazanımları ders içerisinde araştırmacı tarafından anlatılmıştır. Bir kazanımın verilmesinde konu anlatımı, gösterip yaptırma, soru-cevap, bağımsız uygulama-ödev yöntemleri sırası ile uygulanmıştır.



Şekil 16. Yüz Yüze Eğitim Yapan Grup için Kullanılan Öğretim Yöntemleri

Kazanım sözel, metinsel, görsel ve hareketli medyalar kullanılarak öğrencilere anlatılmıştır. Ders anlatımı kod üzerinden yapıp bu kodların çalışır hali gösterilmiştir. Ders içerisinde soru cevap yöntemi ile her öğrenciden konuyu anlayıp anlamadıklarına dair geri bildirimler alınmıştır. Konu anlatımı bittikten sonra öğrencilere öğrendikleri kod bloklarını kullanacakları daha önce araştırmacı tarafından hazırlanan ve son hali öğrencilere gösterilen bir etkinlik veya ödev yapmaları istenmiştir.

Diğer iki şubede ise aynı konular öğrenme yönetim sistemi üzerinden işlenmiştir. Bir kazanımın verilmesinde konuyu kendi kendilerine keşfederek öğrenme, anlamadıkları yerleri not alıp öğretmene soru hazırlama, sınıf içinde öğretmen rehberliğinde aktivite yapma, Bağımsız uygulama-Ödev yöntemleri sırası ile uygulanmıştır.



Şekil 17. Ters Yüz Sınıf Model ile Eğitim Yapan Grupta Kullanılan Öğretim Yöntemleri

Sistemde bulunan derslerden ilgili haftaya ait olan konular öğrencilere açılıp evde konuya çalışarak kazanımı keşfederek öğrenmeleri istenmiştir. Derse geldiklerinde soru cevap yöntemi ile konuyu anlayıp anlamadıklarına dair geri bildirimler alınmıştır. Bu grup diğer gruba ödev olarak verilen etkinlikleri okulda yapmaları istenmiştir. Her hafta yapılan ödevler değerlendirilip Veli-Öğretmen Sistemine işlenmiştir. Her iki gruba o haftaya ait kazanımları değerlendiren 3 soruluk mini sınavlar derste uygulanmıştır.

Araştırmacı her derste o derse ait ders günlüğü tutmuştur. Bu günlükte derste işlenen konular, öğrenci davranışları, konuya ait öğrenci hazırbuluşlukları ve tutumları, oluşan

teknik aksaklıklar ve araştırmacının bireysel gözlemleri yer almaktadır. Uygulama öncesi ve sonrası “Bilgi İşlemsel Düşünme Becerilerine Yönelik Öz Değerlendirme Ölçeği” ve “Bilgi İşlemsel Düşünme Etkinliği” ön test ve son test olarak aynı sorular sorularak uygulanmıştır. Çalışmadaki uygulama takvimi Tablo 10’da görüldüğü gibidir.

Tablo 10. Haftalık Konu Dağılımları

Hafta	Deney Grubu Konuları	Kontrol Grubu Konuları	Etkinlik Adı	Kazanımları
1	Sahne ve Karakter Özellikleri Karakteri Boyutlandırma	Sahne ve Karakter Özellikleri Karakteri Boyutlandırma	Sahne ve Karakter Özellikleri Karakteri Boyutlandırma	Kuklayı boyutlandırır. Sahneyi tanır ve karakter ekler.
2	Karakter Zıplatma Hareketi Karakter Yürüyüşü Animasyonu	Karakter Zıplatma Hareketi Karakter Yürüyüşü Animasyonu	Karakter Zıplatma Hareketi Karakter Yürüyüşü Animasyonu	Kuklayı yatayda ve dikeyde hareket ettirir.
3	Fare ile Etkileşim Klavye ile Etkileşim	Fare ile Etkileşim Klavye ile Etkileşim	Fare ile Etkileşim Klavye ile Etkileşim	Fare ile kuklayı hareket ettirir. Klavye ile kuklaya yön verir. Döngüleri kullanır. Koşul ifadesi yazar.
4	Değişken Kullanımı Fonksiyonlar	Değişken Kullanımı Fonksiyonlar	Penaltı – gol	Değişken tanımlar ve kullanır. Koşul ifadesi yazar.
5	Programlamada Rasgele Kavramı Döngüler	Programlamada Rasgele Kavramı Döngüler	Havai fişek	Programlamada rasgele kavramını öğrenir. Döngüleri kullanır. İkiz oluşturmayı öğrenir.
6	Matematiksel İşlem Koşula Bağlı Döngü Kullanıcıdan Yanıt Alma	Matematiksel İşlem Koşula Bağlı Döngü Kullanıcıdan Yanıt Alma	Biraz da geometri	Özel taş oluşturma ve kullanmayı öğrenir. Koşula bağlı döngü yazmayı öğrenir. Kullanıcıdan yanıt alabilecek yapıyı kurar.
7	Programlamada Zaman Kavramı	Programlamada Zaman Kavramı	Analog saat	Programlamada Zaman Kavramını bilir.(Şimdiki saat, şimdiki dakika, şimdiki saniye)

Tablo 10'a bakıldığında deney ve kontrol gruplarına uygulama sonunda oluşan farkları belirlemek adına 7 hafta boyunca aynı konu ve kazanımlar anlatılmıştır. İlk olarak kazanımlar belirlenmiş olup bu kazanımlara uygun konular ve bu konulara ait etkinlikler belirlenerek öğrencilere uygulanmıştır.

3.4.1. Hazırlık Çalışması

Çalışmalara başlamadan önce okul yönetimi ile ilgili yazışmalar yapılarak araştırmanın uygulanabilmesi için gerekli izinler alınmıştır. Öğrenciler Scratch hakkında daha önce bilgi sahibi olmadıkları için bu dönem içerisinde yapılacak etkinlikler ve ders hakkında bilgiler verilmiştir. Geleneksel ders modeli ile ders işleyecek gruba derste neler yapılacağı ve ödev olarak verilen etkinliklerin nasıl tamamlanacağını, verilen ödevlerin hangi sistem üzerinden, nasıl erişebilecekleri konusunda bilgiler verilmiştir.

Ters yüz sınıf modeli kullanılarak ders işlenen grupta, öğrencilere konu içeriklerine erişebilecekleri öğrenme yönetim sistemi hakkında bilgiler verildi. Konulara nasıl ulaşacaklarını, konuyu öğrenirken hangi kriterlere dikkat etmeleri gerektiği anlatıldı. Sınıf içinde yapılacak etkinlikler konusunda bilgi verildi.

3.4.2. Ön Testlerin Uygulanışı

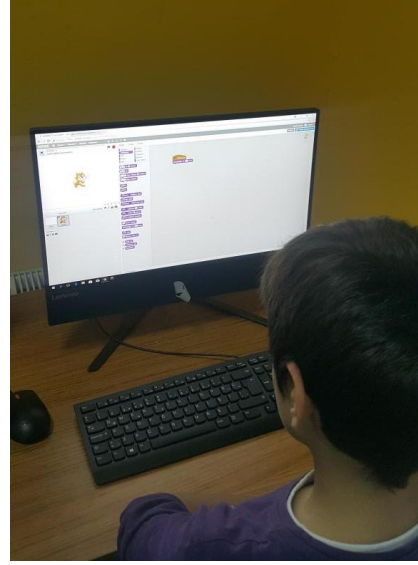
Bilgi İşlemsel Düşünme Becerilerine Yönelik Öz Değerlendirme Ölçeği, Bilge Kunduz Uluslararası Enformatik Yarışmasında kullanılan sorulardan uzman görüşü alınarak seçilmiş 15 soruluk uygulama, iki ders saati süresince ön test olarak uygulanmıştır.

3.4.3. Scratch'a Kayıt ve ÖYS Kullanımı

Öğrenciler ilk olarak Gmail üzerinden birer mail adresi almaları istenmiştir ve bir ders boyunca bu adreslerin alınması sağlanmıştır. İkinci derste Scratch'ın web sitesine girerek bu siteye üye olmaları sağlandı. Öğrencilere kullanacakları ÖYS den derse ve etkinliklere nasıl girecekleri anlatıldı. Ödevlerini nereden erişebileceklerini geribildirim nasıl yapılacağına dair bilgiler verildi.

3.4.4. Sahne ve Karakter Özellikleri, Karakteri Boyutlandırma

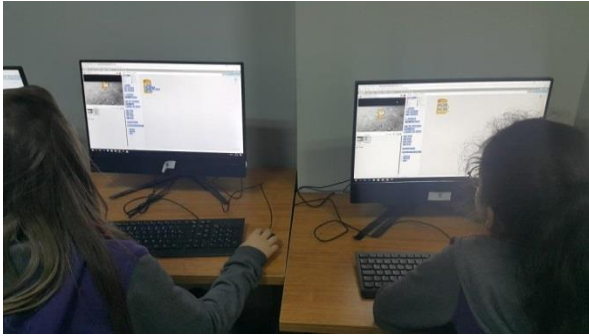
Bu uygulamanın kazanımı Scratch'te sahne, karakter kavramını öğrenir. Bu konunun etkinliklerinde öğrenciler karakter nasıl eklenir, karakterin rengini değiştirme, sahenin düzenlenmesi, karakterin kılıklarını değiştirme işlemlerini yapmışlardır.



Şekil 18. Sahne ve Karakter Özellikleri, Karakteri Boyutlandırma Etkinliği

3.4.5. Karakter Zıplama Hareketi, Karakter Yürüyüş Animasyonu

Bu uygulamanın kazanımı kuklayı yatayda ve dikeyde hareket ettirir. Uygulamada öğrenciler hareket bloğunda bulunan x'i 10 arttır, y'yi 10 arttır ve görünüm bloğundaki büyüklüğünü %35 yap komutlarını kullanarak karakterin hareket etmesini ve ölçeklendirme çalışması yapmışlardır.



Şekil 19. Karakter Zıplama Hareketi, Karakter Yürüyüş Animasyonu Etkinliği

3.4.6. Fare ile Etkileşim ve Klavye ile Etkileşim

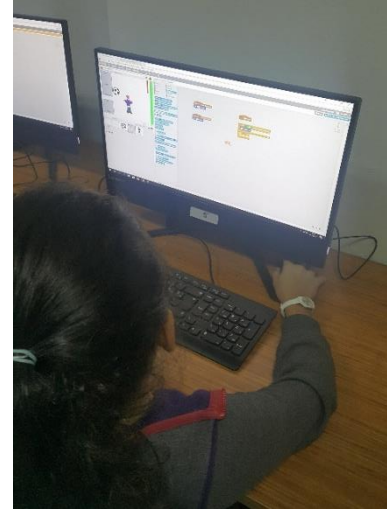
Bu uygulamanın kazanımları fare ile kuklayı hareket ettirir, klavye ile kuklaya yön verir, döngüleri kullanır, koşul ifadesi yazar. Uygulamada öğrenciler kuklaları fare hareketleri ve klavye tuşları ile kontrol etmeyi öğrenmişlerdir. Burada hareket sekmesinde bulunan fare okuna dön, fare okuna doğru git komutlarını, olaylar sekmesinde bulunan tuşa basılınca komutlarını kullanmışlardır. Ayrıca Sürekli tekrarla ve eğer ise gibi döngü ve koşul ifadelerini de ilk defa kullanarak nasıl kullanacakları konusunda bilgilendirilmiştir.



Şekil 20. Fare ve Klavye ile Etkileşim Etkinliği

3.4.7. Penaltı – Gol

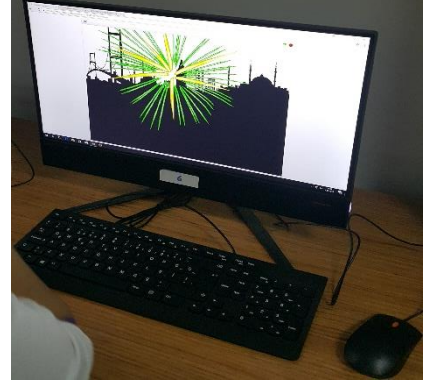
Bu uygulamanın kazanımları değişken tanımlar ve kullanır, koşul ifadesi yazar. Uygulamada bit top bir kale ve birde kaleci bulunmaktadır. Top 5 kere gol oluncaya kadar oyun devam etmektedir. Oyunda skoru tutmak adına veri sekmesinde yer alan değişken oluşturma, değer atama işlemlerini yapmışlardır. İki kuklanın birbirine değip değmediğini kontrol eden koşullu ifadeler kullanmışlardır. Ayrıca hareket sekmesinde yer alan ...derece dön komutunu da kullanmışlardır.



Şekil 21. Penaltı – Gol Etkinliği

3.4.8. Havai Fişek

Bu uygulamanın kazanımları programlamada rasgele kavramını öğrenir, döngüleri kullanır, ikiz oluşturmaya öğrenir. Uygulamada bir top yükselerek patlıyor ve havai fişek efektleri ortaya çıkıyor. Öğrenciler bu uygulamayı yaparken rasgele bir noktadan topu fırlatma, sürekli döngü ve aynı karakteri birden fazla kullanmayı sağlayan ikizini oluşturma algoritmasını kullanmışlardır. Bu sayede öğrenciler programlamada rasgele kavramını, döngü kavramını ve nesneden instance alma kavramını kullanmışlardır.



Şekil 22. Havai Fişek Etkinliği

3.4.9. Biraz da Geometri

Bu uygulamanın kazanımları özel taş oluşturma ve kullanmayı öğrenir, koşula bağlı döngü yazmayı öğrenir, kullanıcıdan yanıt alabilecek yapıyı kurar. Uygulama kullanıcıdan kaç kenarlı çokgen istediğini sorar ve kullanıcın verdiği yanıtı göre o kadar kenarlı çokgen çizmektedir. Öğrenciler bu uygulamada programlamadaki metot oluşturmaya yarayan özel taş kodlarını, kullanıcın yanıt olarak verdiği sayıyı kullanarak kontrol sekmesinde bulunan döngü kodlarını ve Bu sayıyı kullanıcıdan alabilmek için algılama sekmesinde bulunan yanıt okuma kodlarını kullanmışlardır.



Şekil 23. Biraz da Geometri Etkinliği

3.4.10. Analog Saat

Bu uygulamanın kazanımı programlamada Zaman Kavramını bilir. (Şimdiki saat, şimdiki dakika, şimdiki saniye) Bu uygulamada öğrenciler analog bir saat yapmışlardır. Bu saati yaparken öğrenciler programlamada zaman kavramını kullanmışlardır. Akrep, yelkovan ve saniye bir saat içerisindeki dönme açıları matematiksel olarak hesaplanıp anime edilmiştir.



Şekil 24. Analog Saat Etkinliği

3.5. Verilerin Analizi

Verilerin analizi için SPSS 22.0 istatistik programı kullanılmıştır. Bilgi İşlemsel Düşünme Becerilerine Yönelik Öz Yeterlik Algısı Değerlendirme Ölçeği ve Bilge Kunduz Uluslararası Enformatik ve Bilgi İşlemsel Düşünme Soruları çalışma öncesi ve sonrasında deney ve kontrol gruplarına uygulanmıştır. Grupların ön test puanları bakımından bir farklılık olup olmadığına bakmak için bağımsız örneklem t-testi yapılmıştır. Çalışma sonrasında Bilgi İşlemsel Düşünme Becerilerine Yönelik Öz Yeterlik Algısı Değerlendirme Ölçeği ve Bilge Kunduz Uluslararası Enformatik ve Bilgi İşlemsel Düşünme Soruları son-test olarak deney ve kontrol gruplarına uygulanmıştır. Grupların son test puanları bakımından bir farklılık olup olmadığına bakmak için bağımsız örneklem t-testi yapılmıştır.

Bağımsız örneklem t-testi iki ilişkisiz örneklem ortalamaları arasındaki farkın anlamlı olup olmadığını test etmek için kullanılır. Bu testin yapılabilmesi için aşağıda belirtilen varsayımları sağlaması gerekmektedir.

1. Her iki gruptaki ölçümlerin dağılımlarına varyanslar eşittir.
2. Bağımlı değişkene ait ölçümlerin dağılımı her iki grupta da normaldir.
3. Ortalama puanları karşılaştırılacak örneklem ilişkisizdir.

Bu işlemlerin yapılmasından sonra her iki grubun kendi içerisinde ön test ve son test puan ortalamaları bakımından anlamlı bir fark olup olmadığını belirlemek üzere bağımlı örneklem t-testi yapılmıştır. Yapılan tüm analizlerde 0.05 anlamlılık düzeyi kullanılmıştır.

Bu testin güvenilir sonuçlar verebilmesi için aşağıdaki koşulları sağlaması gerekir (Can, 2017).

1. Ortalamaları kıyaslanacak verilerin farklarının oluşturduğu veri dizisi normal dağılım özelliklerini taşımalıdır.
2. Fark puanlar birbirinden bağımsızdır. Örneklem evrenden rasgele seçilmişlerdir.

Yukarıda belirtilen varsayımların sağlanma durumları bulgular bölümünde belirtilmiştir.

En son olarak odak grup görüşmesi ile öğrencilerin farklı stratejiler ile öğrenme konusunda görüşleri alınmıştır. Görüşmede birbirini tekrar eden, birbiriyle örtüşen, birbirini destekleyen kodlar temalar altında toplamıştır. Çalışmada içerik analizi yapılırken araştırma problemleri ile ilgili kodlar belirlenmiş ve bu kodlara uygun temalar oluşturulmuştur. Kodlanan verilerin güvenilirliğini sağlamak amacıyla elde ettiği kodları iki hafta ara ile tekrar kodlamış kodlama güvenilirliği .71 olarak hesaplanmıştır. Daha sonra başka bir uzman araştırmacıdan verilerin kodlanması istenmiş ve iki kodlama arasındaki ilişkiye bakılmıştır. Kodlama güvenliği .83 olarak bulunmuştur.

4. BULGULAR

Bu bölümde bulgular, nicel ve nitel araştırma bulguları olarak sırasıyla açıklanmıştır.

4.1. Ters Yüz Sınıf Modeli ve Yüz Yüze Eğitim ile Programlama Öğretiminin Bilgi İşlemsel Düşünme Becerilerine Etkisine İlişkin Bulgular

Bu bölümde Ters yüz sınıf modeli ve yüz yüze eğitim ile programlama öğretiminin deney ve kontrol gruplarında bilgi işlemsel düşünme becerilerine etkisini belirlemek amaçlı ön test ve son test olarak uygulanan Bilge Kunduz Uluslararası Enformatik ve Bilgi İşlemsel Düşünme sorularına ilişkin bulgular sunulmaktadır.

Deney ve Kontrol gruplarına uygulanan Bilge Kunduz Uluslararası Enformatik ve Bilgi İşlemsel Düşünme Sorularından oluşturulan ön ve son test, Bilgi İşlemsel Düşünme Becerilerine Yönelik Öz Yeterlik Algısı Değerlendirme Ölçeği ön ve son test sonuçlarını gruplar arası farklılıklarının analizinde kullanılan bağımsız örneklem t-testi varsayımları incelendiğinde;

Bilge Kunduz Uluslararası Enformatik ve Bilgi İşlemsel Düşünme Sorularından oluşturulan ön test için bu varsayımlar incelendiğinde;

Her iki gruptaki ölçümlerin dağılımlarına varyanslar eşittir varsayımının geçerliliğinde eğer kontrol ve deney gruplarının örneklem sayıları eşitse homojen olarak kabul edilir. Değilse Levene F testi ile incelenir. Bu çalışmada Deney grubu (37) ve kontrol grubunda (42) bulunan öğrenci sayıları farklı olduğundan Levene testi ile homojenlikleri test edilmiştir. Test sonuçlarına göre p değeri 0,07 olarak bulunmuştur. Bu değer 0.05 ten büyük olduğundan dolayı ($p=0.07$) bu varsayım kabul edilmiştir.

Bağımlı değişkene ait ölçümlerin dağılımı her iki grupta da normaldir varsayımının geçerliliğinde basıklık ve çarpıklık değerlerine bakılmıştır. Deney grubu çarpıklık değeri 0,6, basıklık değeri 0,8 ve kontrol grubu çarpıklık değeri 0.3, basıklık değeri-0.3 olarak belirlenmiştir. Her iki analizdeki değerler normal veri dağılım aralığından kaldığından dolayı verilerin normal dağılım gösterdiği varsayılmıştır. Ortalama puanları karşılaştırılacak örneklem ilişkisizdir varsayımının geçerliliğinde örneklem farklı şubelerden öğrenciler seçilmiş olup gruplar arasında herhangi bir ilişki bulunmamasından dolayı bu varsayım kabul edilmiştir.

Bilge Kunduz Uluslararası Enformatik ve Bilgi İşlemsel Düşünme Sorularından oluşturulan son test için bu varsayımlar incelendiğinde;

Her iki gruptaki ölçümlerin dağılımlarına varyanslar eşittir varsayımının geçerliliğinde deney grubu (37) ve kontrol grubunda (42) bulunan öğrenci sayıları farklı olduğundan Levene testi ile homojenlikleri test edilmiştir. Test sonuçlarına göre p değeri 0.109 olarak bulunmuştur. Bu değer 0.05 ten büyük olduğundan dolayı ($p=0.109$) bu varsayım kabul edilmiştir.

Bağımlı değişkene ait ölçümlerin dağılımı her iki grupta da normaldir varsayımının geçerliliğinde basıklık ve çarpıklık değerlerine bakılmıştır. Deney grubu çarpıklık değeri 0.07, basıklık değeri 0.3 ve kontrol grubu çarpıklık değeri 0.2, basıklık değeri 0.9 olarak

belirlenmiştir. Her iki analizdeki değerler normal veri dağılım aralığından kaldığından dolayı verilerin normal dağılım gösterdiği varsayılmıştır.

Ortalama puanları karşılaştırılacak örneklem ilişkisizdir varsayımının geçerliliğinde örneklem farklı şubelerden öğrenciler seçilmiş olup gruplar arasında herhangi bir ilişki bulunmamasından dolayı bu varsayım kabul edilmiştir.

4.1.1.Uygulama Öncesinde Deney ve Kontrol Gruplarına Ön Test Olarak Uygulanan Bilge Kunduz Uluslararası Enformatik ve Bilgi İşlemsel Düşünme Sorularına İlişkin Bulgular

Uygulama öncesi deney ve kontrol gruplarının başarıları arasında anlamlı bir fark olup olmadığını kontrol etmek amaçlı Bilge Kunduz Uluslararası Enformatik ve Bilgi İşlemsel Düşünme Sorularından oluşan bir ön test uygulanmıştır. Deney grubunun ortalaması 18.57 ($SS=19.2$), kontrol grubunun ortalaması ise 21.57 ($SS=29.2$) olarak hesaplanmıştır. Kontrol grubundaki öğrencilerin ortalamaları deney grubundaki öğrencilerin ortalamasından daha yüksektir. Grupların başarıları arasındaki farkın anlamlılığını kontrol etmek amaçlı bağımsız t-testi uygulanmıştır.

Tablo 11. Ön Test için Yapılan Bağımsız t-Testi Sonuçları

Gruplar	N	\bar{X}	S	sd	t	p
Deney	37	18.57	19.2	77	0.54	.58
Kontrol	42	21.57	29.2			

Tablo 11’de görüldüğü üzere, deney ve kontrol gruplarının ön test başarı puan ortalamaları arasında .05 anlamlılık düzeyinde fark bulunamamıştır ($t(77) = 0.54, p = .58$). Ortalama farkın %95 güven aralığı -8.2 ile 14.2 arasında değişmektedir. Bu durum uygulama öncesi deney ve kontrol grupları arasındaki başarı puan farkının istatistiksel olarak anlamlı olmadığını göstermektedir.

4.1.2.Uygulama Sonrasında Deney ve Kontrol Gruplarına Son Test Olarak Uygulanan Bilge Kunduz Uluslararası Enformatik ve Bilgi İşlemsel Düşünme Sorularına İlişkin Bulgular

Uygulama sonrası deney ve kontrol gruplarının ortalama puanları arasında anlamlı bir fark olup olmadığını kontrol etmek amaçlı Bilge Kunduz Uluslararası Enformatik ve Bilgi İşlemsel Düşünme Sorularından oluşan bir son test uygulanmıştır. Uygulanan teste ilişkin bulgular Tablo 12’de özetlenmiştir.

Tablo 12. Son Test için Yapılan Bağımsız Örneklem t-Testi Sonuçları

Gruplar	N	\bar{X}	S	sd	t	p
Deney	37	38.24	27.9	77	-1.1	.27
Kontrol	42	31.48	26.2			

Tablo 12’de görüldüğü üzere, deney ve kontrol gruplarının son-test başarı puan ortalamaları arasında ,05 anlamlılık düzeyinde fark bulunamamıştır ($t(77) = -1.1, p = .27$). Başka bir şekilde ifade ile yüz yüze öğretim modeli ile öğrenim gören öğrencilerin başarıları ile ters yüz sınıf öğretim modeli ile öğrenim gören öğrencilerin başarı puanları arasında ortalamalar ($\bar{X}_1 = 31.48, \bar{X}_2 = 38.24$) açısından bir fark oluşsa da genel olarak başarı puanları arasında anlamlı bir fark yoktur. Ortalama farkın %95 güven aralığı -18.9 ile 5.3 arasında değişmektedir.

4.2. Programlama Öğretiminin Deney ve Kontrol Grubu Öğrencilerinin Bilgi İşlemsel Düşünme Becerilerine Etkisine İlişkin Bulgular

Bu bölümde programlama öğretiminin deney ve kontrol gruplarında bilgi işlemsel düşünme becerilerine etkisini belirlemek amaçlı ön test ve son test olarak uygulanan Bilge Kunduz Uluslararası Enformatik ve Bilgi İşlemsel Düşünme sorularına ilişkin bulgular sunulmaktadır.

Deney ve Kontrol gruplarına uygulanan Bilge Kunduz Uluslararası Enformatik ve Bilgi İşlemsel Düşünme Sorularından oluşturulan ön ve son test sonuçlarını her grubun kendi içerisindeki farklılıklarının analizinde kullanılan bağımlı örneklem t-testi varsayımları incelendiğinde;

Ortalamaları kıyaslanacak verilerin farklarının oluşturduğu veri dizisi normal dağılım özelliklerini taşımaktadır varsayımının geçerliliğinde gruplardaki örneklem büyüklüğü 30 ve üzerinde olması durumunda normal dağılım gösterdiği varsayılır (Green ve Salkind, 2005). Bu çalışmada kontrol grubu (42) ve deney grubunda (37) bulunan öğrenci sayıları 30 dan büyük olduğu için normal dağılım gösterdiği varsayılmıştır.

Fark puanlar birbirinden bağımsızdır. Örneklem evrenden rasgele seçilmişlerdir varsayımının geçerliliğinde bağımsız t- testinde de belirtildiği üzere örneklem olarak farklı şubelerden öğrenciler seçilmiştir. Bu varsayımın geçerli olduğu kabul edilmiştir.

4.2.1. Deney Grubuna Uygulama Öncesi ve Sonrası Uygulanan Bilgi İşlemsel Düşünme Becerilerine Etkisine İlişkin Bulgular

Ters yüz sınıf modeli ile programlama eğitimi alan deney grubu öğrencilerinin programlama öğretiminin bilgi işlemsel düşünme becerilerine olan etkisini ölçmek için ön test ve son test olarak uygulanan Bilge Kunduz Uluslararası Enformatik ve Bilgi İşlemsel Düşünme Soruları başarı puanları ortalamaları arasındaki fark bağımlı örneklem t-testi ile analiz edilmiştir. Uygulanan teste ilişkin bulgular Tablo 13’ te özetlenmiştir.

Tablo 13. Deney Grubu Ön Test ve Son Test Bağımlı Örneklem t-Testi Sonuçları

Gruplar	N	\bar{X}	S	sd	t	p
Ön Test	37	18.57	19.2	36	-4.0	.00
Son Test	37	38.24	29.9			

Yapılan analiz sonucuna göre deney grubunda ters yüz sınıf modeli ile programlama eğitimi alan öğrencilerin ön test ortalama puanları ($\bar{X}_{ön}=18.57$, $SS_{ön}=19.2$) ile son test puan ortalamaları ($\bar{X}_{son}=38.24$, $SS_{son}=29.9$) arasında 0.05 anlamlılık düzeyinde anlamlı istatistiksel açıdan bir fark görülmüştür ($t(36)=-4.0$, $p<.05$). Test sonucu hesaplanan etki büyüklüğü ($d=0.65$) bu farkın orta düzeyde olduğunu göstermektedir (Green ve Salkind, 2005). Bu durum programlama öğretiminin deney grubu öğrencilerinin bilgi işlemsel düşünme becerilerini geliştirmede anlamlı bir etkisinin olduğunu göstermektedir.

4.2.2. Kontrol Grubuna Uygulama Öncesi ve Sonrası Uygulanan Bilgi İşlemsel Düşünme Becerilerine Etkisine İlişkin Bulgular

Yüz yüze öğretim modeli ile programlama eğitimi alan kontrol grubu öğrencilerinin programlama öğretiminin bilgi işlemsel düşünme becerilerine olan etkisini ölçmek için ön test ve son test olarak uygulanan Bilge Kunduz Uluslararası Enformatik ve Bilgi İşlemsel Düşünme Soruları başarı puanları ortalamaları arasındaki fark bağımlı örneklem t-testi ile analiz edilmiştir. Uygulanan teste ilişkin bulgular Tablo 14' te özetlenmiştir.

Tablo 14. Kontrol Grubu Ön Test ve Son Test Bağımlı Örneklem t-Testi Sonuçları

Gruplar	N	\bar{X}	S	sd	t	p
Ön Test	42	21.57	29.2	41	-2.2	.02
Son Test	42	31.86	26.1			

Yapılan analiz sonucuna göre kontrol grubunda yüz yüze programlama eğitimi alan öğrencilerin ön test ortalama puanları ($\bar{X}_{ön}=21.57$, $SS_{ön}=29.2$) ile son test puan ortalamaları ($\bar{X}_{son}=31.86$, $SS_{son}=26.1$) arasında 0.05 anlamlılık düzeyinde anlamlı istatistiksel açıdan bir fark görülmüştür ($t(41)=-2.2$, $p<.05$). Test sonucu hesaplanan etki büyüklüğü ($d=0.34$) bu farkın küçük düzeyde olduğunu göstermektedir (Green ve Salkind, 2005). Bu durum programlama öğretiminin kontrol grubu öğrencilerinin bilgi işlemsel düşünme becerilerini geliştirmede anlamlı bir etkisinin olduğunu göstermektedir.

4.3. Ters Yüz Sınıf Modeli ve Yüz Yüze Eğitim ile Programlama Öğretiminin Bilgi İşlemsel Düşünme Becerilerine Yönelik Öğrencilerin Öz Yeterlik Algısına İlişkin Bulgular

Bu bölümde ters yüz sınıf modeli ve yüz yüze eğitim ile programlama öğretiminin deney ve kontrol gruplarında bilgi işlemsel düşünme becerilerine yönelik öz yeterlik algısına etkisini belirlemek amaçlı ön test ve son test olarak uygulanan Bilgi İşlemsel Düşünme Becerilerine Yönelik Öz Yeterlik Algısı Ölçeğine (BİDBÖA) ilişkin bulgular sunulmaktadır. Ölçekte Algoritma Tasarlama Yeterliği, Problem Çözme Yeterliği, Veri İşleme Yeterliği, Temel Programlama Yeterliği ve Özgüven Yeterliği olmak üzere beş faktör bulunmaktadır. Bulgular bu 5 faktör üzerinden sunulmuştur.

Bilgi İşlemsel Düşünme Becerilerine Yönelik Öz Yeterlik Algısı Değerlendirme Ölçeği ön test için yapılan analizlerde her iki gruptaki ölçümlerin dağılımlarına varyanslar eşittir varsayımının geçerliliğinde deney grubu (37) ve kontrol grubunda (42) bulunan öğrenci

sayıları farklı olduğundan Levene testi ile homojenlikleri test edilmiştir. Test sonuçlarına göre p değeri 0.06 olarak bulunmuştur. Bu değer 0.05 ten büyük olduğundan dolayı ($p=0.06$) bu varsayım kabul edilmiştir.

Bağımlı değişkene ait ölçümlerin dağılımı her iki grupta da normaldir varsayımının geçerliliğinde basıklık ve çarpıklık değerlerine bakılmıştır. Deney grubu çarpıklık değeri 0.5, basıklık değeri -0.5 ve kontrol grubu çarpıklık değeri -0.1, basıklık değeri -0.5 olarak belirlenmiştir. Her iki analizdeki değerler normal veri dağılım aralığından kaldığından dolayı verilerin normal dağılım gösterdiği varsayılmıştır.

Ortalama puanları karşılaştırılacak örneklem ilişkisizdir varsayımının geçerliliğinde örneklem farklı şubelerden öğrenciler seçilmiş olup gruplar arasında herhangi bir ilişki bulunmamasından dolayı bu varsayım kabul edilmiştir.

Bilgi İşlemsel Düşünme Becerilerine Yönelik Öz Yeterlik Algısı Değerlendirme Ölçeği son test için yapılan analizlerde her iki gruptaki ölçümlerin dağılımlarına varyanslar eşittir varsayımının geçerliliğinde deney grubu (37) ve kontrol grubunda (42) bulunan öğrenci sayıları farklı olduğundan Levene testi ile homojenlikleri test edilmiştir. Test sonuçlarına göre p değeri 0.27 olarak bulunmuştur. Bu değer 0.05 ten büyük olduğundan dolayı ($p=0.27$) bu varsayım kabul edilmiştir.

Bağımlı değişkene ait ölçümlerin dağılımı her iki grupta da normaldir varsayımının geçerliliğinde basıklık ve çarpıklık değerlerine bakılmıştır. Deney grubu çarpıklık değeri -0.6, basıklık değeri -0.1 ve kontrol grubu çarpıklık değeri 0.2, basıklık değeri -0.2 olarak belirlenmiştir. Her iki analizdeki değerler normal veri dağılım aralığından kaldığından dolayı verilerin normal dağılım gösterdiği varsayılmıştır.

Ortalama puanları karşılaştırılacak örneklem ilişkisizdir varsayımının geçerliliğinde örneklem farklı şubelerden öğrenciler seçilmiş olup gruplar arasında herhangi bir ilişki bulunmamasından dolayı bu varsayım kabul edilmiştir.

4.3.1.Uygulama Öncesinde Deney ve Kontrol Gruplarına Ön Test Olarak Uygulanan BİDBÖA İlişkin Bulgular

Uygulama öncesi deney ve kontrol grupları tüm alt faktörler açısından farklı olup olmadıklarını kontrol etmek amaçlı bağımsız örneklem t-testi yapılmıştır. Yapılan analizlerde her bir faktörün deney ve kontrol gruplarındaki ortalaması Tablo 15'te belirtilmiştir.

Tablo 15. BİDBÖA Ön Testi Tüm Faktörler için Yapılan Bağımsız t-Testi Sonuçları

Alt Faktör	Gruplar	N	\bar{X}	S	sd	t	p
Algoritma	Deney	37	12.57	4.1	77	1.8	.86
Tasarlama	Kontrol	42	11.17	2.5			
Problem	Deney	37	25.89	5.3	77	-1.8	.72
Çözme	Kontrol	42	23.81	4.6			
Veri İşleme	Deney	37	15.24	4.7	77	-0.5	.58
	Kontrol	42	14.74	3.4			
Temel	Deney	37	12.00	3.0	77	-0.2	.79
Programlama	Kontrol	42	11.81	3.3			
Özgüven	Deney	37	14.32	2.8	77	-0.2	.98
	Kontrol	42	14.31	3.0			

Tablo 15'te belirtildiği üzere algoritma tasarlama yeterliliği alt faktörü için deney ve kontrol gruplarının ortalamaları arasındaki fark anlamlı değildir ($t(77) = 1.8, p = .86$). Deney grubunun ortalaması ise 12.57 ($SS=4.1$), kontrol grubunun ortalaması ise 11.17 ($SS=2.5$) olarak hesaplanmıştır. Ortalama, farkın %95 güven aralığı -2.9 ile 0.1 arasında değişmektedir.

Problem çözme yeterliliği alt faktörü için deney ve kontrol gruplarının ortalamaları arasındaki fark anlamlı değildir ($t(77) = -1.8, p = .72$). Deney grubunun ortalaması 25.89 ($SS=5.3$), kontrol grubunun ortalaması ise 23.81 ($SS=4.6$) olarak hesaplanmıştır. Ortalama, farkın %95 güven aralığı -4.3 ile 0.1 arasında değişmektedir.

Veri işleme yeterliliği alt faktörü için deney ve kontrol gruplarının ortalamaları arasındaki fark anlamlı değildir ($t(77) = -0.5, p = .58$). Deney grubunun ortalaması 15.24 ($SS=4.7$) kontrol grubunun ortalaması ise 14.74 ($SS=3.4$) olarak hesaplanmıştır. Ortalama, farkın %95 güven aralığı -2.3 ile 1.3 arasında değişmektedir.

Temel programlama yeterliliği alt faktörü için deney ve kontrol gruplarının ortalamaları arasındaki fark anlamlı değildir ($t(77) = -0.2, p = .79$). Deney grubunun ortalaması 12.00 ($SS=3.0$), kontrol grubunun ortalaması ise 11.81 ($SS=3.3$) olarak hesaplanmıştır. Ortalama, farkın %95 güven aralığı -1.6 ile 1.2 arasında değişmektedir.

Özgüven yeterliliği alt faktörü için deney ve kontrol gruplarının ortalamaları arasındaki fark anlamlı değildir ($t(77) = -0.2, p = .98$). Deney grubunun ortalaması 14.32 ($SS=2.8$), kontrol grubunun ortalaması ise 14.31 ($SS=3.0$) olarak hesaplanmıştır. Ortalama, farkın %95 güven aralığı -1.3 ile 1.3 arasında değişmektedir.

Deney ve kontrol gruplarının tüm alt faktör puan ortalamalarına ayrı ayrı bakıldığında birbirlerine yakın puanlar olduğu görülebilir. Tüm alt faktörler arasında istatistiksel olarak .05 anlamlılık düzeyinde fark bulunamamıştır. Bu sonuca göre uygulama öncesi deney ve kontrol gruplarının BİDBÖA tüm faktör yeterlilikleri istatistiksel olarak fark olmadığı düşünülebilir.

4.3.2.Uygulama Sonrasında Deney ve Kontrol Gruplarına Son Test Olarak Uygulanan BİDBÖA İlişkin Bulgular

Uygulama sonrası deney ve kontrol grupları tüm alt faktörler açısından fark olup olmadıklarını kontrol etmek amaçlı bağımsız örneklem t-testi yapılmıştır. Yapılan analizlerde her bir faktörün deney ve kontrol gruplarındaki ortalaması Tablo 16’da belirtilmiştir.

Tablo 16. BİDBÖA Son Testi Tüm Faktörler için Yapılan Bağımsız t-Testi Sonuçları

Alt Faktör	Gruplar	N	\bar{X}	S	sd	t	p
Algoritma	Deney	37	20.05	4.7	77	1.8	.09
Tasarlama	Kontrol	42	21.69	2.8			
Problem	Deney	37	28.84	3.9	77	-0.7	.44
Çözme	Kontrol	42	27.17	3.8			
Veri İşleme	Deney	37	17.84	2.8	77	-0.6	.54
	Kontrol	42	17.50	2.1			
Temel	Deney	37	14.70	2.4	77	-0.7	.44
Programlama	Kontrol	42	14.31	2.0			
Özgüven	Deney	37	15.49	1.9	77	0.3	.74
	Kontrol	42	15.62	1.6			

Tablo 16’da belirtildiği üzere Algoritma tasarlama yeterliliği alt faktörü için deney ve kontrol gruplarının ortalamaları arasındaki fark anlamlı değildir ($t(77) = -1.8, p = .09$). Deney grubunun ortalaması 20.05 ($SS=4.7$), kontrol grubunun ortalaması ise 21.69 ($SS=2.8$) olarak hesaplanmıştır. Ortalama, farkın %95 güven aralığı -0.9 ile 3.3 arasında değişmektedir.

Problem çözme yeterliliği alt faktörü için deney ve kontrol gruplarının ortalamaları arasındaki fark anlamlı değildir ($t(77) = -0.7, p = .44$). Deney grubunun ortalaması 28.84 ($SS=3.9$), kontrol grubunun ortalaması ise 27.17 ($SS=3.8$) olarak hesaplanmıştır. Ortalama, farkın %95 güven aralığı -2.4 ile 1.0 arasında değişmektedir.

Veri işleme yeterliliği alt faktörü için deney ve kontrol gruplarının ortalamaları arasındaki fark anlamlı değildir ($t(77) = -0.6, p = .54$). Deney grubunun ortalaması 17.84 ($SS=2.8$), kontrol grubunun ortalaması ise 17.50 ($SS=2.1$) olarak hesaplanmıştır. Ortalama, farkın % 95 güven aralığı -1.4 ile 0.7 arasında değişmektedir.

Temel programlama yeterliliği alt faktörü için deney ve kontrol gruplarının ortalamaları arasındaki fark anlamlı değildir ($t(77) = -0.7, p = .44$). Deney grubunun ortalaması 14.70 ($SS=2.4$), kontrol grubunun ortalaması ise 14.31 ($SS=2.0$) olarak hesaplanmıştır. Ortalama, farkın %95 güven aralığı -1.4 ile 0.6 arasında değişmektedir.

Özgüven yeterliliği alt faktörü için deney ve kontrol gruplarının ortalamaları arasındaki fark anlamlı değildir ($t(77) = 0.3, p = .74$). Deney grubunun ortalaması 15.49 ($SS=1.9$), kontrol grubunun ortalaması ise 15.62 ($SS=1.6$) olarak hesaplanmıştır. Ortalama, farkın %95 güven aralığı -0.4 ile 0.9 arasında değişmektedir.

Deney ve kontrol gruplarının tüm alt faktör puan ortalamalarına ayrı ayrı bakıldığında birbirlerine yakın puanlar olduğu görülebilir. Tüm alt faktörler arasında istatistiksel olarak .05 anlamlılık düzeyinde fark bulunamamıştır. Bu sonuca göre uygulama sonrası deney ve kontrol gruplarının BİDBÖA tüm faktör yeterlilikleri istatistiksel olarak farklı olmadığı düşünülebilir.

4.4. Programlama Öğretiminin Deney ve Kontrol Grubu Öğrencilerinin BİDBÖA Alt Faktörlerine İlişkin Bulguları

Bu bölümde programlama öğretiminin deney ve kontrol gruplarında BİDBÖA alt faktörlerine etkisini belirlemek amaçlı ön test ve son test olarak uygulanan BİDBÖA'ya ilişkin bulgular sunulmaktadır.

Deney ve Kontrol gruplarına uygulanan Bilgi İşlemsel Düşünme Becerilerine Yönelik Öz Yeterlik Algısı Değerlendirme Ölçeği ön ve son test sonuçlarını her grubun kendi içerisindeki farklılıklarının analizinde kullanılan bağımlı örneklem t-testi varsayımları incelendiğinde;

Ortalamaları kıyaslanacak verilerin farklarının oluşturduğu veri dizisi normal dağılım özelliklerini taşımaktadır varsayımının geçerliliğinde gruptaki örneklem büyüklüğü 30'dan büyük olduğu durumda normal dağılım gösterdiği varsayılır (Green ve Salkind, 2005). Bu çalışmada kontrol grubu (42) ve deney grubunda (37) bulunan öğrenci sayıları 30'dan büyük olduğu için normal dağılım gösterdiği varsayılmıştır.

Fark puanlar birbirinden bağımsızdır. Örneklem evrenden rasgele seçilmişlerdir varsayımının geçerliliğinde bağımsız t-testinde de belirtildiği üzere örneklem olarak farklı şubelerden öğrenciler seçilmiştir. Bu varsayımın geçerli olduğu kabul edilmiştir.

4.4.1. Programlama Öğretiminin Deney Grubu için BİDBÖA Alt Faktörlerine İlişkin Bulgular

Ters yüz sınıf modeli ile programlama eğitimi alan deney grubu öğrencilerinin programlama öğretiminin bilgi işlemsel düşünme becerilerine yönelik öz yeterlik algısı tüm alt faktörleri açısından uygulama öncesi ve sonrası anlamlı bir fark olup olmadığını kontrol etmek için ön test ve son test olarak uygulanan BİDBÖA'ya ait ortalamalar bağımlı örneklem t-testi ile analiz edilmiştir. Uygulanan teste ilişkin bulgular Tablo 17'de özetlenmiştir.

Tablo 17. Deney Grubu BİDBÖA Ön Test ve Son Test Algoritma Tasarlama Yeterliği için Yapılan Bağımlı Örneklem t-Testi Sonuçları

Alt Faktör	Test	N	\bar{X}	S	sd	t	p
Algoritma	Ön Test	37	12.56	4.1	36	-6.3	.00
Tasarlama	Son Test	37	20.05	4.7			
Problem	Ön Test	37	25.89	5.3	36	-2.0	.04
Çözme	Son Test	37	27,83	3.9			
Veri İşleme	Ön Test	37	15.24	4.7	36	-3.0	.00
	Son Test	37	17.83	2.8			
Temel	Ön Test	37	12.00	3.0	36	-3.9	.00
Programlama	Son Test	37	14.70	2.4			
Özgüven	Ön Test	37	14.32	2.8	36	-2.2	.02
	Son Test	37	15.48	1.9			

Tablo 17’de belirtildiği üzere deney grubuna uygulanan ters yüz sınıf modeli ile programlama öğretiminin algoritma tasarlama, problem çözme, veri işleme, temel programlama ve özgüven yeterlilikleri ortalama puanları bağımlı örneklem t testi ile analiz edilmiştir. Bu analiz sonuçları incelendiğinde uygulama öncesi algoritma tasarlama alt faktörü ön test ortalama puanı ($\bar{X}_{ön}=12.56$, $SS_{ön}=4.1$) ile uygulama sonrası algoritma tasarlama alt faktörü son test ortalama puanı ($\bar{X}_{son}=20.05$, $SS_{son}=4.7$) arasında 0.05 anlamlılık düzeyinde anlamlı istatistiksel açıdan bir fark görülmüştür ($t(36) = -6.3$, $p < .05$). Test sonucu hesaplanan etki büyüklüğü ($d=0,9$) bu farkın büyük düzeyde olduğunu göstermektedir (Green ve Salkind, 2005). Bu durum programlama öğretimi deney grubu öğrencilerinin algoritma tasarlama yeterlilikleri üzerinde anlamlı bir etkisinin olduğunu göstermektedir.

Problem çözme alt faktörü ön test ortalama puanı ($\bar{X}_{ön}=25.89$, $SS_{ön}=5.3$) ile uygulama sonrası problem çözme alt faktörü son test ortalama puanı ($\bar{X}_{son}=27.83$, $SS_{son}=3.9$) arasında 0.05 anlamlılık düzeyinde anlamlı istatistiksel açıdan bir fark görülmüştür ($t(36) = -2.0$, $p < .05$). Test sonucu hesaplanan etki büyüklüğü ($d=0.33$) bu farkın az düzeyde olduğunu göstermektedir (Green ve Salkind, 2005). Bu durum programlama öğretimi deney grubu öğrencilerinin problem çözme yeterlilikleri üzerinde anlamlı bir etkisinin olduğunu göstermektedir.

Veri işleme alt faktörü ön test ortalama puanı ($\bar{X}_{ön}=15.24$, $SS_{ön}=4.7$) ile uygulama sonrası veri işleme alt faktörü son test ortalama puanı ($\bar{X}_{son}=17.83$, $SS_{son}=2.8$) arasında 0.05 anlamlılık düzeyinde anlamlı istatistiksel açıdan bir fark görülmüştür ($t(36) = -3.0$, $p < .05$). Test sonucu hesaplanan etki büyüklüğü ($d=0.48$) bu farkın az düzeyde olduğunu göstermektedir (Green ve Salkind, 2005). Bu durum programlama öğretimi deney grubu öğrencilerinin veri işleme yeterlilikleri üzerinde anlamlı bir etkisinin olduğunu göstermektedir.

Temel programlama alt faktörü ön test ortalama puanı ($\bar{X}_{ön}=12.0$, $SS_{ön}=3.0$) ile uygulama sonrası temel programlama alt faktörü son test ortalama puanı ($\bar{X}_{son}=14.70$, $SS_{son}=2.4$)

arasında 0.05 anlamlılık düzeyinde anlamlı istatistiksel açıdan bir fark görülmüştür ($t(36) = -3.9, p < .05$). Test sonucu hesaplanan etki büyüklüğü ($d=0.65$) bu farkın orta düzeyde olduğunu göstermektedir (Green ve Salkind, 2005). Bu durum programlama öğretimi deney grubu öğrencilerinin temel programlama yeterlilikleri üzerinde anlamlı bir etkisinin olduğunu göstermektedir.

Özgüven alt faktörü ön test ortalama puanı ($\bar{X}_{ön}=14.32, SS_{ön}=2.8$) ile uygulama sonrası özgüven alt faktörü son test ortalama puanı ($\bar{X}_{son}=15.48, SS_{son}=1.9$) arasında 0.05 anlamlılık düzeyinde anlamlı istatistiksel açıdan bir fark görülmüştür ($t(36) = -2.2, p < .05$). Test sonucu hesaplanan etki büyüklüğü ($d=0.35$) bu farkın az düzeyde olduğunu göstermektedir (Green ve Salkind, 2005). Bu durum programlama öğretimi deney grubu öğrencilerinin özgüven yeterlilikleri üzerinde anlamlı bir etkisinin olduğunu göstermektedir. Burada özetle ters yüz sınıf modeli ile ders yapılan grupta algoritma tasarlama, problem çözme, veri işleme, temel programlama, özgüven becerilerinde istatistiksel olarak anlamlı bir artış olduğu belirlenmiştir.

4.4.2. Programlama Öğretiminin Kontrol Grubu için BİDBÖA Alt Faktörlerine İlişkin Bulgular

Yüz yüze programlama eğitimi alan kontrol grubu öğrencilerinin, programlama öğretiminin bilgi işlemsel düşünme becerilerine yönelik öz yeterlik algısı tüm alt faktörleri açısından uygulama öncesi ve sonrası anlamlı bir fark olup olmadığını kontrol etmek için ön test ve son test olarak uygulanan BİDBÖA'ya ait ortalamalar bağımlı örneklem t-testi ile analiz edilmiştir. Uygulanan teste ilişkin bulgular Tablo 18'de özetlenmiştir.

Tablo 18. Kontrol Grubu BİDBÖA Ön Test ve Son Test Tüm Alt Faktörler için Yapılan Bağımlı Örneklem t-Testi Sonuçları

Alt Faktör	Test	N	\bar{X}	S	sd	t	p
Algoritma	Ön Test	42	11.16	2.5	41	-19.1	.00
	Son Test	42	21.69	2.8			
Tasarlama	Ön Test	42	23,81	4,6	41	-4.6	.00
	Son Test	42	27,16	3,8			
Problem	Ön Test	42	14.73	3.4	41	-5.1	.00
	Son Test	42	17.50	2.1			
Çözme	Ön Test	42	11.81	3.3	41	-4.9	.00
	Son Test	42	14.31	2.0			
Veri İşleme	Ön Test	42	14.31	3.0	41	-2.9	.00
	Son Test	42	15.62	1.6			

Tablo 18'de belirtildiği üzere kontrol grubuna uygulanan yüz yüze eğitim ile programlama öğretiminin algoritma tasarlama, problem çözme, veri işleme, temel programlama ve özgüven yeterlilikleri ortalama puanları bağımlı örneklem t testi ile analiz edilmiştir. Bu analiz sonuçları incelendiğinde uygulama öncesi algoritma tasarlama alt faktörü ön test ortalama puanı ($\bar{X}_{ön}=11.16, SS_{ön}=2.5$) ile uygulama sonrası algoritma tasarlama alt faktörü son test ortalama puanı ($\bar{X}_{son}=21.69, SS_{son}=2.8$) arasında 0.05

anlamlılık düzeyinde anlamlı istatistiksel açıdan bir fark görülmüştür ($t(41) = -19.1, p < .05$). Test sonucu hesaplanan etki büyüklüğü ($d=3$) bu farkın çok büyük düzeyde olduğunu göstermektedir (Green ve Salkind, 2005). Bu durum programlama öğretimi kontrol grubu öğrencilerinin algoritma tasarlama yeterlilikleri üzerinde anlamlı bir etkisinin olduğunu göstermektedir.

Problem çözme alt faktörü ön test ortalama puanı ($\bar{X}_{ön}=23.81, SS_{ön}=4.6$) ile uygulama sonrası problem çözme alt faktörü son test ortalama puanı ($\bar{X}_{son}=27.16, SS_{son}=3.8$) arasında 0.05 anlamlılık düzeyinde anlamlı istatistiksel açıdan bir fark görülmüştür ($t(41) = -4.6, p < .05$). Test sonucu hesaplanan etki büyüklüğü ($d=0.75$) bu farkın orta düzeyde olduğunu göstermektedir (Green ve Salkind, 2005). Bu durum programlama öğretimi kontrol grubu öğrencilerinin problem çözme yeterlilikleri üzerinde anlamlı bir etkisinin olduğunu göstermektedir.

Veri işleme alt faktörü ön test ortalama puanı ($\bar{X}_{ön}=14.73, SS_{ön}=3.4$) ile uygulama sonrası veri işleme alt faktörü son test ortalama puanı ($\bar{X}_{son}=17.50, SS_{son}=2.1$) arasında 0.05 anlamlılık düzeyinde anlamlı istatistiksel açıdan bir fark görülmüştür ($t(41) = -5.1, p < .05$). Test sonucu hesaplanan etki büyüklüğü ($d=0.79$) bu farkın orta düzeyde olduğunu göstermektedir (Green ve Salkind, 2005). Bu durum programlama öğretimi kontrol grubu öğrencilerinin veri işleme yeterlilikleri üzerinde anlamlı bir etkisinin olduğunu göstermektedir.

Temel programlama alt faktörü ön test ortalama puanı ($\bar{X}_{ön}=11.81, SS_{ön}=3.3$) ile uygulama sonrası temel programlama alt faktörü son test ortalama puanı ($\bar{X}_{son}=14.30, SS_{son}=2.0$) arasında 0.05 anlamlılık düzeyinde anlamlı istatistiksel açıdan bir fark görülmüştür ($t(41) = -4.9, p < .05$). Test sonucu hesaplanan etki büyüklüğü ($d=0.78$) bu farkın orta düzeyde olduğunu göstermektedir (Green ve Salkind, 2005). Bu durum programlama öğretimi kontrol grubu öğrencilerinin temel programlama yeterlilikleri üzerinde anlamlı bir etkisinin olduğunu göstermektedir.

Özgüven alt faktörü ön test ortalama puanı ($\bar{X}_{ön}=14.31, SS_{ön}=3.0$) ile uygulama sonrası özgüven alt faktörü son test ortalama puanı ($\bar{X}_{son}=15.62, SS_{son}=1.6$) arasında 0.05 anlamlılık düzeyinde anlamlı istatistiksel açıdan bir fark görülmüştür ($t(41) = -2.9, p < .05$). Test sonucu hesaplanan etki büyüklüğü ($d=0.46$) bu farkın az düzeyde olduğunu göstermektedir (Green ve Salkind, 2005). Bu durum programlama öğretimi kontrol grubu öğrencilerinin özgüven yeterlilikleri üzerinde anlamlı bir etkisinin olduğunu göstermektedir. Burada özetle yüz yüze eğitim modeli ile ders yapılan grupta algoritma tasarlama, problem çözme, veri işleme, temel programlama, özgüven becerilerinde istatistiksel olarak anlamlı bir artış olduğu belirlenmiştir.

4.5. Haftalık Değerlendirmelerde Kazanım Bazlı Başarı Puanları

Çalışma süresinde öğrencilere haftalık başarı testleri uygulanmıştır. Bu testlerde tüm programlama kazanımlarına ait deney grubu ve kontrol grubu başarı oranı Tablo 19'da verilmiştir. Sonuçlar haftalık soruların kazanımları olarak gruplandırılmıştır.

Tablo 19. Haftalık Değerlendirmeler için Kazanım Bazlı Başarı Sonuçları

Kazanımlar	Deney Grubu Başarı Oranı (%)	Kontrol Grubu Başarı Oranı (%)
Dışarıdan alınan verileri kullanarak problem çözer.	74.43	68.67
Karakteri verilen yönergeler doğrultusunda hareket ettirir.	99.12	95.11
Döngüleri Kullanır.	52.69	56.38
Koşul Yapılarını Kullanır.	41.75	47.09
Değişken Kullanır.	66.23	75.33
Fonksiyon Kullanır.	59.28	55.79
Rasgele Algoritmasını Kullanır.	66.71	58.12
Çizim Araçlarını Kullanır.	81.20	75.03
Kullanıcıdan Yanıt Alır.	78.39	72.56

Tablo 19’da gösterildiği üzere haftalık kazanımlar gruplandırılarak hem deney hem de kontrol grubu için yüzdesel başarı oranları verilmiştir. Deney grubunun başarı oranlarına bakıldığında dışarıdan alınan verileri kullanarak problem çözmeye grubun %74.43 ‘ü, karakteri verilen yönergeler doğrultusunda hareket ettirmede grubun %99.12’ si, döngüleri kullanmada grubun %52.69’ u, koşul yapılarını kullanmada grubun %41.75’ i, değişkenleri kullanmada grubun %66.23’ ü, fonksiyon kullanmada grubun %59.28’ i, rasgele algoritmasını kullanmada grubun %66.71’ i, çizim araçlarını kullanmada grubun %81.20’ si, kullanıcıdan yanıt almada grubun %78.39’ u başarılı olmuştur.

Kontrol grubunun başarı oranlarına bakıldığında dışarıdan alınan verileri kullanarak problem çözmeye grubun %68.67’ si, karakteri verilen yönergeler doğrultusunda hareket ettirmede grubun %95.11’ i, döngüleri kullanmada grubun %56.38’ i, koşul yapılarını kullanmada grubun %47.09’ u, değişkenleri kullanmada grubun %75.33’ ü, fonksiyon kullanmada grubun %55.79’ u, rasgele algoritmasını kullanmada grubun %58.12’ si, çizim araçlarını kullanmada grubun %75.03’ ü, kullanıcıdan yanıt almada grubun %72.56’ sı başarılı olmuştur.

Her bir kazanım grupları açısından incelendiğinde dışarıdan alınan veriler ile problem çözmeye, karakteri hareket ettirme, fonksiyon kullanımı, rasgele kullanımı, çizim araçları ve kullanıcıdan yanıt alma kazanımlarında ters yüz sınıf modeli ile programlama öğretimi yapılan grubun bu kazanıma ait sorulara vermiş olduğu doğru yanıt yüzdesinin diğer gruba göre daha yüksek olduğunu, döngüler kullanır, değişken kullanımı, koşul yapılarını kullanma kazanımlarında yüz yüze eğitim ile programlama öğretimi yapılan grubun bu kazanıma ait sorulara vermiş olduğu doğru yanıt yüzdesinin diğer gruba göre daha yüksek olduğunu söyleyebiliriz.

4.6. Haftalık Öğrenci Çalışmalarının Değerlendirilmesi

Haftalık öğrenci çalışmaları Dr. Scratch adlı web sayfasında yer alan proje değerlendirme sistemi ile puanlandırılmıştır. En düşük puan 0, en yüksek puan 21’ dir. Verilen puanlar deney ve kontrol grubunun her birisi için ortalaması alınarak analiz edilmiştir. Bu çalışmalara ait grup ortalamaları Tablo 20’ de gösterilmiştir.

Tablo 20. Haftalık Öğrenci Çalışmalarının Dr. Scratch ile Elde Edilen Puanların Ortalamaları

Hafta	Etkinlik	Deney Grubu Ortalama Puanı	Kontrol Grubu Ortalama Puanı
1	Sahne ve Karakter Özellikleri Karakteri Boyutlandırma	5	5
2	Karakter Zıplatma Hareketi Karakter Yürüyüşü Animasyonu	8	7
3	Fare ile Etkileşim Klavye ile Etkileşim	12	10
4	Penaltı – Gol	8	9
5	Havai Fişek	12	10
6	Biraz da geometri	11	10
7	Analog saat	6	8

Tablo 20’de belirtildiği üzere öğrencilerin bu araştırma süresince yapmış oldukları etkinliklerin Dr. Scratch uygulaması tarafından verilen puanların ortalaması alınmıştır. En düşüğü 0 ve en yükseği 21 olan puanlamada deney grubu için sahne ve karakter özellikleri, karakter boyutlandırma etkinliklerinin grup ortalama puanı 5, karakter zıplatma hareketi, karakter yürüyüş animasyonu etkinliklerinin grup ortalama puanı 8, fare ile etkileşim, klavye ile etkileşim etkinliklerinin grup ortalama puanı 12, penaltı-gol etkinliğinin grup ortalama puanı 8, havai fişek etkinliğinin grup ortalama puanı 12, birazda geometri etkinliğinin ortalama puanı 11, analog saat etkinliğinin grup ortalama puanı 6 dir.

Kontrol grubu için sahne ve karakter özellikleri, karakter boyutlandırma etkinliklerinin grup ortalama puanı 5, karakter zıplatma hareketi, karakter yürüyüş animasyonu etkinliklerinin grup ortalama puanı 7, fare ile etkileşim, klavye ile etkileşim etkinliklerinin grup ortalama puanı 10, penaltı-gol etkinliğinin grup ortalama puanı 9, havai fişek etkinliğinin grup ortalama puanı 10, birazda geometri etkinliğinin ortalama puanı 10, analog saat etkinliğinin grup ortalama puanı 8 dir.

Grupların aynı etkinliklerde almış oldukları puanlar yorumlandığında karakteri zıplatma ve karakteri yürütme animasyonu, fare ile etkileşim, klavye ile etkileşim, havai fişek, biraz da geometri çalışmalarında deney grubunun kontrol grubundan daha fazla puan ortalamasına sahip olduğunu söyleyebiliriz. Penaltı-gol, analog saat çalışmalarında kontrol grubunun deney grubundan daha fazla puan ortalamasına sahip olduğunu söylenebilir. Sahne ve karakter özellikleri, karakteri boyutlandırma etkinliklerinde her iki grubunda puan ortalaması eşit olduğunu söylenebilir.

4.7. Uygulama Sonrası Öğrencilerin Scratch Programlamayı Farklı Stratejiler ile Öğrenme Konusundaki Görüşler

Öğrencilerin Bilge Kunduz Uluslararası Enformatik ve Bilgi İşlemsel Düşünme Soruları sonuçlarından aldıkları puanlara göre yüksek ve düşük olarak tespit edilen her şubeden 6 kişi olmak üzere toplam 24 öğrenci ile odak grup görüşmesi yapılmıştır. Öğrencilerle

görüşmeler yapılırken onların izinleriyle sesleri kaydedilmiş ve daha sonra veriler nicel bulgulara destek olmak amacıyla içerik analizi ile değerlendirilmiştir.

4.7.1. Scratch Programlama Dilini Farklı Stratejiler ile Öğrenme Konusundaki Görüşler

Yüz yüze eğitim ile ders yapılan gruptaki öğrencilere “Scratch programlama dilini derste öğretmen anlatımı ile öğrenme konusunda ne düşünüyorsun?”, ters yüz sınıf modeli ile ders yapılan gruba ise “Scratch programlama dilini internet üzerinden kendi kendine öğrenme konusunda ne düşünüyorsun?” soruları sorulmuştur. Sorulara verdikleri yanıtlardan elde edilen görüşler ve kaç öğrencinin benzer düşünceyi yansıttığını gösteren frekans değerleri Tablo 21’ de gösterilmiştir.

Tablo 21. Scratch Programlama Dilini Farklı Stratejiler ile Öğrenme Konusundaki Görüşler

Yüz Yüze Eğitim Alan Öğrenciler için Ortaya Çıkan Görüşler	Görüş	Frekans	Ters Yüz Sınıf Modeli ile Eğitim Alan Öğrenciler için Ortaya Çıkan Görüşler	Görüş	Frekans
Öğretmen anlatımı ile iyi öğrenme		11	Kendi kendine öğrenme daha iyi		11
Anlık geri bildirim ve yardım		10	Eğlenceli		5
Bireysel olarak öğrenme güçlüğü		1	Yaratıcılığı artırıyor		1
			Öğretmen anlatımı daha iyi		1
			Öğretmen anlatsa daha az kişi anlayacak		1
			Kendi kendime öğrenmekte zorlanıyorum		1

Yüz yüze eğitim alan öğrencilerin Scratch programlama dilini öğrenimi konusunda ortaya çıkan görüşler incelendiğinde 12 öğrencinin 11’i öğretmen anlatımı ile daha iyi öğrendiğini düşünmekte, 10 öğrenci anlık geri bildirim ve yardımı ile öğrenme süreçlerinin daha iyi olduğunu ve son olarak bir öğrenci de bireysel olarak öğrenme güçlüğü yaşadığını ifade etmiştir.

Ters yüz sınıf modeli eğitim alan öğrencilerin kendi kendine öğrenme konusunda ortaya çıkan görüşler incelendiğinde 12 öğrenciden 11’i kendi kendine öğrenmenin daha iyi olduğunu, 5’i bu yöntem ile öğrenmenin daha eğlenceli olduğunu, 1 öğrenci bu yöntem ile öğrenmenin yaratıcılığı arttırdığını, 1 öğrenci öğretmen anlatımının daha iyi olduğunu, 1 öğrenci Öğretmen anlatımında daha az öğrencini anlayacağını, son olarak bir öğrenci de kendi kendine öğrenmede zorluk yaşadığını ifade etmiştir.

Öğretmen anlatımı ile öğrenme konusunda bir öğrenci düşüncelerini aşağıdaki biçimde ifade etmiştir.

Ö1: “Kendi kendime öğrenmekte zorlanıyorum. Birisi bana gösterdiğinde çok daha iyi öğreniyorum.”

Ters yüz sınıf modeli ile öğrenme konusunda bir öğrenci düşüncelerini aşağıdaki biçimde ifade etmiştir.

Ö2: “Kendi kendine öğrenme yaratıcılığı arttırıyor. Çünkü aklıma yeni fikir geldiğinde nasıl yapılabileceğini tekrar izleyebilme imkanım oluyor.”

Aynı grupta başka bir öğrenci ise

Ö3: “Tekrar edebilme olanağı sağladığından internet üzerinden öğrenme daha kolay. Öğretmen anlatımında anlatılanı aynı şekilde tekrar edemeyeceğimizden daha az kişi konuyu öğrenir.” şeklinde düşüncelerini ifade etmiştir.

Yukarıda belirtilen görüşler incelendiğinde genel olarak her öğrenci öncelikli olarak kendi grubunda yapılan yöntemi benimsediği görülmüştür. Yüz yüze ders işleyen gruptaki öğrenciler öğretmen anlatımın daha iyi olduğunu belirtirken ters yüz sınıf modeli ile ders işleyen öğrencilerin internet destekli kendi kendine çalışabileceği bir ortamda öğrenmenin daha kolay olduğunu belirtmişlerdir. Her iki grupta ki görüşlere bakıldığında hangi yöntem ile ders işlediklerinin farkında oldukları ve bu yöntemlerin olumlu ve olumsuz olarak nitelendirilebilecek özelliklerini ifade ettikleri görülmektedir.

4.7.2. Scratch Programlama Dilini Farklı Stratejiler ile Öğrenme Konusunda Yaşanılan Zorluklar ile İlgili Görüşler

Geleneksel sınıf modeli ile ders yapılan gruptaki öğrencilere “Scratch programlama dilini sınıf ortamında öğrenirken yaşadığın zorluklar nedir?”, ters yüz sınıf modeli ile ders yapılan gruba ise “Scratch programlama dilini internet üzerinden kendi kendine öğrenirken yaşadığın zorluklar nedir?” soruları sorulmuştur. Sorulara verdikleri cevaplardan elde edilen görüşler ve kaç öğrencinin benzer düşünceyi yansıttığını gösteren frekans değerleri Tablo 22’de gösterilmiştir.

Tablo 22. Scratch Programlama Dilini Farklı Stratejiler ile Öğrenme Konusunda Yaşanılan Zorluklar ile İlgili Görüşler

Yüz Yüze Eğitim Alan Öğrenciler için Ortaya Çıkan Görüşler		Ters Yüz Sınıf için Ortaya Çıkan Görüşler	
Görüş	Frekans	Görüş	Frekans
Anlamadığım için yardım alma	5	Anlık Geri Bildirim ve Yardım	12
Sınıf içi etkenler (Dikkat dağıtan öğrenciler)	4	Teknik Sıkıntılar (İnternet Hızı, eklentiler)	4
Dersin hızına yetişememe	4	Problem Yaşamıyorum	1
Öğretmen anlatırken konuyu kaçırma ve anlamama	3		
Kişisel zorluklar (Tahtayı görememe)	2		
Zorlanmadım	2		
Hızlı öğrendiğimden sınıfın geride kalması	1		

Geleneksel sınıf kapsamında Scratch programlama dili öğrenirken yaşanan zorluklar konusunda ortaya çıkan görüşler incelendiğinde 5 öğrencinin anlamadığı yerlerde yardım alarak yapabildiklerini, 4 öğrencinin sınıf içerisinde dikkat dağıtan öğrenciler olduğundan dolayı dersin bazı kısımlarını anlamadıklarını ifade etmişlerdir. 4 öğrenci dersin hızına yetişemediklerini, 3 öğrenci ise öğretmen konuyu anlatırken aynı anda yapmaya çalıştıklarında konuyu kaçırdıkları için anlayamadıklarını ifade etmişlerdir. 2 öğrenci tahtayı net göremediklerinden dolayı sınıf içinde öğrenmede zorluk yaşadıklarını 1 öğrencinin kendisinin sınıftan daha hızlı öğrendiğinden bu yüzden dersin kendisi için daha yavaş geçtiğinden dolayı zorlandığını, 2 öğrencinin ise derste herhangi bir zorluk yaşamadığını ifade etmişlerdir.

Ters yüz sınıf kapsamında Scratch programlama dili öğrenirken yaşanan zorluklar konusunda ortaya çıkan görüşler incelendiğinde 12 öğrencinin kendi kendine öğrenirken anlamadıkları yerlerde anlık olarak destek olamadıkları için zorlandıklarını, 4 öğrenci bilgisayarda konu anlatımlarının bulunduğu sayfalardaki bazı eklentileri çalıştıramadıkları veya çalıştırmak için çok zaman kaybettiklerinden dolayı zorlandıklarını ifade etmişlerdir. Bir öğrenci ise herhangi bir zorluk yaşamadığını ifade etmiştir.

Geleneksel sınıf grubundan bir öğrenci görüşlerini şu şekilde ifade etmiştir:

Ö1: “Benim öğrenme hızım sınıftaki arkadaşlarımdan daha hızlı. Konuyu çok hızlı anlıyorum ve yapıyorum. Konunun devamını öğrenmem için sınıfın benim anladığım yere kadar gelmesini bekliyorum ve bu durumda derste çok sıkılıyorum.”

Ters yüz sınıf grubundan bir öğrenci görüşlerini şu şekilde ifade etmiştir:

Ö2: “Bilgisayarımdaki teknik sıkıntılardan dolayı, örneğin eklentinin yüklenememesi, internetin yavaş olması durumlarında çalışmak için komşuma gitmek zorunda kalıyorum. Bunu devamlı yaptığımda onları rahatsız ettiğimi düşünüyorum.”

Yukarıda ifade edilen yorumlara ek olarak araştırmacının tutmuş olduğu günlüklerde yer alan notlar ile benzerlik göstermektedir. Araştırmacının aldığı notlarda öğrencilerin Flash programı güncellemesi yüzünden etkinliğe giremedikleri, etkinliğin detaylarının fazla olması ve kazanımın tek başına anlaşılmasının zor olduğundan öğrencilerin konuyu anlamadan geldikleri gibi ifadelerin yer aldığı buda öğrenci görüşlerini desteklediği görülmüştür.

4.7.3. Scratch Programlama Dilini Daha İyi Öğrenme Konusunda Görüşler

Her iki grup öğrencilerine “Scratch programlama dilini daha iyi öğrenmek için ne yapılmasını önerirsin?” sorusu sorulmuştur. Sorulara verdikleri cevaplardan elde edilen görüşler ve kaç öğrencinin benzer düşünceyi yansıttığını gösteren frekans değerleri Tablo 23’ te gösterilmiştir.

Tablo 23. Scratch Programlama Dilini Daha İyi Öğrenme Konusundaki Görüşler

Görüşler	Frekans
Keşfederek öğrenme ortamının oluşturulması	5
İnternetteki derslerin içerikleri daha açıklayıcı olması.	4
Önerim yok	4
Ders dışında video ile tekrar ortamının olması	3
Daha çok örnek yapılması	3
Anlık yardım olması	1
Konuşanların dinlemesi	1
Scratch'te kod arama özelliğinin olması	1

Scratch programlama dilini daha iyi öğrenmek için ne yapılması konusunda görüşler incelendiğinde 5 öğrencinin Scratch’i keşfederek öğrenmenin daha iyi olacağını, 4 öğrencinin derste var olan içeriklerin daha detaylı ve açıklayıcı olduğunda öğrenmenin daha iyi olacağını ifade etmiştir. Üç öğrenci ders dışında herhangi bir yerde ders içeriklerine erişmenin iyi olacağını, 3 öğrenci daha fazla örnek yapmanın Scratch’in daha iyi öğrenileceğini ifade etmiştir. Bir öğrenci interaktif ortamda anlık yardım alınabilmesinin öğrenmenin daha kolay olacağını ifade ederken, 1 öğrenci, sınıfta konuşanların olmaması öğrenmenin daha iyi olacağını belirtmiştir. Bir öğrenci Scratch’te kod arama bölümünün olması gerektiğini ifade ederken, 4 öğrenci ise hiçbir öneri ifade etmemiştir.

Odak grup görüşmesi yapılan öğrencilerden 1’i Scratch’i daha iyi öğrenmek için görüşlerini şu şekilde ifade etmiştir.

Ö6: “Bence Scratch’i bir öğretmenden veya internetten video ile öğrenmek yerine problemler verilerek ve adım adım çözüm basamaklarından giderek kodları keşfetmemiz öğrenmemiz açısından daha iyi ve kalıcı olacaktır.”

4.7.4. Scratch Programlama Ortamında Program Yazmanın Kazandırdıkları Konusunda Görüşler

Her iki grup öğrencilerine “Scratch programlama ortamında program yazmak sana neler kazandırmış olabilir?” sorusu sorulmuştur. Sorulara verdikleri cevaplardan elde edilen görüşler ve kaç öğrencinin benzer düşünceyi yansıttığını gösteren frekans değerleri Tablo 24’ te gösterilmiştir.

Tablo 24. Scratch Programlama Ortamında Program Yazmanın Katkıları Konusunda Görüşler

Görüşler	Frekans
Oyun ve animasyon yapmak için kodlama becerisinin gelişmesi	24
Mesleki anlamda karar vermede etkili olması	5
Bilgisayarı ve interneti kullanma becerilerinin arttırması	4
Başka derslerde kullanılması	3
Daha kolay karar vermede yardımcı olması	2
Çok eğlenceli olması	2
Hayal kurmayı sağlaması	1

Scratch programlama ortamında program yazmanın kazandırdıkları konusunda görüşler incelendiğinde 24 öğrencinin oyun ve animasyon geliştirmek için gerekli kodlama becerilerinin arttığını belirtmiştir. Beş öğrenci mesleki anlamda karar vermemde etkili olduğunu belirtmiştir. Dört öğrenci bilgisayar ve interneti daha verimli kullandığına dair görüş bildirmişlerdir. Üç öğrenci öğrendiklerini başka derslerde kullandıklarını, 2 öğrenci karşılaştığı problemlerde daha hızlı karar verdiklerini ifade etmişlerdir. İki öğrenci kodlama öğrenmenin çok eğlenceli olduğunu belirtirken, 1 öğrenci istediği oyunun hayali kurup ve bunu yazabileceğini ifade etmiştir.

Odak grup görüşmesi yapılan öğrencilerden 2 tanesi Scratch programlama ortamında kod yazmanın kazandırdıkları ile ilgili şu görüşleri ifade etmişlerdir.

Ö8: “Scratch te kod yazmak ileride seçeceğim meslek anlamında bana fikir sağladı. İleride yazılımcı, mühendis olup yazılımlar animasyonlar geliştirmek istiyorum. Önceden bunu yapabileceğimi sanmıyordum. Ama artık yapabileceğime inanmaya başladım.”

Ö17:” Scratch ile animasyon yazmak bana hayal kurmamı sağladı. Kafamda bir oyun hayalim vardı. İlk başta basit olarak düşünmüştüm. Fakat sonra kodlamayı öğrendikçe oyun içerisinde farklı nesnelere koymayı hayal ediyorum. Bu oyunumu en kısa sürede yazıp arkadaşlarıma oynatmak istiyorum.” şeklinde ifade etmiştir.

4.7.5. Scratch Programlama Ortamında Programlama ile İlgili Öğrenilen Kavramlar Hakkında Görüşler

Her iki grup öğrencilerine “Scratch programlama ortamında programlama ile ilgili hangi kavramları öğrendiğini düşünüyorsun?” sorusu sorulmuştur. Sorulara verdikleri

cevaplardan elde edilen görüşler ve kaç öğrencinin benzer düşünceyi yansıttığını gösteren frekans değerleri Tablo 25’de gösterilmiştir.

Tablo 25. Scratch Programlama Ortamında Programlama ile İlgili Öğrenilen Kavramları Hakkında Görüşler

Görüşler	Frekans
Scratch'te programlama araçları (Döngü-Değişken-Özel Taşlar-Hareket-Çizim-Kılık değiştirme-Konuşturma-Sahne, Kukla Ayarları)	24
Algoritma	2

Scratch programlama ortamında programlama ile ilgili öğrenilen kavramları hakkında görüşler incelendiğinde 24 öğrencinin tamamı döngü, değişken, hareket, çizim, kılık ayarları, sahne ve kukla ayarları, özel taşları öğrendiklerini ifade etmişlerdir. Son olarak 2 öğrenci bu görüşlerin yanı sıra algoritma kurmayı öğrendiklerini ifade etmişlerdir.

5. TARTIŞMA ve SONUÇ

Çalışmanın amacı 5. sınıf öğrencilerinin yüz yüze eğitim ve ters yüz sınıf modeli ile Scratch programlamayı öğrenmelerinin, öğrencilerin bilgi işlemsel düşünme becerilerine olan etkisini araştırmaktır. Bunun için programlama öğretiminin öğrencilerin bilgi işlemsel düşünme becerilerine etkisi, uygulama öncesi ve sonrası bilgi işlemsel düşünme becerilerine yönelik öz yeterlik algıları ölçülmüş, uygulama sonrasında öğrencilerin bulunduğu gruplara uygulanan öğretim tekniklerinin öğrenci gözünde nasıl bir etki bıraktığına dair öğrenci görüşleri alınmıştır.

Araştırmanın amaçlarından birisi olan iki farklı öğretim yönteminin öğrencilerin bilgi işlemsel düşünme becerilerine etkisini ölçmek amaçlı uygulama öncesi ters yüz sınıf modeli ve yüz yüze eğitim gruplarına uygulanan Bilge Kunduz Uluslararası Enformatik ve Bilgi İşlemsel Düşünme Sorularından oluşturulan ön test puan ortalamaları arasında istatistiksel olarak anlamlı bir fark olmadığı tespit edilmiştir. Her iki grubun ön test başarı puanları ortalamalarına bakıldığında matematiksel olarak her iki grubun ortalamaları birbirine yakın olduğu görülebilir. Bu sonuca göre ters yüz sınıf modeli ve yüz yüze eğitim gruplarının enformatik ve bilgi işlemsel düşünme başarıları arasında bir fark olmadığı söylenebilir.

Uygulama sonrası ters yüz sınıf modeli ve yüz yüze eğitim gruplarına uygulanan Bilge Kunduz Uluslararası Enformatik ve Bilgi İşlemsel Düşünme Sorularından oluşturulan son test puan ortalamaları arasında istatistiksel olarak anlamlı bir fark olmadığı tespit edilmiştir. Her iki grubunda son test başarı ortalamaları ön test ortalamalarından yüksektir. Yüz yüze ve ters yüz sınıf modeli gibi farklı öğretim yöntemlerinin öğrencilerin bilgi işlemsel düşünme becerileri başarılarına herhangi bir etkisi olmadığı söylenebilir. Ancak ortalamalar matematiksel olarak incelendiğinde ters yüz sınıf modeli ve yüz yüze eğitim gruplarının ön test puan ortalamaları arasındaki fark yüz yüze eğitim grubunun lehine 3 puan iken son test puan ortalaması arasındaki fark ters yüz sınıf modeli grubunun lehine yaklaşık 7 puandır. Ters yüz sınıf modeli grubuna uygulanan ön test ve son test ortalama puan farkı yaklaşık 20 puan arttığı gözlenirken, yüz yüze eğitim grubuna uygulanan ön test ve son test puan farkı yaklaşık 10 puan artmıştır. Bu sonuçlara göre ters yüz sınıf modeli ile yapılan derste öğrencilerin ortalamaları matematiksel olarak yüz yüze ders yapan öğrencilerin ortalamalarından daha fazla yükseldiği söylenebilir. Bu durum öğrenci odak görüşmeleri sonucu ortaya çıkan birçok ifade ile desteklenebilir. Öğrencilerin ters yüz sınıf modeli ile öğrenmede ki görüşleri incelendiğinde kendi kendine ve kendi hızlarına göre öğrenme, tekrar edebilme kavramlarının üzerinde durmuşlardır. Bu durum öğrencinin bilgiyi daha organize ve kalıcı olarak öğrenmesi ile ilişkilendirilebilir. Karaca (2016), teknoloji ile öğrenme üzerine yapmış olduğu çalışmada ters yüz sınıf modeli ile programlama öğrenen öğrencilerin kendi kendine öğrenme düzeyleri yüz yüze sınıf ortamında öğrenme gerçekleştiren öğrencilerden daha yüksek olduğunu ifade etmiştir. Bu sonuç araştırmada ortaya çıkan ters yüz sınıf modeli grubunun başarı puanının yüz yüze eğitim grubuna göre daha fazla yükselmesi sonucunu desteklemektedir.

Ters yüz sınıf modeli ile programlama öğretimi yapılan ters yüz sınıf modeli grubunun ön test ve son test ortalamaları arasındaki farkın daha büyük çıkması daha önce öğrenci

görüşlerinde belirtildiği gibi öğrencilerin kendi kendine öğrenmeleri, daha sonra tekrar edebilme imkanlarının bulunması ve öğrenmeyi kendi hızlarına göre gerçekleştirmesi bu gruptaki öğrencilerin ortalamalarında daha büyük bir fark olmasına neden olduğu söylenebilir. Yüz yüze eğitim ile programlama öğretimi yapılan yüz yüze eğitim grubunun son test puan ortalamasının ön test puan ortalamasına göre daha yüksek çıkması öğrencilerin anlık geri bildirim, yardım alabilme ve en önemli unsur olan öğretmen anlatımının daha iyi olduğunu görüşü ile desteklenebilir. Programlamaya yönelik her kazanım öğretmen tarafından verildiğinden ve öğretmenin kendi becerisi ile bu kazanımları öğretmiş olması birçok kazanımın kalıcı bir şekilde öğrenilmesini sağladığı düşünülebilir. Ters yüz sınıf ile programlama öğretimi yapılan ters yüz sınıf modeli grubunun son test ortalamasının ön test puan ortalamasından daha yüksek çıkması bu gruptaki öğrencilerin kendi kendine öğrenmenin daha iyi olduğu, eğlenceli olduğu, yaratıcılığı arttırdığını belirten görüşleri ile desteklenmektedir. Daha çok tekrar edebilme imkânı sunan bu yöntem öğrencilerin kazanımları kavramasında faydalı olduğu söylenebilir. Bu konuda Walsh (2015), ters yüz sınıf modelinin öğrenciler açısından değerlendirilmesi çalışmasındaki kendi kendine öğrenmenin daha iyi olduğu ve bu yöntemin keyifli olduğunu belirtmiştir. Bu çalışma yapılan araştırma sonucunu desteklediği görülmektedir.

Bu araştırma sonucunda yöntemler arası farklılık çıkmamasına rağmen bazı araştırmalarda ters yüz sınıf modelinin daha başarılı bir sonuç verdiği belirtilmiştir. Karaca (2016), yaptığı çalışmada programlama öğretiminde aynı yöntemle ders işlemiş ve ters yüz sınıf modeli ile ders işleyen grubun daha başarılı olduğunu belirtmiştir. Tugun, Uzunboylu, Özdamli, (2017), yapmış oldukları çalışmada programlama öğretiminde ters yüz sınıf ve yüz yüze eğitim ile iki farklı gruba uyguladıkları çalışmada ters yüz sınıf grubunun daha başarılı olduğunu belirtmişlerdir.

Araştırma sırasında bazı kazanımları vermede kullanılan yöntemlerin yetersiz kaldığı durumlar ortaya çıkmıştır. Haftalık değerlendirmelerde başarı yüzdelere bakıldığında döngüleri kullanır, koşul yapılarını kullanır, değişkenleri kullanır kazanımlarında ters yüz sınıf modeli ile eğitim alan grubun ortalamaları daha düşük çıkmıştır. Bu kazanımlar programlamada öğretilmesi karmaşık ve öğrencinin tam olarak öğrenmesi zaman alan kavramlardır. Bu tip kazanımların verilmesinde yüz yüze anlatım daha etkili olacağı düşünülmektedir. Çünkü kazanımın ilk defa anlatılması esnasında araştırmacı konuya olan hakimiyeti ile sınıf ortamında öğrencileri konuya daha iyi odaklayıp anlamalarını sağlayacağı, anlık geri bildirimler ve yardımlar ile öğrenmeyi destekleyip kalıcılığını sağlayacağı düşünülmektedir.

Ters yüz sınıf modeli ve yüz yüze eğitim gruplarına haftalık olarak yapılan değerlendirmelerde birçok öğrencinin o haftaki kazanımları öğrenememiş olduğu, başarılarının diğer öğrencilere kıyasla düşük olduğu gözlenmiştir. Bu durumu öğrencilerin bulunduğu gruptaki yöntemle ilişkilendirilebilir. Odak grup görüşmelerinde yüz yüze eğitim grubu öğrencileri derste yaşadıkları zorlukları, anlamadıklarında daha sonrasında tekrar edememe, sınıf içerisinde dikkat dağılması ve kendi öğrenme hızı ile dersin hızının birbirine uymaması görüşlerini belirtmişlerdir. Ters yüz sınıf modeli grubu öğrencileri ise çoğunlukla anlık geri bildirim ve yardım alamama, teknik sıkıntılardan ve

birinin yardımı ile öğrenebildiklerinden dolayı ilgili hafta verilen kazanımları öğrenemediklerini ifade etmişlerdir. İfade edilen bu görüşlerden çıkan sonuçlar değerlendirildiğinde öğrencilerin haftalık değerlendirme performanslarına olumsuz yansıdığı söylenebilir. Ayrıca araştırmacının tutmuş olduğu günlüklerdeki notlara bakıldığında her iki gruptaki öğrencilerin de teknik sıkıntılar, etkinliğin zorluğu, evde çalışmadan okula gelmiş olmaları, bazı kazanımların verilmesinde zorlanıldığı buna bağlı olarak haftalık değerlendirmelerde istenilen başarıyı gösterememiş oldukları görülmüştür. Tugun, Uzunboylu, Ozdamli, (2017), yapmış oldukları çalışmada öğrencilerin programlama öğretimine ilişkin görüşlerini almışlardır. Ters yüz sınıf öğrencilerinin yaşadığı sıkıntılara bakıldığında internet hızının yavaşlığı, tek başına çalışmama, yöntemin yeni bir yöntem olduğu ve nasıl çalışacağını bilememe gibi görüşler bildirmişlerdir. Yapılan araştırmada ortaya çıkan görüşler bu çalışmadaki öğrenci görüşlerinin birçoğunu desteklediği görülmektedir.

Öğrenci başarılarını artırmaya yönelik, Scratch programlama dilini daha iyi öğrenme ile ilgili öğrenci görüşleri incelendiğinde ortak olarak izleyerek öğrenmeden ziyade bir problem üzerinden keşfederek öğrenme, internet üzerinden yapılan derslerin daha açıklayıcı olması ve yüz yüze eğitim alan öğrencilerin ders dışında tekrar ortamı olması yönünde yoğunlaştığı söylenebilir.

Araştırmanın başka bir amacı olan öğrencilerin programlamayı öğrenmelerinin, bilgi işlemsel düşünme becerilerine etkisine bakıldığında her iki gruba kendi içerisinde yapılan ön test ile son testler arasında anlamlı farklılıklar ortaya çıkmıştır. Daha önce hiç programlama dersi almayan öğrencilerin araştırma öncesinde yapılan ön test ve araştırma süresince programlama dersi almaları sonrasında yapılan son test puanları arasında puan farkları oluşmuştur. Ters yüz sınıf modeli ile programlama öğretimi gerçekleştiren ters yüz sınıf modeli grubuna ait öğrencilerin bilgi işlemsel düşünme becerileri ön test ve son test puan ortalamalarındaki fark istatistiksel olarak anlamlıdır. Aynı şekilde yüz yüze eğitim ile programlama öğretimi gerçekleştiren yüz yüze eğitim grubuna ait öğrencilerin bilgi işlemsel düşünme becerileri ön test ve son test puan ortalamalarındaki fark da istatistiksel olarak anlamlıdır. Buradaki sonuç yöntemden bağımsız sadece programlama öğretiminin öğrencilerin belirtilen alanlardaki değişimini göstermektedir. Grupların ortalamalarının yükselmesini öğrencilerin programlama öğrenme konusundaki istek ve olumlu düşünceleri de desteklemektedir. Öğrenciler Scratch ile programlama öğrenmenin eğlenceli olduğunu, kodlama ve algoritma becerilerinin arttığını, daha kolay ve hızlı karar verdiklerini aynı zamanda bilgisayar kullanma becerilerinin de arttığını ifade etmişlerdir. Bu ifadelerden anlaşılacağı gibi öğrenciler hem eğlenerek hem de öğrenerek bu çalışmayı yaptıkları söylenebilir. Ayrıca bu öğrenciler Scratch ile programlamaya ait döngü-değişken, özel taşlar, algoritma gibi birçok kavramı öğrendiklerini ifade etmişlerdir. Bu öğrenmiş oldukları kavramlar bilgi işlemsel düşünme becerilerine olumlu olarak etkisi olduğu söylenebilir. Çatlak, Tekdal ve Baz (2015), yapmış oldukları Scratch programlama öğretimi doküman inceleme çalışmasında Scratch'in kolay, ilginç, merak uyandırıcı ve keyifli bir uygulama olduğunu, algoritma ve programlama temellerini öğretiminde etkili olduğunu yaratıcı düşünebilmeyi sağladığını, motivasyonu arttırdığı yönünde görüşler elde etmişlerdir. Bu araştırma sonuçları öğrencilerin ifadelerini desteklemektedir.

Araştırması yapılan programlama öğretiminin öğrencilerde bilgi işlemsel düşünme becerisine etkisini araştıran birçok çalışmaya ulaşılmıştır. Fadjo'nun (2012), ortaokul öğrencilerine Scratch kullanımının bilgi işlemsel düşünme becerilerine etkisini incelemek üzerine yapmış olduğu çalışmanın sonucunda Scratch'in bilgi işlemsel düşünme becerisi geliştirmede etkili olduğunu vurgulamıştır. Shin ve Park (2014), yaptıkları çalışmada Scratch ile problem çözme etkinlikleri yapılmıştır. Araştırma sonuçlarına göre Scratch'in problem çözme ve bilgi işlemsel düşünme becerine doğrudan etki ettiğini belirtmişlerdir. Bu sonuçlar araştırmacının almış olduğu sonucu desteklemektedir. Tüm bu analizlerden elde edilecek sonuç programlama öğretimi öğrencilerin bilgi işlemsel düşünme becerilerine pozitif olarak etki ettiği söylenebilir.

Araştırmada kullanılan BİDBÖA alt faktörleri değerlendirildiğinde hem ters yüz sınıf modeli hem yüz yüze eğitim grubunun algoritma tasarlama, problem çözme yeterliği, temel problem çözme yeterliği, özgüven yeterliği ve veri işleme yeterliği faktör puanları ön test ve son test sonuçları istatistiksel açıdan incelendiğinde pozitif olarak anlamlı bir fark ortaya çıkmıştır. Bu sonuçlara göre programlama öğretimi öğrencilerin tüm bu yeterliklerine olumlu açıdan etki ettiği söylenebilir. Belirtilen sonuçlar öğrencilerin odak grup görüşmelerinde belirttikleri Scratch programlama ortamında kod yazmanın kazandırdıkları ile ilgili öğrenci düşüncelerinden oyun ve animasyon yapmak için kodlama becerilerinin geliştiği, bu tarz oyun ve animasyonları yapabildikleri yönünde olduğu ifadeler tarafından da desteklenmektedir. Birçok öğrenci mesleki anlamda karar vermelerinde etkili olduğunu, kendilerine bu konuda güven geldiğini, kodlama ile hayallerindeki oyunları yapabileceklerine inandıklarını söylemişlerdir. Bu ifadeler ile özellikle algoritma tasarlama, veri işleme, temel programlama, problem çözme ve özgüven faktörlerinin öğrenciler tarafından benimsendiğini destekleyen ifadelerdir.

Kafai, Lee, Searle, Fields, Kaplan ve Lui, D. (2014), elektronik programlamayı kullanarak bilgi işlemsel düşünme becerilerini geliştirmeyi amaçladıkları çalışmalarında öğrenciler ile yapmış oldukları görüşmelerde bilgisayarı daha önce kodlama amaçlı kullanmadıklarını ve yapamayacaklarını düşündüklerini fakat gerekli eğitimi aldıktan sonra programlama yapabildiklerini, bu işi yapabileceklerine dair özgüvenlerinin arttığını belirtmişlerdir. Kasalak (2017), yapmış olduğu çalışma kodlama etkinliklerinin, öğrencilerin programlamaya ilişkin özyeterlik algıları üzerindeki etkisini incelemiştir. Scratch ile uygulama geliştiren öğrencilerin programlamaya ilişkin öz yeterlik algılarının daha yüksek olduğunu belirtmiştir. Bu iki çalışmanın sonucu, bu araştırma sonuçlarını desteklediği görülmektedir.

Özcan, Ergün, Köse, Emir ve Gezgin (2017), yapmış oldukları çalışmada Scratch programının kullanımına yönelik öğrenci görüşlerini almışlardır. Öğrenci görüşleri incelendiğinde programlama eğitimine faydalı olduğu, programlama mantığının geliştiği, eğlenceli ve iyi olması, yaratıcılığı geliştirdiği, algoritmayı daha iyi anlamaya katkı sağladığı gibi görüşlerin bu çalışmadaki öğrenci görüşleri ile tutarlılık sağladığı görülmektedir. Bu sonuçlar yapılan çalışmada programlama öğretim aracı olarak Scratch'in tercih edilmesini desteklemektedir.

6. ÖNERİLER

5. sınıf öğrencilerinin yüz yüze eğitim ile teknoloji destekli öğrenme ortamlarında olmak üzere iki farklı öğrenme ve öğretme stratejisi ile blok tabanlı bir programlama aracı olan Scratch programlamayı öğrenmelerinin ve programlama öğretiminin öğrencilerin bilgi işlemsel düşünme becerilerine etkisini araştırmak amacı ile yapılan çalışmanın sonuçları doğrultusunda araştırmacılara, öğretmenlere ve teknoloji destekli eğitim ortamı geliştiren yazılımcılara yönelik öneriler aşağıda sunulmuştur.

Gelecekte Çalışma Yapacak Araştırmacılara Yönelik Öneriler

1. Bu araştırma kapsamında incelenen çalışmalar yüz yüze eğitim yöntemi ve ters yüz sınıf yöntemi, Scratch ile programlama ile sınırlı tutulmuştur. Bu çalışma gelecekte bu yöntemler ile Scratch ile programlama öğretiminin öğrencilerde bilgi işlemsel düşünme becerileri konusunda çalışma yapacak araştırmacılara ışık tutacağı düşünülmektedir. İlerde başka öğretim yöntemleri ve teknikleri ile ortaokul öğrencilerine programlama öğretimi yapılırsa öğretim yöntemlerine ilişkin farklı bakış açıları geliştirilebilir.
2. Araştırmada iki farklı gruba iki farklı yöntem ile ders işlenmiştir. İleride yapılacak çalışmalarda aynı gruba iki farklı yöntem ile ders işlenip, analiz edilmesi durumunda hangi yöntem ile programlama eğitiminin verilmesi daha iyi olabileceği araştırılabilir.
3. Çalışmada öğrencilerin problem çözme becerileri ve bilgi işlemsel düşünme becerilerine olan etkisi araştırılmıştır. Gelecekte yapılacak çalışmalarda farklı değişkenlere olan etkisi araştırılabilir.
4. Kullanıcılar Scratch programlama ile her türlü görsel uygulamalar geliştirebildiğinden bilişim dersi dışındaki diğer derslere de disiplinler arası çalışma yapılıp bu çalışma sonuçları incelenebilir.
5. Bu yöntemler ile gelecekte çalışma yapacak araştırmacılar farklı programlama araçlarını kullanarak öğrencilerin farklı değişkenlere göre tutumlarını inceleyip bu alana yönelik farklı bakış açıları getirebilirler.
6. Uzun sürecek araştırmalarda araştırmacının günlük tutması önerilir. Sınıf içerisinde yaşanan olayları bir günlüğe yazarak uzun zaman sonra tekrar bu verilere ulaşılıp, analizler esnasında bu verileri kullanabilmelerine olanak sağlayacaktır.

Öğretmenlere Yönelik Öneriler

1. Ters yüz sınıf modelini kullanacak eğitimcilerin ilk olarak bu sistemin evde ders okulda ödev felsefesi ile geliştirildiğini unutmamalıdır. Evde ders bölümünde öğrenciler internet üzerinden ders yapacaklarından dolayı sistem içerisinde kullanılacak ders materyallerinin öğrencilerin kendi bilgisayarlarında açılacak bir formatta olması, resim ve video formatlarının uygun olması, özellikle videoların rahat izlenebilmesi için çok büyük boyutta olmamaları, öğrencilerin

evlerinde kullanabilecekleri ortalama internet hızında açılabilir olmaları önemlidir. Aksi takdirde öğrencilerin evde bu ders içeriklerine ulaşmakta problem yaşamaları bu yöntemin uygulanmasında sıkıntılar ortaya çıkaracaktır.

2. Ters yüz sınıf uygulayacak öğretmenin ders içi etkinlikleri ve ders dışı konu anlatımını çok iyi planlaması gerekmektedir. Etkinliği yapmadan önce mutlaka öğrencilerin konuya olan hazırbulunuşluk düzeylerini ölçmesi gerekir. Aksi takdirde konuyu anlamadan yapılacak etkinlikler ilgili kazanım öğrenciler tarafından benimsenmeyecek ve bilgi eksikliği olacaktır. Bu nedenle ters yüz sınıf modeli ile eğitim gören gruptaki öğrencilerin neden konuyu anlamadıklarını, sistem içerisinde ne kadar süre harcadıkları, bu süre içerisinde neler yaptığı, hangi videoda veya içerikte ne kadar zaman geçirdiklerini tespit edip ona göre bir yol haritası çıkarması önemlidir.
3. Ters yüz öğrenmenin sınıf içi bölümünde daha çok işbirlikçi ve proje çalışmaları ayrıca uygulamaya yönelik çalışmalar yapılabilir.
4. Çalışmada kullanılan yöntemlerin bilgi işlemsel düşünme becerilerine etkisi arasında anlamlı bir fark oluşmamasına rağmen, ortalamalar arasında matematiksel farklılar ters yüz sınıf modeli lehine ve öğrencilerin ters yüz sınıf modelini daha eğlenceli bulduğu konusunda sonuçlar ortaya çıkmıştır. Çünkü bu model öğrenciye kendi kendine ve kendi hızında öğrenmeye olanak sağlamaktadır. Ters yüz sınıf modelini diğer derslerde uygulanması öğrencilerin motivasyon anlamında ve öğrenmeyi daha eğlenceli bir hale getirebileceği düşünülmektedir.

OYS Geliştiren Yazılımcılara Yönelik Öneriler

Ters yüz sınıf modeli ile kullanılacak OYS sistemi tasarlanırken;

Sistem ters yüz sınıf modeli kuramsal çerçevesine oturtulmalıdır. Burada bahsedilen kuramsal çerçeve ters yüz sınıf modelinin felsefesine dayanarak öğrencilerin evde ders okulda ödev veya etkinlik yapılabilecek bir yapıda olmalıdır.

1. Günümüzde interaktif sistemleri daha etkin kullanmak ve bu sistemleri daha eğlenceli, bireylerin keyif alabileceği bir yapıya getirmek için kullanılan oyunlaştırma yöntemi kullanılabilir. Burada öğrenciler konu bitiminde görevleri bitirdikçe rozet ve tüm görevler bittiğinde ise sertifika ile ödüllendirilebilir. Bu yapı öğrencilerin sistemde daha verimli ve daha etkin kullanıp sistemde daha fazla zaman geçirerek görevlerinin bitirme konusunda motivasyon sağlayacağı düşünülmektedir.
2. Teknik olarak herhangi bir eklenti gerektirmeyen bir yazılım geliştirilmesi öğrencilerin evde yaşayacağı teknik sıkıntıları en aza indireceği, bu şekilde her öğrenci evde başka birisinin yardımı olmadan sisteme girip konuları izleyip yapılması gereken görevleri daha kolay ve hızlı yapabileceği düşünülmektedir.
3. Öğretmenlerin sistemde hazır bulunan derslere ek olarak ilgili konuya ait kaynak ve içerik yükleme imkânı verilmelidir. Ayrıca tartışma bölümleri açıp burada

öğrencilerin ilgili konularda görüşlerini öğrenip onlara sistem üzerinden geri bildirimlerde bulunabilirler.

4. Öğretmenlerin konu sonlarında sınavlar hazırlayıp öğrenciye sunabileceği bir yapı olması sistemin daha aktif olarak kullanılmasını sağlayacaktır. Burada öğretmen birçok soru tipinde sorular hazırlayıp (çoktan seçmeli, boşluk doldurma, eşleştirme, vb.) soruları puanlandırabilmelidir.
5. Öğretmenin belirlediği puanın üzerinde alan öğrencilere oyunlaştırmada kullanılan rozet vb. ödüller ile zenginleştirilmesi, öğrencilere belirli bir zaman dilimi içerisinde bu sınav bitirmelerini bildirerek hem öğretmenin başka bir platform üzerinden bu işi yapmasını ortadan kaldıracak hem de bu eğitime ait tüm olayları sistem üzerinden yapabilecektir.
6. Sistemde konular ve sınavlar arşivlenip ileriki yıllarda aynı soruları ve konuları tekrar kullanma veya güncellenip kullanılmasını sağlayacak bir yapının olması öğretmenin sistemdeki yoğunluğunu azaltacağı düşünülmektedir.

KAYNAKLAR

- Aho, A. (2012) Computation and Computational Thinking. *The Computer Journal*, 55(7), 832- 835.
- Akpınar, Y. ve Altun, A. (2014). Bilgi toplumu okullarında programlama eğitimi gereksinimi. *İlköğretim Online*, 13(1).
- Albert, M. & Beatty, B. J. (2014). Flipping the classroom applications to curriculum redesign for an introduction to management course: Impact on grades. *Journal of Education for Business*, 89(8), 419-424.
- Arabacıoğlu, T., Bülbül, T. & Filiz. A. (2007). *Bilgisayar Programlama Öğretiminde Yeni Bir Yaklaşım*. 16.01.2016 tarihinde https://www.academia.edu/1441897/Bilgisayar_Programlama_Oğretiminde_Yeni_Bir_Yaklaşım adresinden erişilmiştir.
- Aşkar, P., & Davenport, D. (2009). An investigation of factors related to selfeff icacy for java programming among engineering students. *The Turkish Online Journal of Educational Technology*, 8(1).
- Baytak, A. & Land, S. (2011) An investigation of the artifacts and process of constructing computers games about environmental science in a fifth grade classroom *Educational Technology Research and Development*, 59(7), 765–782.
- BBC. (2018). Introduction to Computational Thinking. 12.06.2018 tarihinde <https://www.bbc.com/education/guides/zp92mp3/revision/1> adresinden erişilmiştir.
- BCS. (2014). Call for evidence - UK Digital Skills Taskforce. 26.05.2018 tarihinde <http://policy.bcs.org/sites/policy.bcs.org/files/BCS%20response%20to%20UKDST%20call%20for%20evidence%20final.pdf> adresinden erişilmiştir.
- Begosso, L. C. & Da Silva, P. R. (2013). Teaching computer programming: A practical review. *IEEE Frontiers in Education Conference (FIE)*, 508-510. 12.11.2017 tarihinde <http://doi.ieeecomputersociety.org/10.1109/FIE.2013.6684875> adresinden erişilmiştir.
- Benton, L., Hoyles, C., & Noss, I. K. anRichard. (2017). Bridging Primary Programming and Mathematics: some findings of design research in England. *Digital Experiences in Mathematics Education*.
- Berksoy, İ., Sözcü, Ö., Armağan, E., & Arslan, A. (2016) Algoritma ve Programlama Eğitiminde Scratch Programı Üzerine Yapılan Çalışmaların Analizi. 18.01.2017 tarihinde <http://ab.org.tr/ab16/bildiri/74.docx> adresinden erişilmiştir.
- Burke, Q. (2012). The markings of a new pencil: Introducing programming-as-writing in the middle school classroom. *The Journal of Media Literacy Education*, 4(2) :121-135.
- Calder, N. (2010). Using Scratch: An Integrated Problem-solving Approach to Mathematical Thinking. 14.11.2017 tarihinde <http://files.eric.ed.gov/fulltext/EJ906680.pdf> adresinden erişilmiştir.

- Can, A. (2017). *SPSS ile Bilimsel Araştırma Sürecinde Nicel Veri Analizi* (5. Baskı). Ankara: Pegem Akademi Yayıncılık.
- Chiu, C. F. (2014). Teaching programming concepts to K-12 teachers with Scratch. *Journalism and Mass Communication*, 4(2), 125-132.
- Creswell, J. W. (2012). *Planning, Conducting, and Evaluating Quantitative and Qualitative Research* (4rd ed). Boston,: Pearson.
- Çakır, E. (2017). Ters Yüz Sınıf Uygulamalarının Fen Bilimleri 7. Sınıf Öğrencilerinin Akademik Başarı, Zihinsel Risk Alma ve Bilgisayarca Düşünme Becerileri Üzerine Etkisi. *Yüksek Lisans Tezi, Ondokuz Mayıs Üniversitesi, Samsun*.
- Çatlak, Ş., Tekdal, M. & Baz, F. Ç., (2015). Scratch Yazılımı ile Programlama Öğretiminin Durumu: Bir Doküman İnceleme Çalışması. *Journal of Instructional Technologies & Teacher Education*, 4(3), 13-25.
- Demiralay, R. & Karataş, S. (2014). Evde Ders Okulda Ödev Modeli. *Eğitim ve Öğretim Araştırmaları Dergisi*. 3(3), 333-340.
- Dill, E. M. (2012). The impact of flip teaching on student homework completion, behavior, engagement, and proficiency. Completed to meet the requirements of University of New England MS Ed. Program. 10.11.2017 tarihinde <http://www.lessonpaths.com/learn/i/flipped-classroom-research-paper/17-flipped-research-paper-action-research-effects-of-flipped-classroom> adresinden erişilmiştir.
- Dreher, C., Reiners, T., and Dreher, H., (2011) Investigating factors affecting the uptake of automated assessment technology. *Journal of Information Technology Education*, 10, 161-181.
- Durak, H., Karaoğlan Yılmaz, F. G., Yılmaz, R. & Seferoğlu, S. S. (2017). Erken Yaşta Programlama Eğitimi: Araştırmalardaki Güncel Eğilimlerle İlgili Bir İnceleme. *Eğitim Teknolojileri Okumaları 2017*, 12, 205-236.
- Ekmekçi, E. (2014). Harmanlanmış öğrenme odaklı tersten yapılandırılmış yazma sınıfı modeli. *Doktora Tezi. Gazi Üniversitesi, Ankara*.
- Erol, O. (2015). Scratch ile Programlama Öğretiminin Bilişim Teknolojileri Öğretmen Adaylarının Motivasyon Ve Başarılarına Etkisi. *Doktora Tezi, Anadolu Üniversitesi Eğitim Bilimleri Enstitüsü, Eskişehir*.
- Erümit, A.K. & Benzer, A.İ & Aksoy, D. A. & Aksoy, A. & Şahin, G. (2017) Algoritmik Düşünme için Programlama Öğretimi Adımları. *Eğitim Teknolojileri Okumaları, 2017(1-15)* , Ankara: TOJET
- Fadjo, C. L. (2012). Developing computational thinking through grounded embodied cognition. *Doktora Tezi, Columbia University, ABD*.
- Fessakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education*, 63, 87-97.
- Feurzeig, W. & Papert, S. (1969). Programming-languages as a conceptual framework for teaching mathematics. Final report on the first fifteen months of the Logo Project,

- Cambridge, MA: BBN. 03.03.2018 tarihinde <https://files.eric.ed.gov/fulltext/ED038034.pdf> adresinden erişilmiştir.
- Fulton, K. (2012). Upside down and inside out: Flip your classroom to improve student learning. *Learning & Leading with Technology*, 39(8), 12–17.
- Futschek, G. (2006). Algorithmic thinking: the key for understanding computer science. In *International Conference on Informatics in Secondary Schools-Evolution and Perspectives* (pp.159-168).
- Garneli, V., Giannakos, M. N., & Chorianopoulos, K. (2015). Computing education in K-12 schools: A review of the literature. *Global Engineering Education Conference (EDUCON), 2015 IEEE* (pp. 543–551). IEEE.
- Genç, Z. & Karakuş, S. (2011). Tasarımla öğrenme: Eğitsel bilgisayar oyunları tasarımında Scratch kullanımı. *International Computer and Instructional Technologies Symposium ICITS*, 981-987, Elazığ, Türkiye.
- Gençer, B.G ., Gürbulak, N. & Adıgüzel, T. (2014). Eğitimde yeni bir süreç: Ters yüz sınıf sistemi. *International Teacher Education Conference (ITEC)*. 12.12.2017 tarihinde <http://www.egitimdeteknoloji.com/egitimde-yeni-bir-surec-ters-yuz-sinif-sistemi> adresinden erişilmiştir.
- Green, S., & Salkind, N. (2005). *Using SPSS for Windows and Macintosh (4th ed.)*. Upper Saddle River, NJ: Pearson Prentice Hall.
- Grover, Pea, & Cooper. (2015). Designing for deeper learning in a blended computer science course for middle school students. *Computer Science Education*, 25(2), 199–237.
- Gülbahar, Y. & Kalelioğlu, F. (2014). Bilge Kunduz Uluslararası Enformatik Yarışması Türkiye Raporu: 2014 yılı pilot uygulaması. 05.05.2018 tarihinde <http://www.bilgekunduz.org/wp-content/uploads/2015/10/bilgekunduz-rapor-2014.pdf> adresinden erişilmiştir.
- Gülbahar, Y. & Kalelioğlu, F. (2014). Türkiye 2014 Bilge Kunduz Etkinliği Soruları. 09.11.2017 tarihinde <http://www.bilgekunduz.org/gecmis-gorevler/> adresinden erişilmiştir.
- Gülbahar, Y. & Kalelioğlu, F. (2015). Türkiye 2015 Bilge Kunduz Etkinliği Soruları. 09.11.2017 tarihinde <http://www.bilgekunduz.org/2015-gecmis-gorevler/> adresinden erişilmiştir.
- Gülbahar, Y., Kalelioğlu, F. & Doğan, D. (2015). Bilge Kunduz Uluslararası Enformatik Yarışması Türkiye Raporu: 2015 yılı pilot uygulaması. 05.05.2018 tarihinde <http://www.bilgekunduz.org/wp-content/uploads/2016/01/bilgekunduz-rapor-2015.pdf> adresinden erişilmiştir.
- Gülbahar, Y., Kert, S. B. & Kalelioğlu, F. (baskıda). Bilgi İşlemsel Düşünmeye Yönelik Öz Yeterlik Algısı Ölçeği (BİDÖA) : Geçerlik ve Güvenirlik Çalışması.
- Gülmez, I. (2009). Programlama Öğretiminde Görselleştirme Araçlarının Kullanımının Öğrenci Başarı ve Motivasyonuna Etkisi. *Yüksek Lisans Tezi, Marmara Üniversitesi, İstanbul*.

- Gür, H. & Hangül, T. (2015). ortaokul öğrencilerinin problem çözme stratejileri üzerine bir çalışma. *Pegem Eğitim ve Öğretim Dergisi*, 5(1), 95-112.
- Hmelo-Silver, C. E. & Barrows, H. S. (2006). Goals and strategies of a problem-based learning facilitator. *Interdisciplinary Journal of Problem-based Learning*, 1, 21-39.
- Howland, K. & Good, J. (2015). Learning to communicate computationally with Flip: A bi-modal programming language for game creation. *Computers & Education*, 80, 224-240.
- Hsu, H. & M., J. (2014). Gender differences in Scratch game design. 2014 International Conference on Information, Business and Education Technology (ICIBET), Taiwan, 11.11.2017 tarihinde http://www.atlantispress.com/php/download_paper.php?id=11390 adresinden erişilmiştir.
- Hu, C. (2011). Computational thinking: what it might mean and what we might do about it. *Proceedings of the 16th Annual Joint Conference on Innovation and Technology in Computer Science Education - ITiCSE '11*, 223-227.
- Kafai, Y. B., & Burke, Q. (2015). Constructionist gaming: Understanding the benefits of making games for learning. *Educational Psychologist*, 50(4), 313-334.
- Kafai, Y. B., Lee, E., Searle, K., Fields, D., Kaplan, E., & Lui, D. (2014b). A crafts-oriented approach to computing in high school: Introducing computational concepts, practices, and perspectives with electronic textiles. *ACM Transactions on Computing Education (TOCE)*, 14(1), 1-20. doi: 10.1145/2576874
- Kalelioğlu, F. & Gülbahar, Y. (2014). The Effects of Teaching Programming via Scratch on Problem Solving Skills: A Discussion from Learners' Perspective. *Informatics in Education*, 13(1), 33-50.
- Kalelioğlu, F. & Gülbahar, Y. (2016). Eğlenceli Bilgisayar Bilimi Etkinliği: Bilge Kunduz. *Eğitimde FATİH Projesi Eğitim Teknolojileri Zirvesi*. Ankara, Türkiye.
- Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code.org. *Computers in Human Behavior*. 52, 200-210.
- Kalelioğlu, F., Gülbahar, Y., & Kukul, V. (2016). A Framework for Computational Thinking Based on a Systematic Research Review. *Baltic Journal of Modern Computing*, 4(3), 583-596.
- Kalelioğlu, F., Gülbahar, Y., Akçay, S. & Doğan, D. (2014). Curriculum Integration Ideas for Improving the Computational Thinking Skills of Learners through Programming via Scratch. *The 7th International Conference on Informatics in Schools: Situation, Evolution and Perspectives*. Istanbul University, Turkey, September 22-25.
- Karaca, C. (2016). Öğretim teknolojilerinde güncel bir yaklaşım: Ters yüz öğrenme. *Pegem Atıf İndeksi*, 0, 1161-1172. doi:10.14527/9786053183563.071
- Kasalak, İ. (2017). Robotik Kodlama Etkinliklerinin Ortaokul Öğrencilerinin Kodlamaya İlişkin Öz-Yeterlik Algılarına Etkisi Ve Etkinliklere İlişkin Öğrenci Yaşantıları. *Yüksek Lisans Tezi, Ankara Üniversitesi, Ankara*.

- Kert, S. B., & Uğraş, T. (2009). Programlama eğitiminde sadelik ve Eğlence: Scratch Örneği. *The First International Congress of Educational Research, Çanakkale*. 11.11.2017 tarihinde https://www.academia.edu/3623864/Programlama_Eğitiminde_Sadelik_ve_Eğlence_Scratch_Örneği adresinden alınmıştır.
- Kocabatmaz, H. (2016). Ters Yüz Sınıf Modeline İlişkin Öğretmen Adayı Görüşleri. *Eğitim ve Öğretim Araştırmaları Dergisi Journal of Research in Education and Teaching*, 5(4).
- Koenig, J. (2011). Assessing 21st Century Skills: Summary of a Workshop. *Washington, DC: National Research Council*.
- Kukul, V., & Gökçearslan, Ş. (2014). Scratch ile Programlama Eğitimi Alan Öğrencilerin Problem Çözme Becerilerinin İncelenmesi. *8th International Computer & Instructional Technologies Symposium, Trakya University Edirne*.
- Lamb, A., & Johnson, L. (2011). Scratch: computer programming for 21st century learners. *Teacher Librarian*, 38(4), 64–68.
- Lau, W. W., & Yuen, A. H. (2011). Modelling programming performance: Beyond the influence of learner characteristics. *Computers & Education*, 57(1), 1202-1213.
- Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, 41, 51–61.
- Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L. & Settle, A. (2014). Computational thinking in K-9 education. In: Proceedings of the Working Group Reports of the 2014 on Innovation & Technology in *Computer Science Education Conference, ITiCSE-WGR 2014*, 1–29. ACM, New York.
- Mazman, S. & Altun, A. (2013) Programlama-1 Dersinin BÖTE Bölümü Öğrencilerinin Programlamaya İlişkin Öz Yeterlilik Algıları Üzerine Etkisi. *Journal of Instructional Technologies & Teacher Education*, 2(3), 24-29
- Miller, A. (2012). Five Best Practices for the Flipped Classroom. 13.05.2017 tarihinde <https://www.edutopia.org/blog/flipped-classroom-best-practices-andrew-miller> adresinden erişilmiştir.
- Moreno, J, Robles, G. & Román, M. (2015) Dr. Scratch: Automatic Analysis of Scratch Projects to Assess and Foster Computational Thinking. *RED-Revista de Educación a Distancia.Número 46*.
- Nederveld, A. (2014) Flipped Primer – Part 1. 11.11.2017 tarihinde <http://https://abnederveld.com/2014/05/02/flipped-primer-1> adresinden erişilmiştir.
- Nikou, S., & Economides, A. (2014). *Transition in student motivation during a Scratch and an Appinventor Course*. Proceedings of Global Engineering Education Conference (EDUCON). İstanbul.
- Numanoğlu, M. & Keser, H. (2017). Programlama Öğretiminde Robot Kullanımı - Mbot Örneği. *Bartın Üniversitesi Eğitim Fakültesi Dergisi*, 6 (2), 497-515. DOI: 10.14686/buefad.306198

- Oluk, A , Korkmaz, Ö , Oluk, H . (2018). Scratch'ın 5. Sınıf Öğrencilerinin Algoritma Geliştirme ve Bilgi-İşlemsel Düşünme Becerilerine Etkisi. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 9 (1), 54-71. DOI: 10.16949/turkbilm.399588.
- Özcan, S., Ergün, K., Köse, Ö., Emir, N. & Gezgin, D. (2017, Nisan). Bilgisayar Programlama Eğitiminde Scratch Programı Kullanımına İlişkin Lise Öğrencilerinin Görüşleri. *2nd International Scientific Researches Congress on Humanities and Social Sciences (IBAD-2017), İstanbul*.
- Özden, Y. (2005). *Öğrenme ve Öğretme*. (7. Baskı), Ankara: Pegem Yayıncılık.
- Özer, B. (2002). İlköğretim ve ortaöğretim okullarının eğitim programlarında öğrenme stratejileri. *Eğitim Bilimleri ve Uygulama*. 1(1),17-32.
- Pulimood, S. M., Pearson, K., & Bates, D. C. (2016). *A study on the impact of multidisciplinary collaboration on computational thinking*. Proceedings of the 47th ACM Technical Symposium on Computing Science Education (pp. 30-35). New York, NY: ACM.
- Sant., J.A., (2009) Mailing it in: email-centric automated assessment. *ITiCSE '09: Proceedings of the 14th annual ACM SIGCSE Conference. on Innovation and technology in computer science education*, 308–312, New York, NY, USA. ACM.
- Seferoğlu, S. S. (2015). Okullarda teknoloji kullanımı ve uygulamalar: gözlemler, sorunlar ve çözüm önerileri. *Artı Eğitim*, 123, 90-91.
- Serin, O., Bulut Serin, N. & Saygılı, G. (2010). İlköğretim düzeyindeki çocuklar için problem çözme envanteri'nin (ÇPÇE) geliştirilmesi. *İlköğretim Online*, 9 (2), 446–458.
- Shin, S., & Park, P. (2014). A Study on the Effect affecting Problem Solving Ability of Primary Students through the Scratch Programming. 11.12.2017 tarihinde http://onlinepresent.org/proceedings/vol59_2014/27.pdf adresinden erişilmiştir.
- SoonsikSeo (2008). Effects of Digital Textbook Usage on the Improvement of Problem Solving Competency. *Korea Education and Research Information Service*
- Şanal, S.Ö. & Erdem, M. (2017). Kodlama ve Robotik Çalışmalarını Problem Çözme Süreçlerine Etkisi: Sesli Düşünme Protokol Analizi. *Uluslararası Bilgisayar ve Öğretim Teknolojileri Sempozyumu* (24 – 26 Mayıs 2017). Malatya.
- Talbert, R. (2012). Inverted Classroom. *Colleagues*, 9(1), 7
- Teague, D., & Lister, R. (2014). Programming: reading, writing and reversing. *Proceedings of the 2014 conference on Innovation & technology in computer science education* (pp. 285–290). ACM.
- Tekerek, M. & Altan, T. (2014). The effect of Scratch environment on student's achievement in teaching algorithm. *World Journal on Educational Technology*, 6(2), 132-138.

- Tseng, S. S., & Weng, J. F. (2009). Wiki-based design of Scientific Inquiry assessment by game-based Scratch programming. *Advanced Learning Technologies, 2009. ICALT 2009. Ninth IEEE International Conference*, 704-706.
- Tugun, V., Uzunboylu, H., & Ozdamli, F. (2017). Coding Education in a Flipped Classroom. *TEM Journal*, 6(3), 599-606
- UETB & BBÖD. (2011). Operational Definition of Computational Thinking for K–12 Education 11.11.2017 tarihinde <http://www.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf> adresinden erişilmiştir.
- UETB (2016.). The ISTE National Educational Technology Standards (NETS•S) and Performance Indicators for Students. 18.04.2017 tarihinde <http://www.iste.org/standards/standards/for-students-2016> adresinden erişilmiştir.
- Verleger, M. A., & Bishop, L. J. (2013). The flipped classrom: A survey of the research. 120th ASEE Conference & Exposition. *American Society for Engineering Education*.
- Vihavainen, A., Vikberg, T., Luukkainen, M., and Pärtel, M., (2013) Scaffolding students' learning using test my code. In *Proceedings of the 18th ACM conference on Innovation and technology in computer science education*, 117-122.
- Walsh K. (2015), Results Summary of 2014 Pilot of Flipped Classroom Techniques . 11.11.2017 tarihinde http://emergingedtech.com/ebook/2014_Pilot_Flipped_Courses_College_of_Westchester.pdf adresinden erişilmiştir.
- Webb, D. C., Repenning, A., & Koh, K. H. (2012). Toward an emergent theory of broadening participation in computer science education. *Proceedings of the 43rd ACM technical symposium on Computer Science Education*, 173–178.
- Wilson, A., Hainey, T., & Connolly, T. (2012). Evaluation of computer games developed by primary school children to gauge understanding of programming concepts. *Proceedings of the 6th European Conference on Games-Based Learning (ECGBL)*, 4-5.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society*, 366, 3717-3725. doi:10.1098/rsta.2008.0118
- Wing, J. M. (2011). Research notebook: Computational thinking—What and why? The Link Magazine, Spring. Carnegie Mellon University, Pittsburgh. 18.04.2017 tarihinde <https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why> adresinden erişilmiştir.
- Yeni, S. (2017). Bilgi İşlemsel Düşünme Becerisi Nasıl Değerlendirilir? *Pegem Atıf İndeksi*, 0, 359-394. doi:10.14527/3123
- Yükseltürk, E. & Altıok, S. (2016). Investigation of pre-service information technology teachers' game projects prepared with Scratch. *SDU International Journal of Educational Studies*, 3(1), 59-66.

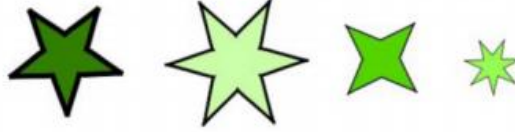
- Yüzer, V., & Kılınç, H. (2015). Açık öğrenme sistemlerinde dijital öykülemeyi faydalanmak. *Eğitim ve Öğretim Araştırmaları Dergisi*, 4(1), 243-250.
- Zappe, S., Leicht, R., Messner, J., Litzinger, T., & Lee, H. (2009). "Flipping" the classroom to explore active learning in a large undergraduate course. *Proceedings of the 2009 American Society for Engineering Education Annual Conference and Exhibition*.
- Zhang, H., Du, X., Yuan X., & Zhang L. (2016) The Effectiveness of the Flipped Classroom Mode on the English Pronunciation Course. *Creative Education* 7, 1340-1346.
- Zownorega, J. S. (2013). Effectiveness of flipping the classroom in a honors level, mechanics-based physics class. *Master's Thesis. Eastern Illinois University, Illinois*.

EK -1: BİLGE KUNDUZ ULUSLARARASI ENFORMATİK VE BİLGİ İŞLEMSEL DÜŞÜNME SORULARI

1.

Çılgın Yıldızlar

Ayşe'nin dört adet plastik yıldız vardır. Her yıldızın büyüklüğü, rengi, kenar kalınlığı ve köşe sayısı farklıdır. Ayşe, yıldızlarını bu özelliklerine göre sıralamayı seviyor. Örneğin, aşağıdaki resimde yıldızlar kenar kalınlıklarına göre kalından inceye doğru sıralanmıştır.




Soru:

Aşağıdaki seçeneklerden hangisinde yıldızlar belirli bir özelliğe göre sıralanmamıştır? Örneğin, yıldızlar koyu renkten açık renge veya açık renkten koyu renge göre sıralanmamalıdır.

2.

Ali, yandaki şekildeki gibi tuşları olan bir mobil telefon ile arkadaşına, bir ismi mesaj olarak göndermek istemektedir. Aynı tuşa birkaç kez basıldığında harfler ekranda görünmektedir.

Örneğin, C harfi için  tuşuna 3 kere basmak gerekmektedir.

CAN kelimesini yazmak için ise  tuşuna 3 kere, sonra 

tuşuna bir kere ve son olarak  tuşuna iki kere basılmalıdır.



Soru :

Ali, 7 kez tuşa basarak arkadaşının ismini mesaj olarak göndermiştir. Aşağıdaki seçeneklerden hangisi gönderdiği isimdir?

3.

Bir dondurmacıda, dondurma topları müşterilerin istediği sırayla külahı koyulmaktadır.

Soru:

Yandaki şekilde bulunan dondurma külahını alan müşteri nasıl sipariş vermiştir?

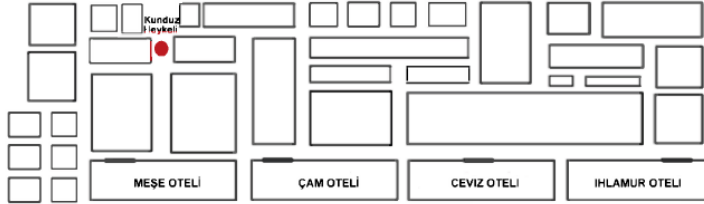


4.

Bilge Kunduz, kunduz kasabasında bir otelde kalmaktadır. Otel çalışanları tarafından kendisine verilen yol tarifine göre, kasabanın en ünlü kunduz heykelinin fotoğrafını çekmek üzere yola çıkmıştır:

1. Otel kapısından hemen sola dön.
2. İki yol ayrımı boyunca düz git.
3. Üçüncü yol ayrımında sağa dön.
4. Düz devam et. İlk yol ayrımında sola dön.
5. Düz devam et. İlk yol ayrımında sağa dön.

Aşağıdaki şekilde dört otelin haritası görülmektedir.



Soru :

Bilge Kunduz hangi otelde kalmaktadır?

5.

Her 7 dakikada yeni bir siyah araba üretim hattından geliyor. Her 5 dakikada diğer bir hattan yeni bir beyaz araba geliyor. Bir sürücü araba taşıyıcısına bırakılacak şekilde sırayla üretim hattına arabaları park ediyor. İlk olarak araba taşıyıcının üst katına yükleme oluyor. İki üretim hattında aynı anda çalışmaya başlıyor.

Soru

Yükleme sonrasında araba taşıyıcısı nasıl gözükür?

6.

Bilge Kunduzun, aynı şekilde olan 16 adet fayansı bulunuyor. Bunları döndürerek farklı desenler ortaya çıkarabiliyor.



Bir banyonun duvarını bu 16 fayans ile kaplamak istiyor. Bunun için bir tasarım düşünüyor.

Soru:

Aşağıdaki desenlerden hangisi bu fayanslar ile yapılamaz?

7.

Bilge Kunduzun, yedi bölmeli göstergesi olan dijital bir saati bulunmaktadır. Sayılar farklı çizgilerin yanması sonucu oluşmaktadır.



Bilge Kunduz saatini bozmuş ve saatteki birkaç çizginin yanmadığını fark etmiştir.

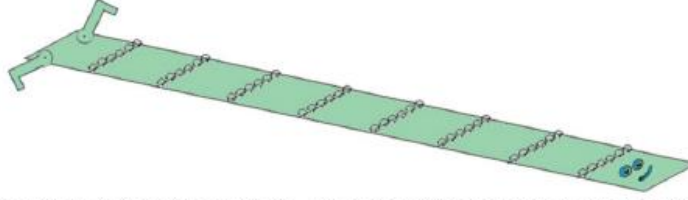


Soru

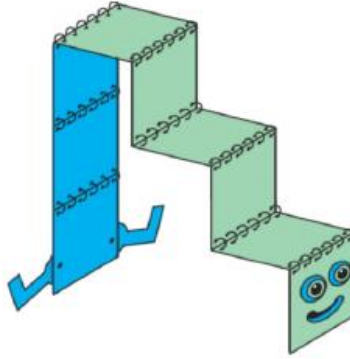
Aşağıdaki seçeneklerden hangisi yukarıdaki saatin doğru gösterilmiş halidir?

8.

Bilge Kunduz, bir robot yapmaktadır. Robot, her biri eşit, kare levhadan oluşmaktadır. Bilge Kunduz yatay bir zemin üzerinde tüm levhaları menteşe ile birbirine bağlayarak robotu tamamlamıştır:



Bilge Kunduz, robotu menteşelerinden katlayarak şekillendirebiliyor. Örneğin, kunduz, 9 levhadan oluşan robotu, yerden yüksekliği 3 levha olan 3 basamaklı bir merdivene dönüştürebiliyor.



Soru:

Merdivenin yerden yüksekliğinin 7 levha olması için robotun toplam kaç levhadan oluşması gerekmektedir?

9.

Bilge Kunduz hayalindeki elbiseyi almak istemektedir. Almak istediği elbisenin özellikleri aşağıdaki gibidir:

- Kısa kollu olmalıdır.
- 3'ten fazla düğmesi olmalıdır.
- Kollarında yıldız olmalıdır.

Dört markaya ait satılan elbiseler vitrinde gösterilmektedir.

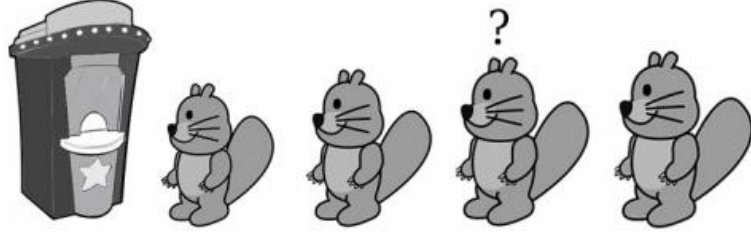


Soru

Bilge Kunduzun almak istediği elbiseyi hangi marka satmaktadır?

10.

Dört kunduz: Aylin, Efe, Nazan ve Hakan bir filme bilet almak için sıraya girdiler.



Bilindiği üzere:

- 1) Hakan sıranın başında değil.
- 2) Nazan Efe'nin hemen önünde.
- 3) Aylin sırada Hakan'ın arkasında.

Soru

Sırada üçüncü olan kimdir?

11.

Ağaç Kesimi

İki bilge kunduzun kesmek istediği üç ağaç vardır. Bir ağacı bir kunduz 10 dakikada kesiyor. Kunduzlar, her bir ağaç için istedikleri kadar zaman harcıyor. Ancak, iki kunduz tek bir ağaçta aynı anda çalışmıyor; çünkü dişleri ile birbirlerine zarar veriyor.



Soru:

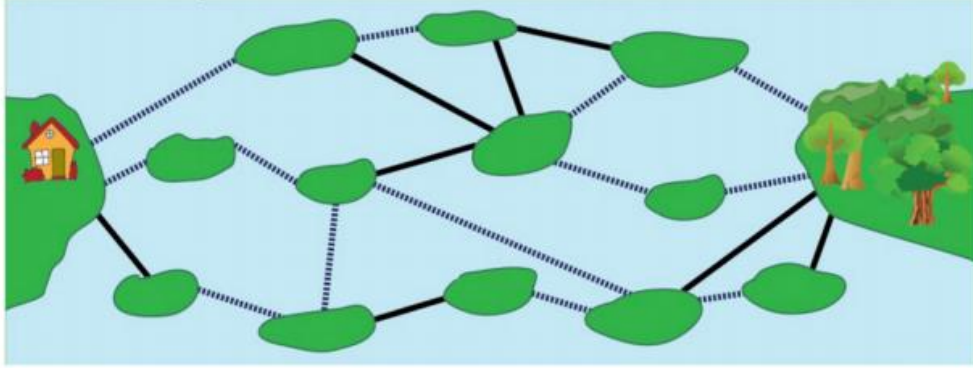
İki kunduzun üç ağacı kesebileceği en kısa süre ne kadardır?

- A) 30 dakika
- B) 20 dakika
- C) 15 dakika
- D) 10 dakika

12.

Köprüler

Bir şehir parkında, üzerinde birçok ada olan büyük bir göl vardır. Aşağıdaki şekilde de gösterildiği gibi adalar birbirlerine iki farklı türde köprüler ile bağlıdır. Düz çizgilerle gösterilen köprülerden geçiş halka açık ve ücretsiz, kesik çizgiler ile gösterilen köprüler ise ücretlidir. Bilge kunduz evinden ormana gitmek istemektedir. En fazla iki ücretli köprüye yetecek kadar parası bulunmaktadır.



Soru:

Bilge kunduzun geçebileceği en az sayıdaki köprü sayısı kaçtır?

13.

Alışveriş

Çınar, arkadaşı Bulut'u ziyaret eder. Arkadaşına hediye olarak bir parça kunduz peyniri almıştır. Bulut hediye için ücretini öğrenmesini diye fişin bir kısmını koparmıştır. Ama yine de Bulut ücreti öğrenebilmektedir.

Soru:

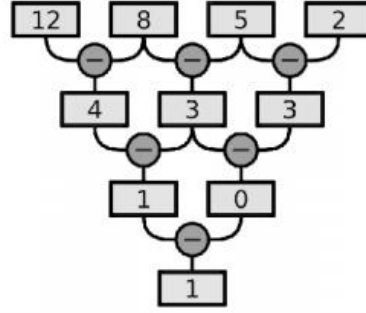
Fişin şekilde gösterilen hangi kısımları ile peynire yapılan ödeme miktarı hesaplanabilir?



14.

Piramit Satırları

İşlem makinesi ilk satırdaki 4 rakamı girdi olarak almaktadır. Her satırda, makine sayılar arasındaki farkı hesaplamaktadır. Aşağıdaki resimde örnek bir işlem görülmektedir.



Soru

Aşağıdaki girdi rakamlarından hangisi son satırda sonucun "0 (sıfır)" olmasını sağlar?

15.

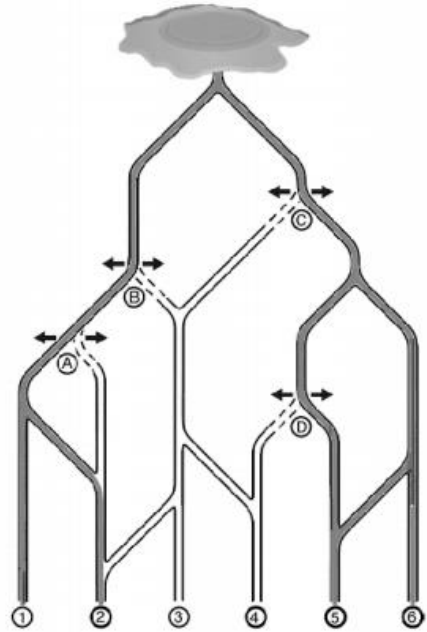
Su Kanalı

Bilge Kunduzlar, kendi bölgeleri için bir sulama sistemi oluşturmuşlardır. Bu sistemde, tepedeki bir gölden 1 ile 6 arasındaki bölgeye aşağıya doğru su akmaktadır.

Su kanallarında sol ok (←) ve sağ ok (→) olan yerlerde A'dan D'ye isimlendirilen dört adet kapı bulunmaktadır. Bu kapıların olduğu bölgeden sular hem sola hem de sağa doğru akabilmektedir.

Soru

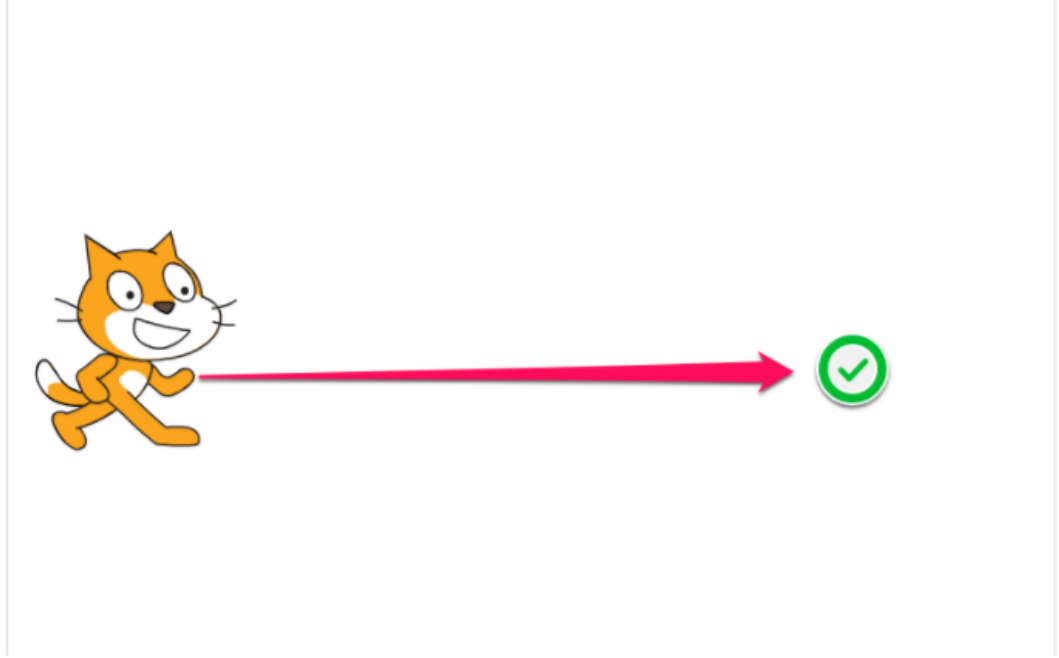
Suyun sadece 2, 4, 5 ve 6 no'lu bölgelere akabilmesi için doğru kapı seçenekleri hangisidir?



EK -2: HAFTALIK DEĞERLENDİRME SINAVLARI ÖRNEK SORULARI

Örnek Soru:

Aşağıdaki karakterin ok ile gösterilen yönde yeşil noktaya varması için aşağıdaki kodlardan hangisinin kullanılması gerekir?



Haftalık Başarı Testi Örnek Soru

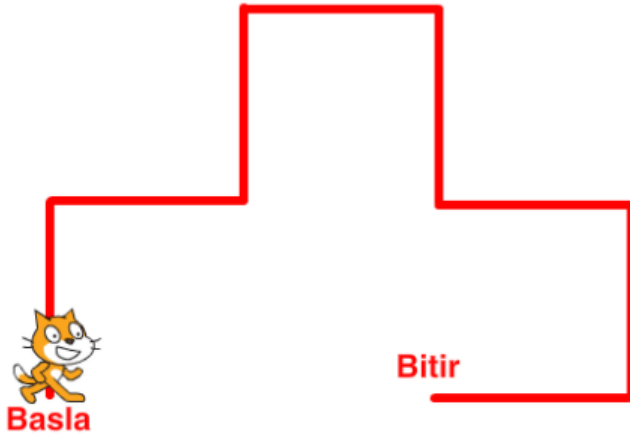
Örnek Soru:

Aşağıdaki kod bloğu seçeneklerde belirtilen görevlerden hangisini yapar?



Örnek Soru:

Aşağıdaki resimde sahnedeki karakter çizgi üzerinde klavye tuşları(yukarı ok, aşağı ok,sağ ok,sol ok) ile hareket ettirmek isteniyor. Fakat seçeneklerdeki klavye tuşları ile hareketlendirme kod bloklarının birisi yanlış yazıldığından karakter çizgi üzerinde gidememektedir. Bu yanlış yazılan kod bloğu hangisidir?



Haftalık Başarı Testi Örnek Soru

Örnek Soru:

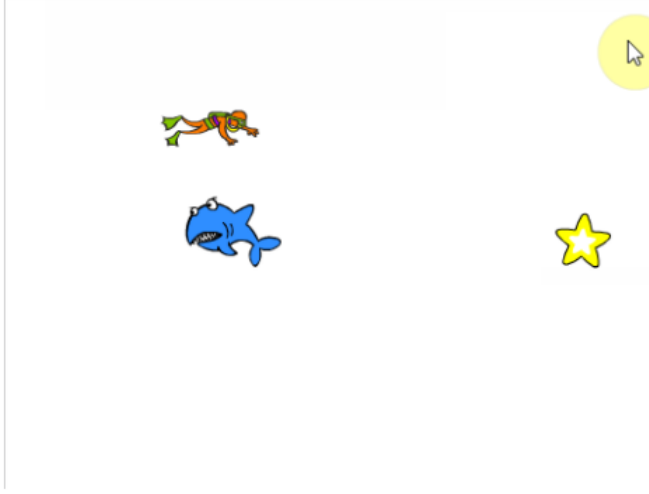


Aşağıdaki soruları yukarıda verilen animasyona göre cevaplayınız.

Mustafa, top kırmızı çizgiye değdiğinde canın bir azalmasını istiyor. Bu işi aşağıdaki kod bloklarından hangisi yapabilir?

Haftalık Başarı Testi Örnek Soru

Örnek Soru:



Aşağıdaki açıklamayı okuyup soruları yukarıda verilen animasyona göre cevaplayınız.

Deniz dalgıçın köpekbalıklarını geçerek yıldızı almasını amaçlayan bir oyun yapmak istiyor. Oyunda dalgıç yıldızı aldığı anda oyun bitecektir. Yıldızı alırken köpekbalıklarından zıplayarak kaçması gerekiyor. Köpekbalıkları farklı zamanlarda sahnede görünmektedir. Sahnede aynı anda birden fazla köpekbalığı görünebilir.

Deniz köpekbalığı dalgıça değdiğinde ekrandan kaybolması için bir "balikyedi" adında bir haber sal kodu oluşturmuştur. Aşağıdaki kod bloklarından hangisi "balikyedi" haberinin kodları olabilir?

Haftalık Başarı Testi Örnek Soru

Örnek Soru:

Sena "Geometrik Şekilleri Tanıyalım" isimli bir oyun yapmıştır. Kullanıcıdan gelen cevaba göre geometrik şekiller çizmektedir. Bu oyundaki şekilleri doğru çizmek için Sena aşağıdaki kodlardan hangisini yazmış olabilir?



Haftalık Başarı Testi Örnek Soru

EK -3: SCRATCH PROGRAMLAMA DİLİNİ FARKLI STRATEJİLER İLE ÖĞRENME KONUSUNDAKİ ODAK GRUP GÖRÜŞMESİ ÖRNEK FORMU

AD SOYAD:

SINIF:

NUMARA:

Bu görüşmede Scratch programlama kendi grubunuzdaki yöntem ile öğrenme konusunda görüşme soruları hazırladım. Odak grup görüşmesi yönteminde, bir sorudan diğerine hepimizin cevabını aldıktan sonra geçeceğiz.

Bu süreçte uzun cevaplar karşısında bekleyip sıkılabileceğiniz durumlar da gerçekleşebilir. Sabrınız için şimdiden teşekkür ederim.

Açıklamalarınız sadece bu araştırma kapsamında kullanılacaktır.

Görüşme için izin verirseniz konuşmalarımızın tamamı eksiksiz biçimde araştırmada yer alması için bir ses kayıt cihazıyla kayıt altına almak istiyorum. Ayrıca görüşme esnasında bir yandan önemli gördüğümüz hususlarla ilgili notla alacağım.

Görüşmeye başlamadan önce, konuyla ilgili belirtmek istediğiniz bir düşünce ya da sormak istediğiniz bir soru var mı?

Bu görüşmenin yaklaşık 30 dakika saat süreceğini tahmin ediyorum ve izin verirseniz sorularımıza başlamak istiyorum.

Geleneksel Sınıf Modeli	Ters Yüz Sınıf Modeli
1. Scratch programlama dilini derste öğretmen anlatımı ile öğrenme konusunda ne düşünüyorsun?	1. Scratch programlama dilini internet üzerinden kendi kendine öğrenme konusunda ne düşünüyorsun?
2. Scratch programlama dilini sınıf ortamında öğrenirken yaşadığın zorluklar nedir?	2. Scratch programlama dilini internet üzerinden kendi kendine öğrenirken yaşadığın zorluklar nedir?
3. Scratch programlama dilini daha iyi öğrenmek için ne yapılmasını önerirsin?	
4. Scratch programlama ortamında program yazmak sana neler kazandırmış olabilir?	
5. Scratch programlama ortamında programlama ile ilgili hangi kavramları öğrendiğini düşünüyorsun?	

ÖZGEÇMİŞ

Kimlik Bilgileri

Adı Soyadı: Ecmen ERDEM

Doğum Tarihi : 02.08.1985

Doğum Yeri: ANKARA

E-posta: ecmenerdem@gmail.com

Eğitim

Lisans: BAŞKENT ÜNİVERSİTESİ BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ
ÖĞRETMENLİĞİ (2003-2008)

İş Deneyimi:

2011- ... Özel Maya Okulları - Bilişim Teknolojileri Öğretmeni

2008 - 2011 TED Eskişehir Koleji - Bilişim Teknolojileri Öğretmeni

Ecmen Erdem Tez

ORJİNALLİK RAPORU

% 3	% 2	% 1	%
BENZERLİK ENDEKSİ	İNTERNET KAYNAKLARI	YAYINLAR	ÖĞRENCİ ÖDEVLERİ

BİRİNCİL KAYNAKLAR

1	www.ittes2016.org İnternet Kaynağı	% 1
2	TOK, Şükran. "MİKRO ÖĞRETİM YÖNTEMİNİN ÖĞRETMEN ADAYLARININ ÖZYETERLİK İNANÇLARINA VE MESLEĞE YÖNELİK TUTUMLARINA ETKİSİ", Milli Eğitim Bakanlığı, 2016. Yayın	% 1
3	ejercongress.org İnternet Kaynağı	% 1

Alıntıları çıkart üzerinde Eşleşmeleri çıkar < %1
Bibliyografyayı Çıkart üzerinde