



**T.C.
AMASYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**TOXICODE VE GOOGLE BLOCKLY UYGULAMALARININ
ÖĞRENCİLERİN BİLGİ İŞLEMSEL DÜŞÜNME BECERİSİNE VE
ALGORİTMA GELİŞTİRMeye İLİŞKİN ÖZ YETERLİK ALGISINA ETKİSİ**

YÜKSEK LİSANS TEZİ

BÜLENT KOÇAK

KASIM

BÜLENT KOÇAK

**FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR VE ÖĞRETİM
TEKNOLOJİLERİ EĞİTİMİ
ANABİLİM DALI**

KASIM 2023

**TOXICODE VE GOOGLE BLOCKLY UYGULAMALARININ
ÖĞRENCİLERİN BİLGİ İŞLEMSEL DÜŞÜNME BECERİSİNE VE
ALGORİTMA GELİŞTİRMEYE İLİŞKİN
ÖZ YETERLİK ALGISINA ETKİSİ**

Bülent KOÇAK

**YÜKSEK LİSANS TEZİ
BİLGİSAYAR ve ÖĞRETİM TEKNOLOJİLERİ EĞİTİMİ
ANABİLİM DALI**

Danışman

Dr. Öğr. Üyesi Mesut TÜRK

**AMASYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

KASIM 2023

Yüksek Lisans Tezi Kabul ve Onay Sayfası

Bülent KOÇAK tarafından hazırlanan “Toxicode ve Google Blockly Uygulamalarının Öğrencilerin Bilgi İşlemsel Düşünme Becerisine ve Algoritma Geliştirmeye İlişkin Öz Yeterlik Algısına Etkisi” adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile Amasya Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalında YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Danışman: Dr. Öğr. Üyesi Mesut TÜRK

Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalı, Amasya Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum

Başkan : Doç. Dr. Fatih SALTAN

Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalı, Amasya Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum

Üye : Dr. Öğr. Üyesi Ahmet Berk ÜSTÜN

Bilgisayar Teknolojisi ve Bilişim Sistemleri Anabilim Dalı, Bartın Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum

Tez Savunma Tarihi: 14/11/2023

Jüri tarafından kabul edilen bu tezin Yüksek Lisans Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

.....

Doç. Dr. Ümit YILDIRIM

Fen Bilimleri Enstitüsü Müdürü

ETİK BEYAN

Amasya Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ıvtandar kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmada yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Bülent KOÇAK

TOXICODE VE GOOGLE BLOCKLY UYGULAMALARININ ÖĞRENCİLERİN
BİLGİ İŞLEMSEL DÜŞÜNME BECERİSİNE VE ALGORİTMA GELİŞTİRMEYE
İLİŞKİN ÖZ YETERLİK ALGISINA ETKİSİ
(Yüksek Lisans Tezi)

Bülent KOÇAK

AMASYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Kasım 2023

ÖZET

Bu çalışma, Toxicode ve Google Blockly uygulamalarının öğrencilerin bilgi işlemsel düşünme becerileri ve algoritma geliştirme öz yeterlik algısı üzerindeki etkisini incelemeyi amaçlamaktadır. Araştırma; iç ve dış geçerliği tehdit eden faktörleri kontrol etmek ve olası ön-test etkilerini gidermek için deneysel tasarım yöntemlerinden biri olan Solomon dört grup modeline göre yapılmıştır. Araştırmanın çalışma grubu Samsun ilinde ortaokul 5. Sınıfta öğrenim gören deney gruplarında 54, kontrol gruplarında 48 olmak üzere toplam 102 öğrenciden oluşmaktadır. Katılımcıların öğrenim görmekte olduğu mevcut dört şubeden ikisi Toxicode kullanan deney grubu, diğer ikisi ise Google Blockly kullanan kontrol grubu olmak üzere seçkisiz bir biçimde belirlenmiştir. Deney gruplarında, Toxicode program etkinlikleri uygulanırken; kontrol gruplarında, Milli Eğitim Bakanlığının Bilişim Teknolojileri ve Yazılım Dersi Öğretim Materyalleri kapsamında Google Blockly uygulama aracılığıyla öğretmen rehberinde çeşitli etkinliklere yer verilmiştir. Deneysel süreçte tüm gruplara son – test uygulanırken, ön – test sadece deney 1 ve kontrol 1 gruplarına uygulanmıştır. Veri toplama sürecinde “Bilgisayarca Düşünme Beceri Düzeyleri Ölçeği” ile “Algoritma Geliştirmeye İlişkin Öz Yeterlik Algısı Ölçeği” kullanılmıştır. Deneysel sürecin sonunda, gruplar arası kıyaslamalar için meta-analitik yaklaşımından yararlanılmıştır. Sonuçlar, Toxicode ve Google Blockly uygulamalarının her iki beceri üzerinde de olumlu bir etkiye sahip olduğunu göstermektedir. Karşılaştırmalı analiz sonuçlarında öğrencilerin algoritma geliştirmeye ilişkin öz yeterlik algılarının ve bilgi işlemsel düşünme becerilerinin anlamlı düzeyde arttırdığını ancak bu farkın kontrol grupları lehine daha fazla olduğu tespit edilmiştir. Öğrencilerin algoritma geliştirmeye ilişkin öz yeterlik algısı ve bilgi işlemsel düşünme becerileri artarken bu artışın istatistiksel olarak sadece algoritma geliştirmeye ilişkin öz yeterlik algısında kontrol grubu lehine anlamlı bir fark olduğu sonucuna ulaşılmıştır. Sonuç olarak blok tabanlı ve görsel programlama araçları, programlama öğrenmeyi daha kolay ve eğlenceli hale getirerek bilgi işlemsel düşünme, algoritma geliştirmeye yönelik özyeterlik gibi çeşitli becerilerin gelişimine katkı sağlamaktadır. Bu potansiyelle birlikte bu tarz araçlar, geniş bir kullanıcı kitlesine ulaşabilir. Etkili bir şekilde kullanıldığında, bu araçlar öğrencilerin bilgi işlemsel düşünme becerilerini geliştirmede, algoritma geliştirme konusundaki özgüvenlerini artırmada ve programlama becerilerini öğretmede önemli bir rol üstlenebilir ve gelecekte daha fazla kişinin dijital becerilere sahip olmasına katkıda bulunabilir.

Sayfa Adedi : 95
Anahtar Kelimeler : Toxicode, Blockly, Algoritma geliştirme, Öz yeterlilik algısı, Bilgi işlemsel düşünme
Danışman : Dr. Öğr. Üyesi Mesut TÜRK

THE EFFECT OF TOXICODE AND GOOGLE BLOCKLY APPLICATIONS ON
STUDENTS' COMPUTATIONAL THINKING SKILLS AND SELF-EFFICACY
PERCEPTIONS REGARDING ALGORITHM DEVELOPMENT

(M. Sc. Thesis)

Bülent KOÇAK

AMASYA UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

November 2023

ABSTRACT

This study aims to investigate the impact of Toxicode and Google Blockly applications on students' computational thinking skills and algorithm development self-efficacy. The research was conducted using the Solomon four-group design, an experimental design method, to control internal and external validity threats and mitigate potential pretest effects. The study group consisted of a total of 102 students, 54 in the experimental groups using Toxicode and 48 in the control groups using Google Blockly, from the 5th grade of middle schools in Samsun, Turkey. Two branches from the participating four branches were randomly assigned to the experimental groups using Toxicode, while the other two branches were assigned to the control groups using Google Blockly, all within the same curriculum. During the experimental process, Toxicode program activities were implemented in the experimental groups, while the control groups followed various activities guided by teachers through the Google Blockly application tool as part of the Ministry of National Education's Information Technologies and Software Course Teaching Materials. Post-tests were administered to all groups, whereas pre-tests were only conducted with the experimental group 1 and control group 1. Data was collected using the "Computer Thinking Skill Levels Scale" and the "Self-Efficacy Perception Scale for Algorithm Development". At the end of the experimental process, a meta-analytic approach was employed for intergroup comparisons. The results indicate that both Toxicode and Google Blockly applications have a positive impact on both skill sets. Comparative analysis results show a significant increase in students' self-efficacy perception for algorithm development and computational thinking skills. However, this improvement was found to be more significant in favor of the control groups. While students' self-efficacy perception for algorithm development and computational thinking skills increased, it was concluded that this increase only led to a significant difference in favor of the control group in terms of algorithm development self-efficacy perception. In conclusion, block-based and visual programming tools contribute to the development of various skills such as computational thinking and self-efficacy for algorithm development by making learning programming easier and more enjoyable. With this potential, these tools could reach a wide range of users. When these tools used effectively, they can play an important role in developing students' computational thinking skills, increasing their confidence in algorithm development, and teaching programming skills, contributing to more people having digital skills in the future.

Number of pages : 95

Keywords : Toxicode, Blockly, Algorithm development, Self-efficacy perception,
Computational thinking

Supervisor : Asst. Prof. Dr. Mesut TÜRK

ÖN SÖZ VE TEŞEKKÜR

Bu çalışma, çabalı bir sürecin ürünüdür; bir araştırmacının derinlemesine gözlemleri, analizleri ve tutkulu öğrenme çabalarının bir yansımasıdır. Öncelikle bu çalışmanın ortaya konma süreçlerinde bana bilgi birikimi ve akademik yaşam tecrübesi ile ufkumu geliştiren, her koşulda ve konuda desteğini esirgemeyen değerli danışmanım Dr. Mesut TÜRK'e teşekkürü bir borç bilirim.

Derin bilgi ve özgün fikirleriyle bilimsel bakış açımın gelişmesine ve aynı zamanda önerileri ile bu çalışmanın daha nitelikli hale gelmesine katkı sağlayan, her biri kendi alanında değerli katkılarıyla bu çalışmanın zenginleşmesine yardımcı olan değerli hocalarım Prof. Dr. Recep ÇAKIR, Prof. Dr. Özgen KORKMAZ, Dr. Volkan KUKUL ve Dr. Öğr. Üyesi Feray UĞUR ERDOĞMUŞ'a teşekkürlerimi sunarım.

Araştırma sürecinde her zaman yanımda olan, beni hep destekleyen ve bu süreci keyifli kılan sevgili öğretmen meslektaşlarım Nurullah AKSEL, Aydın KAPLAN, Mustafa SAĞLAM ve Hakan BAYIK'a teşekkürü bir borç bilirim.

Bu uzun süreçte benden desteğini esirgemeyen, bana sevmeyi öğreten, karşılaştığım tüm zorluklara benimle (benden daha çok) göğüs geren ve beni hep yüreklendiren sevgili eşim ve meslektaşım Yasemin KOÇAK'a sonsuz teşekkürlerimi sunuyorum. Hayatıma anlam katan bir diğer odak noktam olan Yiğit'im sen olmasan hiçbir şey bu kadar tamam olmazdı. Son olarak, bu çalışmanın eksiklikleri ve sınırlılıkları olduğunu unutmayalım. Her araştırma gibi, bu çalışmanın da sınırlamaları vardır ve daha fazla araştırmaya ihtiyaç duyulan alanlar bulunmaktadır. Umarım bu çalışma, okuyuculara, öğrencilere ve araştırmacılara değerli bir kaynak sunar ve çalışmanız için ilham verici bir başlangıç noktası oluşturur.

İÇİNDEKİLER

	Sayfa
ÖZET	iv
ABSTRACT.....	v
ÖN SÖZ ve TEŞEKKÜR.....	vi
İÇİNDEKİLER	vii
ÇİZELGELER DİZİNİ	ix
ŞEKİLLER DİZİNİ	x
SİMGELER VE KISALTMALAR DİZİNİ	xi
1. GİRİŞ	1
1.1. Problem Durumu	5
1.2. Araştırmanın Önemi	5
2. KURAMSAL ÇERÇEVE.....	7
2.1. Bilgi İşlemsel Düşünme Becerisi	7
2.1.1. Bilgi işlemsel düşünme becerisi ve programlama	9
2.2. Algoritma ve Algoritmik Düşünme	11
2.3. Öğrencilerin Algoritma Geliştirmeye Yönelik Öz Yeterlilik Algısı	14
2.4. Programlama, Bilgi İşlemsel Düşünme ve Algoritma Geliştirme İlişkisi	15
2.5. Programlama Öğretim Yaklaşımları.....	16
2.5.1. Toxicode uygulama platformu	18
2.5.2. Google Blockly uygulama platformu.....	24
2.6. İlgili Çalışmalar	26
3. YÖNTEM	28
3.1. Araştırma Modeli.....	29

Sayfa

3.2. Solomon Dört Gruplu Deneysel Model.....	29
3.3. Çalışma Grubu.....	31
3.4. Uygulama Süreci	32
3.5. Deneyisel Uygulama.....	32
3.6. Veri Toplanması	33
3.6.1. Bilgisayarca düşünme beceri düzeyleri ölçeği (BDÖ).....	33
3.6.2. Algoritma geliştirmeye ilişkin öz yeterlik algı ölçeği (AGYÖ)	33
3.7. Verilerin Analizi	34
4. BULGULAR.....	37
5. SONUÇ VE ÖNERİLER.....	45
5.1. Sonuçlar.....	45
5.2. Öneriler.....	48
5.2.1. Araştırmacılara yönelik öneriler	48
5.2.2. Uygulayıcılara yönelik öneriler	50
KAYNAKÇA.....	51
EKLER.....	68
ÖZGEÇMİŞ	95

ÇİZELGELER DİZİNİ

Çizelge	Sayfa
Çizelge 3.1. Solomon deneysel uygulama modeli (dört grup) (Braver ve Braver, 1988) ...	30
Çizelge 3.2. Araştırma modeline göre deneysel uygulama.....	31
Çizelge 3.3. Çalışma grubuna ait dağılım.....	32
Çizelge 4.1. BDÖ ve AYGÖ tanımlayıcı sonuçları.....	37
Çizelge 4.2. Ön test uygulanan sınıfların karşılaştırılmasına ilişkin ilişkisiz örneklem t-testi sonuçları.....	38
Çizelge 4.3. MANOVA (2X2) analizi deseni.....	38
Çizelge 4.4. Ön test bağımsız değişkenine göre BDÖ ve AYGÖ son test puanlarında MANOVA (2x2) sonuçları	39
Çizelge 4.5. Yöntem bağımsız değişkenine göre BDÖ ve AYGÖ son test puanlarında MANOVA (2x2) sonuçları	40
Çizelge 4.6. BDÖ ve AGYÖ testi için grupların düzeltilmiş son test puanları	42
Çizelge 4.7. BDÖ ve AGYÖ testi için ön test-son test puanlarının kontrol edildiği ANCOVA analizi sonucu.....	43
Çizelge 4.8. Deney_1 ve Kontrol_1 grupları bağımsız örneklem t-Testi fark puan analizi sonuçları	44

ŞEKİLLER DİZİNİ

Şekil	Sayfa
Şekil 2.1. Zsako ve Szlavi (2012)'nin geliştirdiği algoritmik düşünme beceri düzeyleri....	13
Şekil 2.2. Toxicode uygulama arayüzü.....	19
Şekil 2.3. Toxicode öğrenme modülleri.....	21
Şekil 2.4. Silent Teacher etkinlik modülü.....	22
Şekil 2.5. Compute It etkinlik modülü.....	23
Şekil 2.6. Little Dot Adventure etkinlik modülü	24
Şekil 2.7. Google Blockly uygulama arayüzü	25
Şekil 3.1. Solomon meta-analitik basamakları	34
Şekil 4.1. Grupların ön test alma durumlarına göre AYGÖ testi son test puanlarında değişim.....	39
Şekil 4.2. Grupların ön test alma durumlarına göre BDÖ testi son test puanlarında değişim.....	40
Şekil 4.3. Grupların uygulama yöntemine göre AYGÖ testi son test puanlarında değişim	41
Şekil 4.4. Grupların uygulama yöntemine göre BDÖ testi son test puanlarında değişim ...	42

SİMGELER VE KISALTMALAR DİZİNİ

Bu çalışmada kullanılmış bazı simgeler ve kısaltmalar, yanda açıklamaları verilmek üzere aşağıda listelenmiştir.

Kısaltmalar	Açıklama
BDÖ	Bilgisayarca Düşünme Beceri Düzeyleri Ölçeği
AYGÖ	Algoritma Geliştirmeye İlişkin Öz Yeterlik Algısı Ölçeği
BTYD	Bilişim Teknolojileri ve Yazılım Dersi
BİD	Bilgi İşlemsel Düşünme
MEB	Millî Eğitim Bakanlığı
TTKB	Talim Terbiye Kurulu Başkanlığı
EARGED	Milli Eğitim Bakanlığı Eğitim Araştırma Dairesi Başkanlığı
K12	İlkokul 1. Sınıftan 12. Sınıfa kadar olan eğitim dönemi
$\bar{X}$	Aritmetik ortalama
ss	Standard sapma
F	F istatistiği (gruplar arasındaki varyans farkı)
KT	Kareler toplamı
KO	Kareler ortalaması
S	Standard hata
η^2	Kısmi (Partial) etki büyüklüğü
df	Serbestlik derecesi
n	Birey sayısı

1. GİRİŞ

Günümüz dünyası hızın ve kendini bu hıza bağlı olarak yenileyenlerin ön plana çıkacağı bir yapıya dönüşmektedir. Teknoloji okuryazarlığı konusunda yeterli, problem çözme konusunda yeteneği gelişmiş ve yaratıcılığını hayatın her alanında kullanabilen, bilgi işlemsel düşünme becerisine sahip kişiler bu yeni ve hızla değişen çağa ayak uydurabileceklerdir (Özel, 2019). Bilgi İşlemsel Düşünme (BİD) kavramını Papert (1980) alanyazına kazandırmış olsa da bu kavramın geniş kitlelerce öğrenilmesi Wing (2006)'ın araştırmalarının sonucunda olmuştur.

Çeşitli kaynaklarda, “Computational Thinking” terimi, Türkçe olarak “kompütasyonel düşünme” (Aldağ ve Tekdal, 2015; Şahiner ve Kert, 2016; Demir ve Seferoğlu, 2017), “bilgisayarsal” (Doğan ve diğerleri, 2015), “bilgisayarca” (Kormaz ve diğerleri, 2015) şeklinde ifade edilmiş ve akademik yayınlarda da kullanılmıştır. Papert (1980), bilgi işlemsel düşünce becerisinin, bireylerin bilişsel gelişimini, öğrenme ve düşünme becerilerini destekleyerek duyuşsal gelişim açısından öğrenenlere katkı sağlayabileceği görüşünü ortaya koyar iken, Wing (2008) tarafından yapılan araştırmada bilgi işlemsel düşünmenin, bireylerin problem çözümünde kullandıkları bir çeşit analitik düşünme olduğunu ifade etmiştir. Futschek (2006) ise bilgi işlemsel düşünme kavramının problemi analiz edebilme, anlama, ifade etme, çözüm önerileri getirebilme ve geliştirilen çözümlere yönelik algoritmalar oluşturabilme gibi alt becerilerden oluştuğunu ifade etmektedir.

Teknolojinin içine doğan yeni kuşak öğrencilerde bulunması gereken becerileri MEB, “21. Yüzyıl Öğrenci Profili” çalışmasında; problem çözme, analitik ve eleştirel düşünme, yaratıcı düşünme becerilerine sahip, araştıran ve sorgulayan, bilgiye ulaşabilen ve bilgiyi yönetebilen, teknolojiyi etkili ve verimli kullanabilen, teknoloji kullanımında seçici olan, fark oluşturabilen ve bilimsel okur-yazar olabilen bireyler olarak tanımlamıştır (EARGED, 2011). Yeni kuşaklar geleneksel yetenek sergilemelerinin dışında yenilikçi yöntemler ile yeteneklerini yaratıcılıklarıyla birleştirerek yeni fırsatların içinde yaşamaktadırlar (Yünkül, Durak ve Çankaya, 2018). Günümüz dünyasında bir gencin yaratıcılığını yetenekleriyle birleştirerek sergileyebileceği en etkili yollardan birisi de yaygınlaşmaya başlayan kodlamadır.

İnsanın en büyük yeteneği olan aklını bilgisayar veya diğer teknolojik araçların yardımıyla geliştirmesi günlük hayatın ayrılmaz bir parçası haline gelmiştir (Barr ve diğerleri, 2011). Beşinci sınıf Bilişim Teknolojileri ve Yazılım ders (BTYD) müfredatı incelendiğinde öğrencilerin bilgi işlemsel düşünme becerilerinin alt bileşenlerinin bazılarının halihazırda öğretim programında yer aldığı görülmektedir (MEB, 2018). Ayrıca öğrencilerin bu konuda gerekli olan becerilere sahip olduklarının bilinmesi değilse giderilmesi günümüzde bir tercih değil zorunluluk haline gelmiştir (Barr ve diğerleri, 2011).

Bilgi işlemsel düşünme sayesinde öğrenciler öğrenme süreçlerini anlayabilmekte, problemleri anlamlandırarak bunlara çözüm üretecek algoritmalar oluşturabilmektedir. Bu becerileri öğrencilerin edinmesini sağlamak okul sisteminin artık günümüzde çocuklara sunması gereken hizmetlerden birisidir. Kodlamanın çocuklara erken yaşlarda verilmesi onların hem özgüvenlerini artıracak hem de erken öğrenilen bir becerinin bilişsel süreçte daha güçlü ve uygulanabilir olmasını sağlayacaktır (Kert ve Uğraş, 2009).

Okul çağında ilk defa bilgi işlemsel düşünme ve kodlama ile karşılaşan çocuklar bocalama yaşayabilmektedir. Bunun en önemli sebebi kodlama mantığını anlamada güçlük yaşamalarıdır (Çağiltay ve Fal, 2014). Bu nedenle hangi eğitimin verileceği kadar o eğitimin nasıl ve hangi yöntemler ile verileceği de önem kazanmıştır.

Günümüzde problem çözme, algoritma, algoritmik düşünme, mantıksal akıl yürütme, soyutlama ve ayrıştırma bilgi işlemsel düşünme beceri tanımı içerisine dâhil edilen kavramlar arasındadır (Selby, 2014). Üzümcü ve Bay (2018)' a göre ise parçalara ayırma, örüntü bulma/ tanımlama, soyutlama, algoritma ve test etme/ hata ayıklama bilgi işlemsel düşünmenin en sık karşılaşılan boyutlarıdır. Soyutlama; önemli noktaları göz ardı etmeden gizlenecek detaylar saptanmak olarak tanımlanırken (Csizmadia ve diğerleri, 2015) değerlendirme, kullanılan komutların uygunluğu ve sonuca ulaştıran çözümün kalitesini ortaya koymak (Marcelino ve diğerleri, 2017) olarak ifade edilmektedir. Ayrıştırma ise verileri veya problemleri, daha küçük alt bileşenlere ve yönetilebilir bölümlere ayırmaktır. Daha sonra bu ayrıştırılmış alt bileşenler veya bölümler ayrı ayrı gözden geçirilebilir, geliştirilebilir ve yeniden değerlendirilebilir (Csizmadia ve diğerleri, 2015). Genelleme, kullanılan uygulamalardan daha geniş uygulanabilirliğe geçiş yeteneğidir (Selby, 2013). Mantıksal akıl yürütme ise bireylerin sonuca ulaşmak için, koşullara göre çıkarımlarda

bulunmasını sağlayan düşünme becerisidir (Yang ve diğerleri, 2016). Algoritmik düşünme; algoritmaları anlayıp, değerlendirip yeniden üretme yeteneği olarak tanımlanmaktadır (Brown, 2015).

Denning (2003) farklı perspektiflerden yaklaşarak bilgi işlemsel düşünme kavramını, bilgi girişi ile algoritmanın oluşturulması ve uygulanmasını içeren bir süreç olarak tanımlamaktadır. Algoritma terimi, ilk olarak Orta Asya’da yaşamış olan Muhammed ibn Musa Al – Khwarizmi (El- Harezmi) tarafından 9. yüzyılda kullanılmıştır. Onun soyadından yola çıkılarak bu kavrama algoritma ismi verilmiştir (Schneider ve Gersting, 2018). Algoritma, hedeflenen amaca ulaşmak için tasarlanan işlem adımlarına izlenilen yola verilen isimdir (Aytekin, Sönmez – Çakır, Yücel ve Kulaöz, 2018). Sonuca ulaşabilmek için, oluşturulacak algoritmalarda çözüm yöntemleri ayrıntılı bir şekilde belirlenmelidir. Aslında bir sorunu bilgisayarda veya hayatta çözerken atılması gereken adımları belirleyen program değil o programın nasıl olması gerektiğini belirleyen yöntemdir. Yani algoritma sorunu çözmek için oluşturulan yöntemlerin bütünüdür (Sedgewick ve Wayne, 2011). Akademik çalışmalar içerisinde öğrencilerin “Algoritma Geliştirmeye İlişkin Öz Yeterlik Algısı” konusunda alan yazının yetersiz olduğu ifade edilmektedir (Ramazanoğlu, 2021).

Son yıllarda programlamanın önemi giderek artmaktadır ve birçok ülke, okul öncesinden başlayarak çeşitli eğitim kademelerinde programlama faaliyetlerine yer vermektedir. Ancak, programlamanın soyut yapısı ve söz dizimi, başlangıç seviyesindeki öğrenciler için karmaşık ve sıkıcı olabilir. Bu nedenle, öğrencilerin programlamaya ait temel yapıları öğrenmelerini kolaylaştırmak için yapılan çalışmalar sonucu geliştirilmiş çeşitli çevrimiçi ve çevrimdışı pek çok blok tabanlı kodlama platformları bulunmaktadır.

Çevrimiçi blok tabanlı kodlama platformları, içerdikleri oyun temelli etkinlikler, basit seviyelerden karmaşığa, kolaydan zora doğru ilerleyerek öğrencilere programlamayı eğlenceli bir şekilde öğrenme fırsatı sunmaktadır. Bu araçlarla gerçekleştirilen etkinliklerde ortaya çıkan hatalar, sunulan çeşitli ipuçları ve geri bildirimlerle çözüme kavuşturulabilmekte; ayrıca öğretmen yardımına olan ihtiyacı azaltarak öğrencilerin bağımsız bir şekilde programlama yapmalarını desteklemektedir.

Çevrimiçi blok tabanlı kodlama platformları genel olarak değerlendirildiğinde, okul öncesinden üniversite seviyesine kadar farklı yaş gruplarında daha çok algoritma oluşturma ve temel programlama becerileri kazandırmaya yönelik kullanıldıkları söylenebilir. Türkiye’deki programlama eğitimleri incelendiğinde ilköğretim düzeyindeki Bilişim Teknolojileri ve Yazılım dersinde Google Blockly, Scratch ve code.org gibi çevrimiçi blok tabanlı kodlama araçlarının sıklıkla tercih edildiği görülmektedir (Dönmez Usta ve Turan Güntepe, 2019; Yüksel, 2017).

Günümüzde kullanılan çevrimiçi blok tabanlı kodlama platformlarının birçoğunda (code.org, Google Blockly vb.), kullanıcıların etkinliklerdeki görevleri yerine getirmek için gerekli olan kod bloklarını kullanmaları beklenirken, Toxicode platformunda kullanıcıların kod bloklarını temsil edecek şekilde verilen ifadelerle göre görevi tamamlamaları beklenmektedir. "Kod bloklarını kullanarak görev tamamlamak" ve "kod bloklarını temsil edecek şekilde verilen ifadelerle göre görev tamamlamak" arasındaki fark, öğrenme sürecinde farklı beceri setlerini hedeflemekte ve öğrencilere farklı türde bilgi işleme yetenekleri kazandırmaktadır. Kod bloklarını kullanarak görev tamamlama platformlarında, öğrencilere genellikle belirli bir görevi yerine getirmek için kullanılacak kod blokları sağlanır. Öğrenciler, belirli bir algoritma oluşturmak için mevcut blokları sıralar ve bu blokları kullanarak bir program oluşturur. Bu yöntem, algoritmik düşünce, problem çözme ve mantıksal akıl yürütme becerilerini geliştirmeye odaklanır. Kod bloklarını temsil edecek şekilde verilen ifadelerle göre görev tamamlama platformlarında ise öğrencilere belirli bir görev için kod bloklarını temsil eden ifadeler sunulur, ancak bloklar kendileri verilmez. Öğrenciler, verilen ifadelerle göre adım adım ilerleyerek sonuca ulaşması istenir. Bu yöntem, öğrencilere algoritmayı uygulama konusunda daha yaratıcı bir yaklaşım benimsemelerine olanak tanır. Literatür taramalarında, kod bloklarını kullanarak görev tamamlamayı amaçlayan çevrimiçi blok tabanlı kodlama platformlarının öğrencilerin bilgi işlemsel düşünme becerileri ve algoritma geliştirmeye ilişkin öz yeterlik algılarına yönelik araştırmalara rastlanırken; kod bloklarını temsil edecek şekilde verilen ifadelerle göre görev tamamlamayı amaçlayan çevrimiçi blok tabanlı kodlama platformlarının bu iki beceriye etkisi için yapılan araştırmalara rastlanılmamıştır. Bu nedenle bu araştırmanın ilgili alanyazına katkı sağlayacağı öngörülmektedir.

1.1. Problem Durumu

Bu araştırmanın amacı ortaokul 5. sınıf bilişim teknolojileri dersinde Toxicode uygulamasına ait etkinliklerin, Google Blockly uygulamasındaki etkinliklere kıyasla öğrencilerin bilgi işlemsel düşünme becerisine ve algoritma geliştirmeye ilişkin öz yeterlik algısına etkisini incelemektir.

Bu amaca bağlı olarak aşağıdaki denenceler test edilmiştir:

1. Toxicode online platformundaki uygulamalar kullanılarak yürütülen bilişim teknolojileri dersinde, öğrencilerin bilgi işlemsel düşünme becerileri anlamlı düzeyde artar.
2. Toxicode online platformundaki uygulamalar kullanılarak yürütülen bilişim teknolojileri dersinde, öğrencilerin algoritma geliştirmeye ilişkin öz yeterlik algısı anlamlı düzeyde artar.

1.2. Araştırmanın Önemi

Dijital çağda eğitim paradigması, öğrencilerin bilgi işlemsel düşünme becerilerini geliştirmek için blok tabanlı kodlama uygulamalarının eğitimde kullanılmasına doğru kaymıştır (Lye ve Koh, 2014). Eğitimde kullanımının artan önemine odaklanarak, problemlere çözüm üretme, eleştirel düşünme, mantıksal düşünme, derse katılım, okul motivasyonu gibi birçok alanda bu uygulamaların potansiyel etkisini incelemek çoğu zaman araştırmaların konusu olmuştur (Ramazanoğlu, 2021).

Hsu, Chang ve Hung (2018) tarafından 2006 – 2017 yılları arasındaki araştırmaların derlendiği uluslararası bir çalışma, BİD konusunda işbirlikçi öğrenme, oyun tabanlı öğrenme ve probleme dayalı öğrenme gibi konularda çalışmaların yapıldığını ortaya koymuştur. Scratch ve Google Blockly gibi blok tabanlı uygulamaların öğrencilerin eğitim süreçlerinin büyük bir kısmında yaygın olarak kullanıldığı bilinmektedir. Bu platformların öğrencilerin problem çözme, bilgi işlemsel düşünme, algoritma geliştirme, öz yeterlilik algısı gibi pek çok beceri düzeyinde etkilerinin incelendiği çalışmalar yapılmaktadır. Bu çalışma, App Inventor, code.org, Scratch ve Google Blockly gibi blok tabanlı kodlama platformlarının öğrencilerin bilgi işlemsel düşünme becerisi ve algoritma geliştirme süreçlerindeki öz yeterlilik algısı üzerindeki etkisini inceleyen az sayıda araştırmadan biridir. Ancak mevcut

alan yazında özellikle Toxicode benzeri platformların öğrencilerin BİD ve algoritma geliştirme süreçlerindeki öz – yeterlik algıları üzerinde nasıl bir etkiye sahip olduğunu anlama noktasında yeterli sayıda araştırmanın olmadığı görülmektedir (Kalelioğlu ve Gülbahar, 2014; Yükseltürk ve Altıok, 2015). Bu çalışma, Toxicode platformunun öğrencilerin bilgi işlemsel düşünme becerisi ve algoritma geliştirme süreçlerinde öz-yeterlik algıları üzerindeki etkilerine dair bilgiler sunmayı hedeflemektedir. Ayrıca, Millî Eğitim Bakanlığı'nın ders kitabında yer alan Blockly platformu da araştırma kapsamında incelenmektedir (Gülbahar ve diğerleri, 2018). Bu nedenle araştırma en uygun öğretim ortamının belirlenmesi anlamında eğitimcilere ve araştırmacılara yol gösterebilecektir.



2. KURAMSAL ÇERÇEVE

Bu bölümde araştırmanın kuramsal bilgileri ve konuyla ilgili yapılan önceki araştırmalar bulunmaktadır.

2.1. Bilgi İşlemsel Düşünme Becerisi

Bilgisayarlar ve dijital teknolojilerin hayatımızın hemen her alanında yaygınlaşmasıyla birlikte, bilgi işlemsel düşünme becerisi gibi bazı yetkinlikler giderek daha önemli hale gelmektedir. Bilgi işlemsel düşünme kavramı bilimsel terim olarak alanyazında ilk kez Wing (2006) tarafından kullanılmış olmasına rağmen, bu terimin kökenleri Seymour Papert'in (1980) "Mindstorms" isimli eserine kadar dayanmaktadır. Papert (1996) bilgisayarın geometri problemlerinde nasıl kullanılabileceği üzerine tartışırken problem ve çözümü arasındaki ilişkiyi açıklamış, aynı zamanda bilgi işlemsel düşünmede bilginin yapılandırılmasında kullanılabileceğini göstererek bu konuda öncülük etmiştir. Papert (1980, 1996) bilgisayarın eğitimde ve özellikle matematik problemlerinin çözüm sürecinde, verimli bir iletişim aracı olarak kullanılmasının bilgi işlemsel düşünme mantığı ile uyumlu olduğunu belirtmiştir.

Bugün, bilgisayar ve internet teknolojileri aracılığıyla veri işleme, sistem otomatikleştirme ve daha düşük maliyetli hızlı üretim gibi olanakların sunduğu yararlar, bilgi işlemenin her alanda yaygın hale gelmesinin bir göstergesidir. Bu bağlamda, Furber (2012), bireylerin kendi çalışma alanlarında karşılaştıkları sorunları çözebilmeleri için bilgi işlem kapasitesinin önemini ve bilgi işlem temelli sistemleri anlama gerekliliğini vurgulamıştır. Bilgi işlemsel düşünme, bu bağlamda, bilgisayar bilimleri temelinde problem çözme, sistem tasarlama ve insan davranışını anlama kavramları kullanarak önem kazanmıştır (Wing, 2006).

Bilgi işlemenin gücü ve sınırları üzerinde durarak bilgi işlemsel düşüncenin önemine dikkat çeken Wing, her birey için temelde geliştirilmesi gereken önemli bir beceri olduğunu ifade etmiş (Wing, 2006), daha sonra bilgi işlemsel düşünmenin uygulama alanlarını genişleterek, matematiksel düşünme, mühendislik alanında düşünme ve sistem düşünmeyi kapsayan analitik düşünme biçimi olduğunu belirtmiş, soyutlama ve otomasyon kavramları üzerinde önemle durmuştur (Wing, 2008). Bu noktada Denning (2009), bilgi işlemsel düşünmenin bilgisayar biliminde olduğu gibi tüm disiplinler için de uygulama alanı olduğunu

vurgulamıştır. Bu bakış açısı, Bundy'nin (2007) bilgi işlemsel düşünmenin, hesaplamanın sunduğu bir yetenek olarak tüm disiplinler için yeni bir sorgulama biçimi olduğunu ifade etmesiyle eşleşmektedir. Ayrıca; Cuny, Snyder ve Wing (2010'dan alıntı yapan Wing, 2010, s.1), bilgi işlemsel düşünmeyi, bilgi işlem biriminin etkin biçimde uygulayabilmesi için sorunların tanımlanma, çözümlerin bulunma sürecini kapsayan bir düşünce biçimi olarak nitelendirmişlerdir.

Alanyazında yapılan incelemelerde başka yazarlar tarafından bilgi işlemsel düşünmenin çeşitli tanımlamalarına da rastlanmaktadır. Bu tanımlamalar değerlendirildiğinde, bilgi işlemsel düşünme becerisinin bir yetenek ve düşünme süreci olduğunda hemfikir olunmasına rağmen kavramın kesin tanımı konusunda farklı yaklaşımların olduğu görülmüştür. Örneğin, Aho (2012) bunu, bir sorunun çözümü için gerekli bilgi işleme basamaklarının tespit edilmesi ve algoritmaların çıkarılmasını kapsayan bir düşünme süreci olarak ifade ederken, Yadav ve diğerleri (2014) bu kavramı, problemin temel yapısını anlamak ve çözümünün sistemli olarak gerçekleştirilmesini kapsayan bir bilişsel süreç olarak tanımlamıştır. Csizmadia ve diğerleri (2015) tarafından yapılan tanıma göre, bilgi işlemsel düşünme, mantıksal akıl yürütme sürecini kullanarak problemleri çözmeyi, ürünleri, sistemleri ve süreçleri daha iyi anlamayı amaçlayan bir bilişsel süreçtir. Korkmaz ve diğerleri (2015) ise bilgi işlemsel düşünmeyi, günlük yaşamda karşılaşılan problemleri çözmek için bilgisayarların etkin bir şekilde kullanılabilmesi için gereken bilgi, beceri ve tutumları içeren bir kavram olarak belirtmiştir.

Bilgi işlemsel düşünme, geçmişten günümüze uzanan bir kavram olduğu için, karmaşık bir yapıya sahiptir ve bu da kavramın anlamlandırılmasını zorlaştırmaktadır (Hsu ve diğerleri, 2018; Tang ve diğerleri, 2020; Wing, 2008). Bu nedenle literatürde, bilgi işlemsel düşünmeyi belirli bir bağlam çerçevesinde ele alarak tanımlamanın önemi vurgulanmaktadır (Grover ve Pea, 2018; Tang ve diğerleri, 2020; Weintrop ve diğerleri, 2016). Ek olarak, bu tanımlamalar, bilgi işlemsel düşünmeyi daha geniş bir çerçevede değerlendirerek çeşitli yönleri ve özellikleriyle anlaşılabilir hale getirmeye çalışmaktadır (Hsu ve diğerleri, 2018; Korkmaz ve diğerleri, 2017).

2.1.1. Bilgi işlemsel düşünme becerisi ve programlama

Gelecekte, öğrencilerin hangi becerilere sahip olması gerektiği konusunda birçok araştırma yapılmakta ve yapılmaya devam etmektedir. Bu çalışmalar arasında en sık rastlanana, 21. yüzyıl becerileri kavramıdır (Sayın ve Seferoğlu, 2016; BatteforKids, 2023). Bu yetkinlikler arasında yaratıcılık ve inovasyon, eleştirel düşünme ve problem çözme, iletişim, işbirliği, bilgi ve teknoloji okuryazarlığı, evrensel yetkinlikler gibi beceriler öne çıkmaktadır. Bu bağlamda, bilgi ve teknoloji okuryazarlığı içinde yer alan programlama becerisi de öne çıkan beceriler arasında yer almaktadır.

Yazılım geliştirme (programlama), bir sorunun çözümüne yönelik, tasarım aşamasından uygulamaya kadar olan tüm adımları ve algoritmik düşünme, eleştirel düşünme, yaratıcı düşünme, yansıtıcı düşünme ve bilgi işlemsel düşünme gibi üst düzey düşünme becerilerini içeren bir süreçtir (Akçay ve Çoklar, 2016; Brennan, Resnick, 2012; Duncan, Bell ve Tanimoto, 2014; Fessakis ve diğerleri, 2019; Morelli ve diğerleri, 2011). İçinde barındırdığı algoritmik yapılar sayesinde günlük yaşamın bir parçası olan işlemlerin adımlarının zihinsel süreçlerden geçirilerek gerçekleştirilmesini sağlayan bir olgudur.

Bilgi işlemsel düşünme, problem çözme, analitik düşünme, mantıksal akıl yürütme ve yaratıcılık gibi bilişsel yetenekleri içerir (Wing, 2006). Programlama sürecinde ise bu yeteneklerin kullanılması gerekmektedir. Bir programcı, bir problemi analiz etmeli, uygun bir çözüm stratejisi belirlemeli, adımları mantıklı bir şekilde sıralamalı ve sonuçları değerlendirmelidir. Bu süreçte, bilgi işlemsel düşünme becerileri, programlamanın temelini oluşturur.

Programlama ile bilgi işlemsel düşünme becerisi arasında pozitif bir ilişki olduğu bilinmektedir (Oluk ve Korkmaz, 2016). Ancak programlama, algoritma geliştirme sürecini içerir ve algoritma geliştirme bilgi işlemsel düşünmeyi gerektirir. Bir programcı, bir problemi analiz etmek, uygun bir algoritma oluşturmak, adımları düzenlemek ve sonuçları değerlendirmek için bilgi işlemsel düşünme becerilerini kullanır. Bu nedenle, programlama eğitimi ve uygulamaları, öğrencilerin hem programlama becerilerini geliştirmelerine hem de bilgi işlemsel düşünme becerilerini güçlendirmelerine yardımcı olur. Programlama aktiviteleri, öğrencilerin algoritmik düşünme becerilerini, yansıtıcı düşünme becerilerini,

eleştirel düşünme becerilerini, problem çözme stratejilerini ve mantıksal akıl yürütme becerilerini geliştirmelerine olanak tanımaktadır (Akçay ve Çoklar, 2016; Duncan, Bell ve Tanimoto, 2014).

Yapılan çeşitli araştırmalar, programlamaya yönelik araçların öğrencilerin üst düzey düşünme becerileri olarak kabul edilen algoritmik düşünme (Fessakis ve diğerleri, 2019), yaratıcı düşünme (Morelli ve diğerleri, 2011) ve bilgi işlemsel düşünme (Brennan and Resnick, 2012; Fessakis ve diğerleri, 2019) konularındaki gelişimlerini desteklediğini göstermektedir. Programlama etkinlikleri, öğrencilerin analitik düşünme yeteneklerini, problem çözme stratejilerini, mantıksal akıl yürütme becerilerini ve yaratıcılıklarını desteklemektedir (Akçay ve Çoklar, 2016; Duncan, Bell ve Tanimoto, 2014).

Yazılım geliştirmenin etkilerini inceleyen diğer pek çok çalışmada birbirine yakın ve benzer sonuçlar olduğu görülmüştür. Örneğin, Papadakis, Kalogiannakis ve Zaranis (2016) okul öncesi öğrencileriyle gerçekleştirdikleri bir çalışmada, ScratchJr adlı blok tabanlı programlama aracının temel yazılım geliştirme kavramlarının öğrenilmesine ve bilgi işlemsel düşünme becerisinin gelişimine katkı sağladığını bulmuşlardır. Wei ve diğerleri (2021) ise ilkokul öğrencileriyle yaptıkları bir deneysel çalışmada, kısmi eşli programlamanın öğrencilerin bilgi işlemsel düşünme becerilerini ve öz-yeterliklerini anlamlı bir şekilde geliştirdiğini bulmuşlardır. Bu çalışmalarda programlama eğitiminin farklı araçlar ve yöntemler kullanılarak gerçekleştirildiği görülmektedir. Örneğin, Lego görsel programlama aracı, Scratch ve Light Bot gibi araçlar kullanılmış ve bu araçların öğrencilerin bilgi işlemsel düşünme becerilerini desteklediği ve geliştirdiği bulunmuştur (Gouws ve diğerleri, 2013; Pinto-Llorente ve diğerleri, 2016). Bazı çalışmalarda ise bilgi işlemsel düşünme becerisi, programlama veya bilişim teknolojileri kullanım becerisi olarak ele alınmıştır. Ancak bilgi işlemsel düşünme, bilgisayar bilimlerinin felsefesini anlamlandırmayı ifade eden kavramlara odaklanan bir süreçtir. Programlama ise bu mekanik sürecin uygulanmasından önce başlayan zihinsel bir süreçtir. Bu nedenle programlama, bilgi işlemsel düşünme becerisini destekleyen bir eğitim bağlamı olarak ele alınmalıdır (Buitrago Florez ve diğerleri, 2017; Grover ve Pea, 2018).

Sonuç olarak, programlama eğitiminin algoritmik düşünme, eleştirel düşünme, yaratıcı düşünme, yansıtıcı düşünme ve bilgi işlemsel düşünme gibi üst düzey düşünme becerilerinin

gelişimine katkı sağladığı birçok çalışma tarafından desteklenmektedir. Bu çalışmalarda farklı araçlar ve yöntemler kullanılarak yapılan programlama eğitimi, öğrencilerin problem çözme, analitik düşünme ve yaratıcılık gibi üst düzey düşünme becerilerini geliştirmelerine yardımcı olmaktadır. Bu nedenle programlama, eğitim süreçlerinde bilgi işlemsel düşünme becerisini destekleyen bir araç olarak kullanılabilir (Barr ve Stephenson, 2011).

2.2. Algoritma ve Algoritmik Düşünme

Algoritma, genel olarak bilgisayar programlamaya ait bir terim olarak kullanılır. Algoritma, işletildiklerinde belirli bir görevi, açık ve etkili bir şekilde gerçekleştirilebilen tasarlanmış adımların bir demeti olarak tanımlanır (Schneider ve Gersting 2018). Algoritma kelimesinin tarihsel süreci El- Harezmi'ye dayanmaktadır (Knuth, 1969). 9. Yüzyılda Orta Asya'da yaşamış bir matematikçi olan El- Harezmi, algoritmayı problemin çözüm süreci için üretilmiş metotlar olarak tanımlamaktadır (Sedgewick ve Wayne, 2011). Bu metodun işletilmesini bir bilgi işleme birimi üstlenir. Bu, spesifik olarak algoritma bilgi işleme birimi demek değildir. Bilgi işleme birimi bir bilgisayar olarak düşünülürse, algoritma farklı programlama dillerinde çalışabilirken aynı zamanda farklı bilgisayarları da çalıştırabilir. Burada problemin çözüm yolunun belirleyicisi yazılan programdan çok metodun kendisidir.

Algoritma sadece bilgisayar programlamada kullanılıyor olsa da, algoritmik düşünme programlamadan bağımsız bir şekilde geliştirilebilir bir özelliğe sahiptir (Otaran, 2017). Algoritma, bilgi işlemsel düşünmenin alt bileşenleri arasında gösterilmektedir. Bu nedenle programlama eğitimi ile ilişkisi aşikârdır (Israel ve diğerleri, 2015). Algoritmaların her durumda tam ve doğru olarak çalışmama ihtimali olduğundan dolayı algoritma aşamalarının teker teker kontrol edilmesi gereklidir ve gereken durumlarda sıralamanın değiştirilmesi, deneme – yanılma yapılması gibi çözümlerin üretilmesi gerekir. Farklı durumlarda ise verilen bir görev için öğrenci grubundaki tüm bireylerin farklı algoritmalar geliştirerek buldukları çözümleri birbirleriyle kıyaslayarak en uygun çözüme ulaşmaları gerekir (CSTA, 2017).

Algoritmaların bilgisayar biliminde önemli bir yerinin olduğu kabul edilmektedir (Schneider ve Gersting, 2018). Özellikle otomasyon sistemleri pek çok alanda işleri kolaylaştırarak hem zamandan hem de maliyetlerden tasarruf etmeye olanak sağlamıştır. Örneğin ülke genelinde

yapılan çeşitli sınavların bir bilgi işlem birimi tarafından ve arka planında çalışan sağlam bir algoritma ile kısa sürede sonuçlandırılması, kontrollerinin hızlıca yapılabilmesi büyük kolaylıklar sağlamaktadır.

Algoritmik düşünme, bilgisayar bilimleri ve uygulamaları ile ilgili problem çözmek için kullanılabilecek temel beceriler olarak tanımlanmaktadır (Syslo ve Kwiatkowska, 2013). Algoritmik düşünme programlamadan bağımsız geliştirilebilir. Algoritmik düşünme, farklı disiplinlerde farklı anlamlara gelmektedir. Knuth'a (1969) göre algoritmik düşünme bilgi işlemsel düşünme anlamına gelmektedir. Matematik eğitiminde, yaratıcı düşünmeye karşılık olarak işlemsel düşünme anlamında kullanılmaktadır (Jonsson ve diğerleri, 2014). Güncel anlamda alan yazında algoritmik düşünmenin, belirli problemlerin çözümünde bilgi işlem birimi aracılığıyla üretilen çözüm adımlarından oluştuğu şeklinde ifade edilir. Kurallar ve bu kuralların işletilmesi için gerekli adımlardan oluşan bir anlamlandırma yolu ve becerisidir (Csizmadia ve diğerleri, 2015). Başka bir tanımda "yeni bir algoritma oluşturmak ve mevcut bir algoritmayı anlamak için gereken farklı beceriler" şeklinde açıklanmıştır. Bu beceriler şöyle sıralanmaktadır (Futschek, 2006):

- Verilen problemin analiz edilmesi becerisi,
- Problemin açıkça ifade edilme becerisi,
- Verilen bir problem için gerek duyulan temel işlemleri saptama becerisi,
- Temel işlemler aracılığıyla doğru algoritmanın inşa edilme becerisi,
- Problemin olası özel ve genel tüm durumlar açısından ele alınma becerisi,
- Bir algoritmanın verimliliğinin artırılması becerisi

Algoritma geliştirmek tek başına bir bütün olarak ele alınabileceği gibi yeterlilik seviyeleri olarak da düşünülebilir. Algoritmik düşünme becerisi için Zsako ve Szlavi (2012) tarafından yedi seviye (Şekil 2.1) belirlenmiştir:



Şekil 2.1. Zsako ve Szlavi (2012)’nin geliştirdiği algoritmik düşünme beceri düzeyleri.

Futschek (2006), algoritmik düşüncenin geliştirilmesi sürecinde bir programlama diline bağlı kalmanın gerekli olmadığı düşüncesini desteklemektedir. Öğrencilerin programlamanın özelliklerine odaklanmaktan algoritma tasarlamaya yeterince vakit bulamadıklarından yakınmaktadır. Bu sebeple öğrencilerin kendi yetenek seviyelerine uygun sözde kod ve algoritmalar ile kullanarak, algoritmik düşünmelerini geliştirmelerinin daha doğru olacağı görüşündedir (Futschek, 2006). Bilgisayarsız bilgisayar bilimi hareketi (Henderson, 2008), bilgisayar kullanmadan algoritmik düşünmenin geliştirilmesi düşüncesini öne sürmektedir. Bilgisayar kullanmadan algoritmik düşünme becerisi, bilgi işlemsel düşünme üzerine geliştirilmiş bir etkinlik olan ve uluslararası “Bebras Computing Challenge” etkinliğidir. Bu etkinlikler Türkiye’de “Bilge Kunduz” adı altında (<https://bilgekunduz.org>) olarak gerçekleştirilmektedir.

2.3. Öğrencilerin Algoritma Geliştirmeye Yönelik Öz Yeterlilik Algısı

Öz yeterlilik, bireyin belirli bir konuda kendi yeteneklerine ve becerilerine olan inancını ifade eder. Schunk (1995), öz yeterliliği bireylerin belirli bir görevi başarıyla tamamlayabileceklerine dair inançları olarak tanımlarken, Bandura'ya (1986) göre ise bireylerin eylemlerini, hedeflerini, düşünce süreçlerini ve duygusal tepkilerini belirler. Çalışmalara bakıldığında öz yeterlilik algısıyla başarı düzeyleri arasında güçlü bir ilişki olduğu görülmektedir (Pajares, 2002). Başarıyla tamamlanan görevler, olumlu geribildirim, uygun zorlukta görevler ve öğretmenler veya akranlar tarafından sağlanan destek, öğrencilerin algoritma geliştirmeye yönelik öz yeterlilik algılarını güçlendirebilir (Akçayır, 2016). Bu nedenle, algoritma geliştirme eğitiminde öz yeterlilik algısının güçlendirilmesi, öğrencinin bu alandaki başarısını artırmada önemli bir rol oynar (Korkmaz, Çakır ve Özden, 2017). Öğrencilerin başarılarına odaklanmak ve onları problem çözme sürecinde desteklemek de öz yeterlilik geliştirmeye katkı sağlayabilir. Algoritmik düşünme bağlamında, öz yeterlilik algısı bireyin kendi algoritmik düşünme becerisine olan inancını etkileyebilir (Gülbahar, Kert ve Kalelioğlu, 2019).

Algoritma geliştirmeye yönelik öz yeterlilik algısı, öğrencinin kendi algoritma oluşturma becerisine olan güvenini ifade eder. Öğrencinin bu alanda kendini ne kadar yetenekli hissettiği, onun bu alandaki performansını ve motivasyonunu doğrudan etkiler (Zimmerman, Bandura ve Martinez-Pons, 1992). Araştırmalar, öğrencilerin algoritmik düşünme becerilerinin gelişmesiyle birlikte öz yeterliliklerinin de arttığını göstermektedir. Algoritmik düşünme becerilerini geliştiren öğrenciler, problem çözme sürecinde daha fazla öz yeterlilik hissi yaşarlar (Korkmaz, Çakır ve Özden, 2017). Bu da onların daha motive olmalarını, daha iyi performans göstermelerini ve daha başarılı olmalarını sağlar. Öz yeterlilik, algoritmik düşünme becerilerinin kazanılmasında önemli bir faktör olarak değerlendirilmektedir. Bu nedenle, öğrencilerin algoritmik düşünme becerilerini geliştirmek için öz yeterliliklerini artırmaya yönelik stratejiler kullanmaları önemlidir (Demir ve Cevahir, 2020). Algoritma geliştirebilme yeteneği, özellikle bilgisayar bilimleri ve programlama alanında oldukça kritik bir öneme sahiptir. Dolayısıyla, algoritma geliştirmeye yönelik öz yeterlilik, öğrencinin bu kritik beceriyi geliştirebilme güvencesi olabilir.

Bireyin sahip olduđu becerilerle ilgili inançları, yalnızca davranışlarını değil aynı zamanda düşünme süreçlerini ve motivasyonunu da etkileyen bir süreçtir (Bandura, 1997). Bu süreç, öğrencilerin kendi yeteneklerine olan güvenlerini ve algoritma geliştirmeye yönelik öz yeterlilik inançlarını da içermektedir. Algoritma tasarım süreçlerinde programcının kendine dair algısı ve becerilerine olan inancı, programlama performansı üzerinde oldukça etkilidir (Boulay, 1986). Daha yüksek öz yeterliğe sahip bireyler, programlama zorluklarına özgüvenle, azimle ve öğrenme hevesiyle yaklaşırlar. Zorluklarla karşılaştıklarında daha dirençli olurlar ve çözüm arayışında aktif rol alırlar (Ramalingam, LaBelle ve Wiedenbeck, 2004).

2.4. Programlama, Bilgi İşlemsel Düşünme ve Algoritma Geliştirme İlişkisi

Programlama, bir sorunun çözüm sürecinde algoritma geliştirme ve uygulama süreçlerini içeren bir faaliyettir (Akçay ve Çoklar, 2016; Duncan, Bell ve Tanimoto, 2014). Bu süreçte, problemi analiz etmek, mantıksal düşünceyi kullanmak, adımları düzenlemek ve programlama dilleri aracılığıyla bu adımları bilgisayarın anlayabileceği bir şekilde ifade etmek gerekmektedir. Programlama becerisi, bu süreçleri etkin bir şekilde yürütebilmeyi ve algoritma geliştirme yeteneğini içerir. Programlama eğitimi ve uygulamaları, öğrencilerin bu becerilerini geliştirmede etkili bir araç olabilir (Lye ve Koh, 2014). Programlama aktiviteleri, öğrencilerin bilgi işlemsel düşünme becerileri ve algoritmik düşünme becerilerini kullanarak problem çözme stratejilerini geliştirmelerini sağlar, ayrıca öğrencilerin analitik düşünme yeteneklerini, mantıksal akıl yürütme becerilerini ve yaratıcılıklarını da destekler (Kalelioğlu, 2015).

Programlama, bilgi işlemsel düşünme ve algoritma geliştirme arasındaki ilişki oldukça güçlüdür (Grover ve Pea, 2013). Programlama yetkinliğinin önemli unsurları olan bu iki beceri, programlama becerisinin, algoritmik düşünmeyi teşvik ederek bilgi işlemsel düşünme becerilerini geliştirdiği şeklinde bir ilişki içindedir (Korkmaz, Çakır ve Özden, 2017). Programlama sürecinde algoritmik düşünme stratejileri kullanılırken, bilgi işlemsel düşünme becerileri de devreye girer ve problem çözme sürecinin etkili bir şekilde yönetilmesini sağlar (Guadalupe ve Fumarola, 2015).

Algoritmik düşünme, programlama sürecinin temel bir aşamasıdır ve bilgi işlemsel düşünme ile sıkı bir ilişkiye sahiptir. Algoritmik düşünme, problemi parçalara ayırma, adımları mantıklı bir sıra içinde düzenleme ve etkili bir çözüm bulma yeteneği olarak tanımlanabilir

(Papert, 1980). Bilgi işlemsel düşünme ise algoritma geliştirme sürecinin temelini oluşturan bir bilişsel süreçtir. Bu süreç, problemleri analiz etme, uygun stratejileri belirleme, adımları sıralama ve sonuçları değerlendirme becerilerini içerir. Bilgi işlemsel düşünme becerisi, öğrencilerin problem çözme yeteneklerini geliştirmelerine ve analitik düşünme yeteneklerini kullanmalarına yardımcı olur (Akçay ve Çoklar, 2016).

2.5. Programlama Öğretim Yaklaşımları

Programlama araçları tarih içerisinde bakıldığında zaman 1940' lı yıllarda Plankalkül (Konrad Zuse), ENIAC coding system, C – 10, 1950' li ve 1960' lı yıllarda BASIC (Beginner's All purpose Symbolic Instruction Code), ALGOL (ALGOritmic Language), COBOL (Common Business Oriented Language) vs., 1970 ve 1980' li yıllarda Pascal, C++, Eiffel, SQL gibi, 1990' lı yıllarda ise Python, Java, PHP, Delphi gibi metin tabanlı dillere sahip olan programlarla metinsel odaklı olarak ortaya çıkmıştır. Sonraları soyut bir yapıya sahip olan bu programlama dilinin öğretilmesi için arayışlar ve yeni geliştirilen sistemler devreye girmeye başlamıştır.

Programlama dillerinin yazım kurallarının karmaşıklığı, öğrenme sürecini zora sokan faktörler arasındadır. Ek olarak, kullanıcının etkileşim halinde olduğu arayüzün kullanıcı dostu olması, bir programlama dilini seçmede önemli bir etkindir (Akoğlu, 2014). Yaş ilerledikçe, basitten karmaşığa doğru söz dizilimlerini anlama ve kullanma becerilerinde bir gelişim gözlenmektedir. Fakat bazı çocuklarda, özellikle soyut işlemler dönemine geçmemiş olanlar için söz dizilimlerini anlamak zor olabilir (Turan ve Akoğlu, 2011). Ayrıca, programlama dillerinin anlaşılması güç yapısı, kod satırlarının yabancı dilde yazılması gibi etkenler, öğrencilerin yazılım geliştirmeye karşı içsel motivasyonunu düşürebilir. Bu süreçlerdeki öğrenme zorluklarını azaltarak öğrenme kayıplarının önüne geçebilmek için önerilen yöntemler arasında, blok tabanlı ve görsel programlama ortamlarının kullanılması bulunmaktadır (Cevahir ve Özdemir, 2017).

Geleneksel programlama dillerinde, belirli söz dizimi kuralları bulunmaktadır. Ancak görsel veya blok tabanlı uygulamalar, bu tür kurallara ihtiyaç duymadan kodlama yapma imkanı sağlar (Kim ve diğerleri, 2012). Görsel ve blok tabanlı kodlamanın avantajları sayesinde, küçük yaşlardan itibaren kodlama eğitimi daha kolay hale gelmiştir. Öğrenciler, metin

tabanlı programlama dillerinde karşılaştıkları zorlukları yaşamadan görsel ve blok tabanlı kodlama ile programlama konusunda olumlu bir tutum geliştirirler (Alp, 2019). Ayrıca, bu programlama ortamları, sözdizimi hatalarını azaltmaya, öğrencilerin bilişsel yükünü hafifletmeye ve bilgi işlemsel düşünme süreçlerine daha çok odaklanmalarına yardımcı olur (Feijóo – García ve diğerleri, 2021). Bu ortamlar aracılığıyla problem çözme süreci, öğrencilerin yaratıcılık, sorgulama, algoritmik düşünme ve bilişsel becerilerini geliştirmek için etkili bir yaklaşımdır (Çakıroğlu ve diğerleri, 2021). Bu nedenle, kodlama eğitimi sadece yazılım geliştirmek isteyen öğrenciler için değil, tüm öğrenciler için önemlidir, çünkü birçok açıdan fayda sağlamaktadır (Aytekin ve diğerleri, 2018).

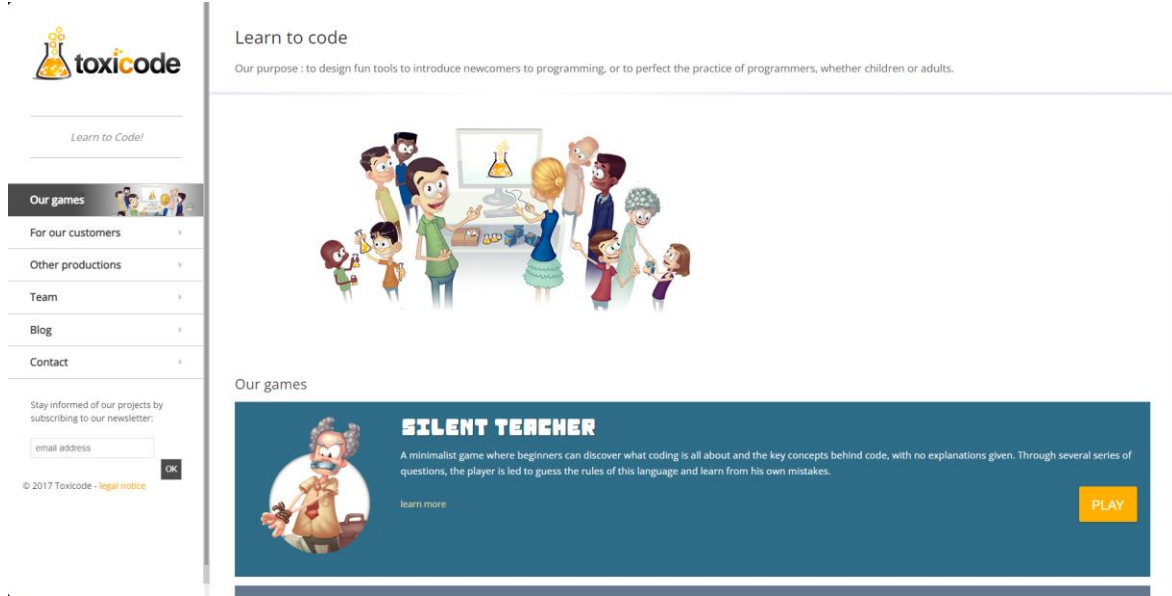
Erken yaşta anlaşılması ve öğretilmesi zor olan metinler yerine kolay olan simgeler, düğmeler, semboller ve bloklar koyarak sürükle – bırak mantığı ile görsel ve blok tabanlı programlama araçları (ortamları) geliştirilmeye başlanmıştır. Bireyin programlama öğretimine başlaması için geliştirilmiş olan görsel ve blok tabanlı programlama araçları, özellikle erken yaş çocuklar için tasarlanmış bir uygulama olarak düşünülebilir.

Blok tabanlı programlama araçları, erken yaşta öğrencilerin programlama öğrenmesinin ilk aşamasında karşılarına çıktığında önlerine konan, programlama etkinliklerini yaptıkları uygulama ortamlarıdır. Blok tabanlı programlama araçlarını erken yaşlarda çocukların kullanımına sunulması demek onlara eğlenceli bir ortam içerisinde, düşünme yapılarının algoritma mantığıyla çalışmasını, karşılaştığı problemlerin çözümünde en doğru yolu bulmalarını ve onlardan beklediğimiz bilgi işlemsel düşünme yetilerini erken kazanmaları anlamına gelmektedir (Alp, 2019). Programlama öğretimi erken yaşlarda eğitim öğretimin içerisine sokulduğu zaman, çocukların anlayabileceği, ilgilerini en üst seviyede tutabileceği, eğlenceli bir şekilde etkinliklerini yapıp ürünlerini görebileceği ortamların oluşmasını sağlayacaktır. Anaokulu ve ilkokul düzeyinde soyut ve anlaşılması güç olan kodların yerine sürükle – bırak şeklinde çalışabilen, hatalarının daha kolay bir şekilde düzeltilebildiği scratch, code.org, kodable, code monkey, kodu game lab gibi programlama araçlarının kullanım tercihi daha doğru olacaktır (Resnick ve diğerleri, 2009). Bu programlama araçları içerisinde Scratch, AppInventor ve Code.org örnekleri kullanılabilirliği artan, ara yüzünde, içeriğinde görsel açıdan zengin olmayı amaç edinen blok tabanlı programlama araçlarıdır (Taşdöndüren ve Korucu, 2022). Bireylerin kişisel oyunlarını yapmalarını, yeni projeler üretmesini, eğitsel içerikler hazırlamalarını sağlayan ve en çok bilinen programlama

ortamlarından bazıları şunlardır: Alice, Bitsbox, Codecombat, Microsoft Small Basic, Scratch, Lightbot, KoduGame Lab, Stagecast Creator, ToonTalk, App Inventor, CodeAcademy, KhanAcademy, Code.org vb.' dir (Korkmaz ve diğerleri, 2021). Yalnızca erken yaşlarda değil ileriki yaşlarda da ilk aşamada öğrenilmesi, uygulaması, kolay kullanım özellikleri ile blok tabanlı programlama öne çıkmaktadır. Alice, code.org, App Inventor gibi uygulamalar öğrenmenin ilk aşamasında ortaya çıkan zorlukları en aza indirgemektedir (Polat ve Hopcan, 2019). Erken yaşlarda blok tabanlı bir programlama seçme sebebi, ilerleyen süreçte karşılarına çıkacak olan metin tabanlı kodlamada karşılaşılan güçlükleri, soyut yapının anlaşılma güçlüğü en aza indirgeyerek ve o sürece kadar ilgiyi en üst düzeyde tutabilmektir. Bu uygulamaların hepsinin ortak bir amacı olduğu görülmektedir. Programlamayı en anlaşılır, en kolay ara yüzle, ilgi çekici bir halde “Nasıl öğretilbilir?” sorularının cevabını arayan uygulamalar olarak ortaya birçok programlama aracı çıkmaktadır. Bu ortamlar herhangi bir kod yazımına ihtiyaç duyulmadan kullanılabilen, animasyonlar yapılabilen, sürükle-bırak mantığını barındıran, hataları ön izleme ile anında gösterebilen uygulamalardır. Blok tabanlı programlama araçları, görselleştirilmiş blok kod yapısı, kullanıcı dostu arayüz ve çevrimiçi paylaşım imkanları gibi temel özellikleri ile geleneksel programlama dillerinden farklı bir yapıya sahiptir (Resnick ve diğerleri, 2009). Bu temel özelliklerinden dolayı programlama eğitiminde blok tabanlı uygulamaların kullanımının öğrencilerin motivasyonlarını artırdığı ifade edilmektedir (Erol ve Kurt, 2017). Toxicode platformu; özellikle yeni başlayan kullanıcıların, programlama ve algoritmik düşünme becerilerini geliştirme süreçlerinde karşılaştıkları soyut kavramları pratik yaparak daha hızlı somutlaştırmalarına yardımcı olmak için hazırlanmış, etkileşimli bir ortamdır (<https://compute-it.toxicode.fr/en/about.html>).

2.5.1. Toxicode uygulama platformu

Toxicode, kullanıcıların programlama ve algoritmik düşünme becerilerini geliştirmek için Toxicode SAS şirketi tarafından oluşturulmuş ve ücretsiz olarak kullanıma sunulmuş çevrimiçi platformdur. Bu platformda yer alan kullanım klavuzları ve dökümanlar Fransızca, uygulamaları açıklamaları İngilizce dil desteği vermesinin yanı sıra alt modüllerdeki kodlama içerikleri İngilizce ve Türkçe dil desteğine sahiptir (<http://www.toxicode.fr/learn>).



Şekil 2.2. Toxicode uygulama arayüzü.

2.5.1.1. Toxicode'nin öğrenme yaklaşımı

Blok tabanlı platformlarda kullanıcılar, belirli görevleri tamamlamak ve belirli problemleri çözmek için gereken kodları yazma ve test etme konusunda yönlendirilirler. Bu süreç, kullanıcıların kodlama kavramlarını ve dillerini pratik bir şekilde uygulayarak anlamalarına yardımcı olur (Resnick ve diğerleri, 2009). Kullanıcılar, belirli bir problemi çözmek için gereken adımları belirleme ve bu adımları etkili bir şekilde uygulama konusunda yardımcı olacak algoritmalar oluşturma ve uygulama konusunda teşvik edilirler. Bu, kullanıcıların problem çözme ve algoritmik düşünme becerilerini geliştirmelerine yardımcı olur (Grover ve Pea, 2013).

Şekil 2.2'de görülen bu platform, kullanıcıların karmaşık kodlama kavramlarını kolay ve anlaşılır bir şekilde öğrenmelerini sağlamak için basitten karmaşığa doğru olacak şekilde yapılandırılmış çeşitli etkileşimli ve görsel araçlar sunmaktadır. Bu nedenle etkinliklerin giriş bölümleri ilkökul düzeyine yönelikken, ilerleyen bölümler ortaokul düzeyindeki öğrencilere daha çok hitap etmektedir. Toxicode, programlamaya yeni başlayan öğrencilere kodlama mantığını öğretmek, kodları yorumlayabilmelerini sağlamak, sezgisel akıl yürütme becerilerini geliştirmek, algoritmik düşünmeyi ve problem çözme becerilerini geliştirmeye odaklanmış bir platformdur. Bu platform, kullanıcıların kendi kodlarını yazmalarını ve test

etmelerini saęlayan bir dizi farklı modül ve etkinlikler barındırmaktadır. Bu etkinlikler, kullanıcıların belirli bir problemi çözme konusunda farklı yaklaşımları nasıl deęerlendirebileceklerini, hangi yaklaşımın en etkili olacağını belirlemelerine yardımcı olur (Toxicode, 2023).

2.5.1.2. Toxicode' nin öğretim materyalleri ve araçları

Toxicode platformu, kullanıcıların programlama ve algoritmik düşünme becerilerini geliştirmek için çeşitli öğretim materyalleri ve araçları bulunmaktadır. Bunlar arasında çeşitli görsel programlama dilleri, etkileşimli öğrenme modülleri ve çeşitli kodlama etkinlikleri yer almaktadır.

Görsel programlama dilleri, kullanıcıların kodlama kavramlarını anlamalarını ve uygulamalarını kolaylaştırır. Bu diller, kullanıcıların kodları yazma ve test etme sürecini görselleştirmelerine ve anlamalarına yardımcı olur (Weintrop ve dięerleri, 2016).

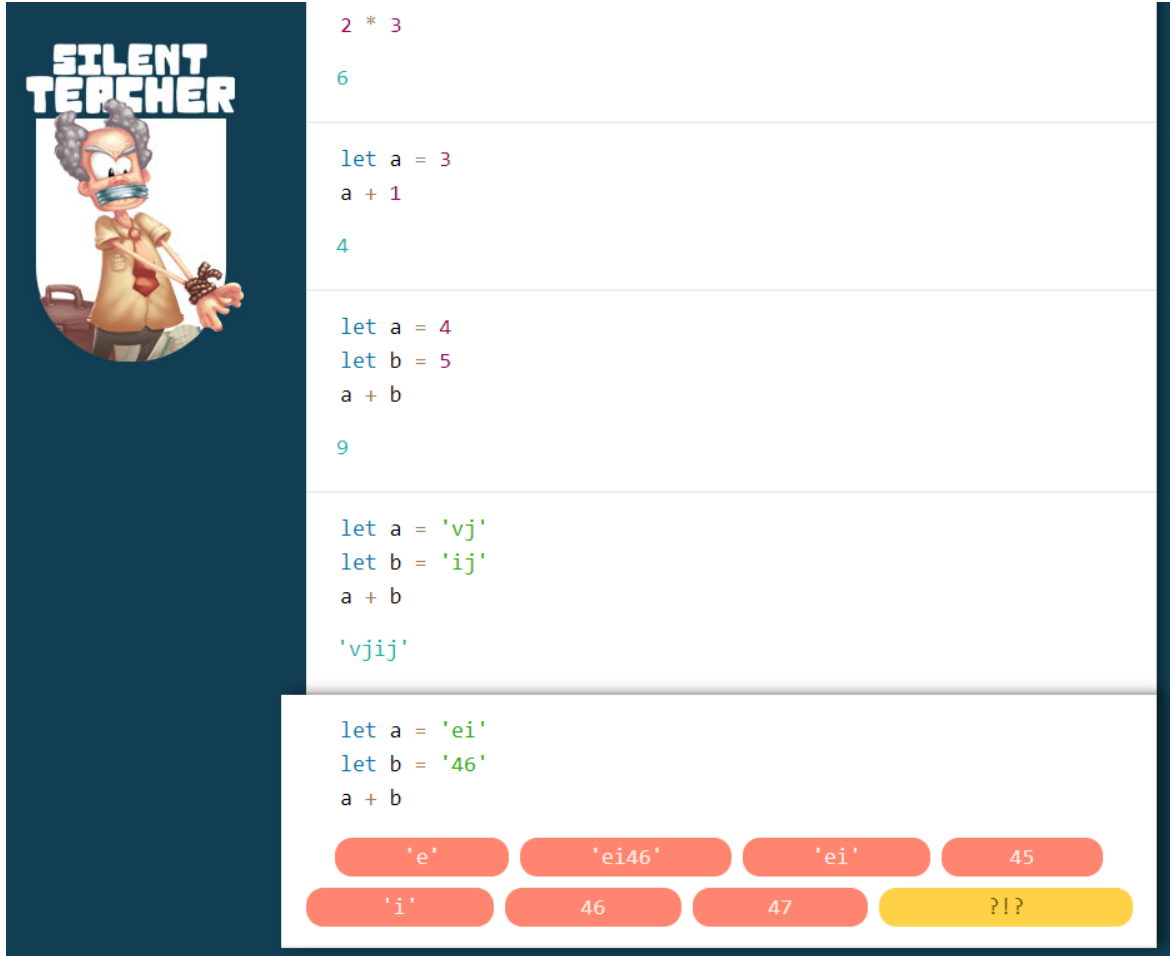
Etkileşimli öğrenme modülleri, kullanıcıların belirli kodlama kavramları ve dilleri hakkında daha fazla pratik yapma imkânı sağlamaktadır. Bu modüllerin her biri, kullanıcıların belirli soyut kavramları ve teknikleri anlama ve uygulama konusunda bir uzman rehberliğinde adım adım keşfettirerek sarmal bir yapıda ilerleyen ve kolaydan zora doğru aşamaları olan etkinlikleri barındırmaktadır (Şekil 2.3) (Lye ve Koh, 2014).



Şekil 2.3. Toxicode öğrenme modülleri.

2.5.1.2.1. Silent teacher (Sessiz öğretmen)

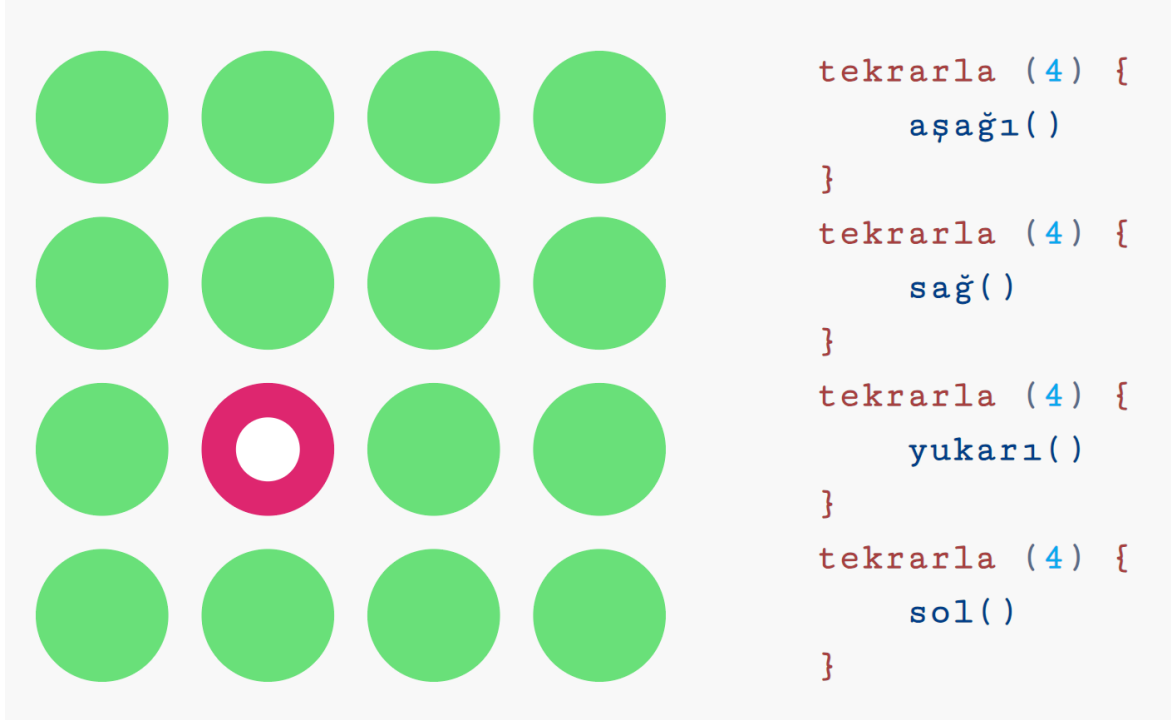
Türkçeye "Sessiz Öğretmen" olarak çevrilebilecek Silent Teacher alt modülünde, Python ve JavaScript kodlama dilleri metin tabanlı olarak öğretilmektedir. Modül, toplamda 14 etkinlikten oluşmakta ve üyelik gerektirmemektedir. Etkinlikler Türkçe veya İngilizce olarak yazılmış kodlarla gerçekleştirilebilmektedir. Kullanıcıların ilerleme bilgileri tarayıcı çerezlerinde saklanmakta, böylece sonradan oyuna tekrar girdiklerinde kaldıkları yerden devam edebilmektedirler. Etkinliklerin amacı, kodlama bilgisine sahip olmayan kullanıcıların değişkenler, döngüler, koşullu ifadeler, fonksiyonlar ve matematiksel işlemler gibi temel kavramların mantığını kavrama süreçlerinde öğretmen yardımı olmadan adım adım ilerleyerek deneme-yanılma yöntemiyle öğrenmelerini sağlamaktır.



Şekil 2.4. Silent Teacher etkinlik modülü

2.5.1.2.2. Compute it (Hesapla)

Compute It adıyla da bilinen ve Türkçe'ye "Hesapla" olarak çevrilebilen modül, algoritma öğretimi ve kod yorumlama becerisi kazandırmak amacıyla geliştirilen bir çevrimiçi uygulamadır. Bu modüldeki etkinlikler, klavye yön tuşları kullanılarak gerçekleştirilmekte ve üyelik gerektirmemektedir. Kullanıcıların ilerleme bilgileri tarayıcı çerezlerinde saklanmakta, böylece sonradan oyuna tekrar girdiklerinde kaldıkları yerden devam edebilmektedirler. Etkinliklerdeki talimatlar kodlar şeklinde sunulmakta ve kullanıcıların bu kodları doğru bir şekilde yorumlayarak istenilen adımları tamamlamaları beklenmektedir.

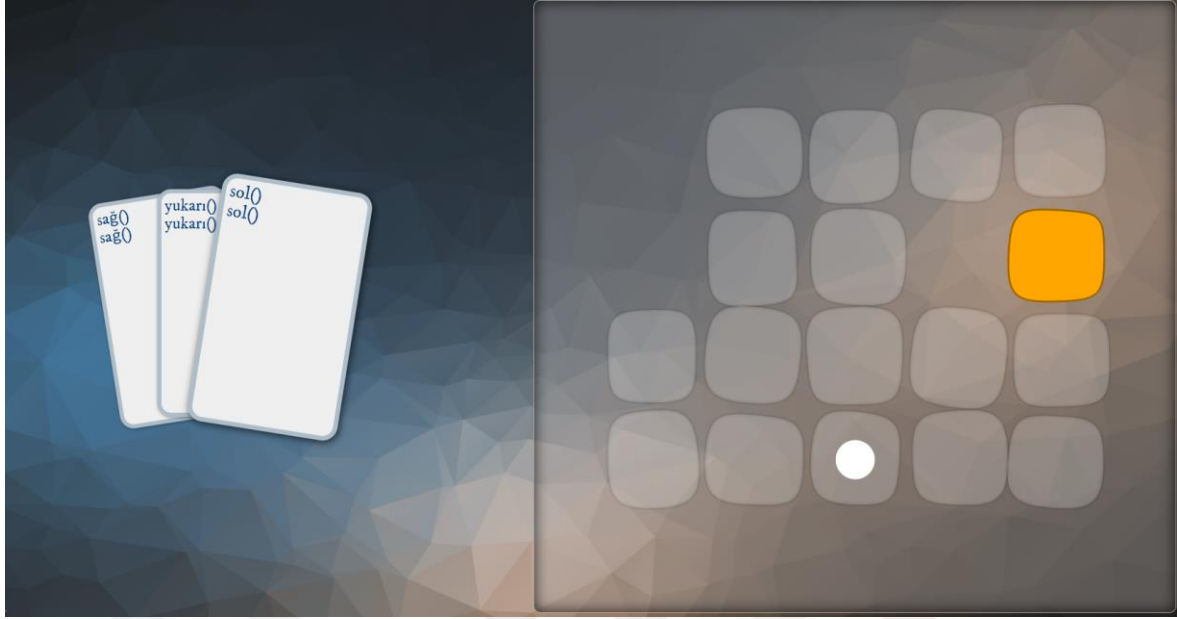


Şekil 2.5. Compute It etkinlik modülü

Modül, toplamda 58 etkinlikten oluşmakta ve etkinlikler Türkçe veya İngilizce olarak yazılmış kodlarla gerçekleştirilebilmektedir. Başlangıç bölümlerinde temel yönlendirme algoritmaları öğretilirken daha ileri bölümlerde temel programlama yapılarına odaklanılmaktadır. Bu etkinlikleri gerçekleştirirken de eğitim açıklamaları ve geri bildirimler olmadan kullanıcıların bağımsız bir şekilde kodlama becerisi kazanması hedeflenmektedir.

2.5.1.2.3. Little dot adventure (Küçük noktanın macerası)

"Küçük Noktanın Macerası" adıyla da bilinen uygulama, kullanıcıların algoritma ve kod yorumlama becerilerini geliştirmeyi amaçlamaktadır. Etkinliklerin her birinde, kullanıcılara kartlar üzerine yazılmış algoritmalar sunulmakta ve bu algoritmaları kullanarak klavyenin yön tuşlarını kullanarak küçük nokta'yı yönlendirmeleri istenmektedir.

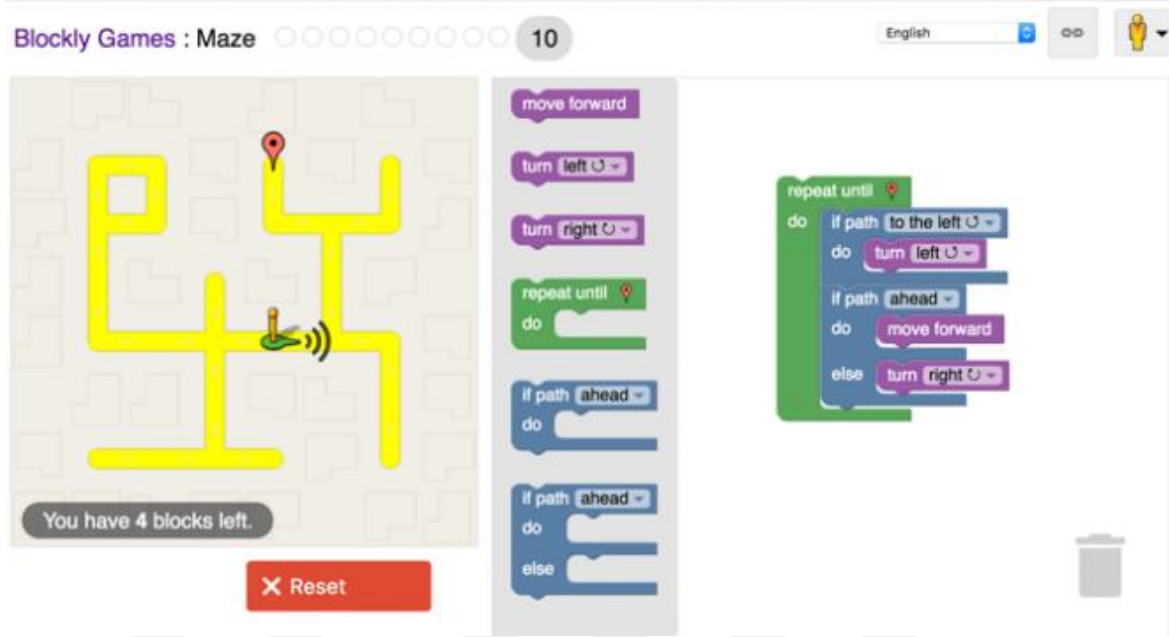


Şekil 2.6. Little Dot Adventure etkinlik modülü

Türkçe dil desteği de sunulan bu modülde, etkinlikleri gerçekleştirirken eğitim açıklamaları ve geri bildirimler olmadan kodlamaya yeni başlayan kullanıcıların bağımsız bir şekilde kodları anlama ve yorumlama yeteneği kazanması beklenmektedir. Temel yön hareketleriyle başlayan uygulamalar, daha karmaşık programlama yapıları olan döngüler, koşullar ve fonksiyonlar gibi konulara doğru ilerlemektedir. Etkinlikler gerçekleştirilirken hata yapıldığında uyarı sesi ve açıklama ile geri bildirim verilmekte ve aynı bölüm tekrar başlatılmaktadır.

2.5.2. Google Blockly uygulama platformu

Google Blockly, Google tarafından geliştirilmiş bir görsel programlama platformudur. Bu platform, kullanıcılarına algoritma mantığını ve programlama kavramlarını anlamalarını sağlayan bir araçtır. Kullanıcılar, Şekil 2.7’de görülen sürük ve bırak (drag – and – drop) bloklarını kullanarak Blockly uygulaması üzerinden programlama yapabilmektedir. Bu bloklar, temel programlama kavramlarını – döngüler, ifadeler, değişkenler ve daha fazlasını – görsel bir biçimde temsil eder (Blockly Games, 2021).



Şekil 2.7. Google Blockly uygulama arayüzü.

Blockly, çeşitli programlama dillerine (JavaScript, Python, PHP, Lua ve Dart gibi) kod çıktısı verebilir. Bu, öğrencilerin farklı dillerde kodlama becerilerini geliştirmelerine yardımcı olur (Kalelioğlu, 2015).

Google Blockly, hem eğitimcilerin hem de öğrencilerin bilgisayar bilimini ve programlama kavramlarını öğrenmeye ve öğretmeye yönelik etkili bir araç olarak kabul edilir. Bu nedenle, onun eğitim süreçlerinde daha fazla kullanılması ve bu alanda daha fazla araştırma yapılması gerektiği düşünülmektedir (Sünger ve diğerleri, 2022).

Öğrenciler, belirli bir problemi çözmek için gerekli blokları seçip birleştirerek, problem çözme, mantıksal düşünme ve algoritma oluşturma becerilerini kullanmaktadır. Bu süreç, onların bilgisayar biliminde ve özellikle programlamada gereken düşünce süreçlerini geliştirmelerine yardımcı olduğu ifade edilmektedir (Kalelioğlu ve Gülbahar, 2022). Bu platformun, öğrencilere karmaşık kodlama dillerini öğrenmeden önce temel programlama kavramlarına aşina olma fırsatı sunarak, öğrencilerin ileride daha karmaşık dilleri öğrenmelerini kolaylaştırdığı ve programlamaya olan öz yeterlilik algılarını artırdığı ifade edilmektedir (Fraser, 2015).

2.6. İlgili Çalışmalar

Oluk ve diğerleri (2018)'nin çalışması, Scratch programlama aracının 5. sınıf öğrencilerinin algoritma geliştirme ve bilgi işlemsel düşünme becerilerine etkisini incelemektedir. Bu çalışma, ön test-son test kontrol gruplu yarı deneysel bir tasarımla, 31'I deney 31'ikontrol grubunda olmak üzere toplam 62 öğrenci ile yürütülmüştür. Araştırma süreci 5. sınıf BTYD kapsamında 6 hafta sürmüştür. Bu süre zarfında deney grubundaki öğrencilere Scratch programı kullanılarak algoritma anlatılmış, kontrol grubundaki öğrencilere ise mevcut müfredat doğrultusunda algoritma öğretilmiştir. Uygulanan ön test ve son test sonuçlarına göre, öğrencilerin algoritma geliştirme becerileri ile bilgi işlemsel düşünme becerileri deney grubu lehine anlamlı derecede artış göstermiştir. Oluk ve diğerleri (2018)'nin çalışması, Scratch programının algoritma geliştirme ve bilgi işlemsel düşünme becerilerini geliştirmede etkili bir öğrenme aracı olduğunu göstermektedir.

Rodriguez-Martinez ve diğerleri (2019), 13-14 yaşlarındaki 120 ortaokul öğrencisi ile yaptığı bir çalışmada, programlamanın bilgi işlemsel düşünme becerileri gelişimine katkılarını incelemiştir. Çalışmada, öğrenciler 12 hafta boyunca Scratch programlama dilini kullanarak farklı seviyelerde zorluğa sahip olan 12 proje üzerinde çalışmışlardır. Çalışmanın sonunda, programlama öğretimine tabi tutulan öğrencilerin bilgi işlemsel düşünme becerileri, kontrol grubuna göre önemli ölçüde daha yüksek bulunmuştur. Bu bulgular, programlamanın bilgi işlemsel düşünme becerileri gelişimine etkili bir yöntem olduğunu göstermektedir.

Akgün ve diğerleri (2023), bilgisayar mühendisliği öğrencilerinin algoritma sorunlarını çözme süreçlerini göz izleme yöntemi ile incelemiş ve bu süreçte öğrencilerin algoritma dersine olan ilgilerini ve algılarını da değerlendirmiştir. Bu araştırmada, bir üniversitenin bilgisayar mühendisliği programından 12 öğrenci gönüllü olarak katılmıştır. Öğrencilere algoritma konusunda metinsel ve akış diyagramı içeren iki soru sorulmuştur. Sonuçlar, öğrencilerin metinsel soruyu %25 doğrulukla, akış diyagramı içeren soruyu ise %83,33 doğrulukla cevapladığını göstermektedir. Ayrıca, öğrencilerin algoritma dersine olan ilgileri yüksek bulunmuş ve algoritma sorularını uygulamalı olarak çözmeyi tercih ettikleri belirlenmiştir.

Ma ve diğerleri (2021) lise öğrencilerine 14 hafta boyunca scratch eğitimi vermiş ve eğitim sonunda lise öğrencilerinin bilgi işlemsel düşünme becerilerini önemli ölçüde geliştirdiğini; öğrencilerin bilgi işlemsel düşünme öz yeterliliği üzerinde, özellikle de eleştirel düşünme, algoritmik düşünme ve problem çözme üzerinde önemli bir olumlu etkiye sahip olduğu sonucuna ulaşmıştır.

Atmatzidou ve Demetriadis (2016) çalışmalarında, eğitsel robotik öğrenme etkinliği bağlamında öğrencilerin bilgi işlemsel düşünme becerilerinin gelişimini araştırmıştır. Araştırmada, cinsiyetlere ve iki farklı yaş grubuna (15 ve 18 yaş) göre öğrencilerin bilgi işlemsel düşünme becerileri gelişimini operasyonel hale getirmek ve araştırmak için uygun bir bilgi işlemsel düşünme modeli kullanılmıştır. Farklı eğitim seviyelerindeki 164 öğrenci (Ortaokul: 89; Meslek Yüksek Okulu: 75) eğitsel robotik öğrenme faaliyetlerine katılmış (haftada 2 saat, toplam 11 hafta) ve bilgi işlemsel düşünme becerileri, aktivite sırasında farklı aşamalarda farklı yöntemler (yazılı) kullanılarak değerlendirilmiştir. Sonuçlar şunu göstermektedir: öğrenciler sonunda yaş ve cinsiyet değişkenlerinden bağımsız olarak eşit seviyede bilgi işlemsel düşünme becerileri gelişimine ulaşırlar, bu becerilerin çoğu durumda tam olarak gelişmesi için zamana ihtiyaç vardır ve öğrencilerin puanları aktivitenin sonuna doğru önemli ölçüde artar.

del Olmo-Muno ve diğerleri (2020) çalışmalarında bilgisayarsız etkinliklerin ilköğretimin ilk yıllarındaki öğrencilerde bilgi işlemsel düşünmenin gelişimine katkı sağlayıp sağlamadığını değerlendirmektedir. Bu amaçla, hem bilgisayarsız hem de bilgisayarlı aktiviteleri birleştiren karma bir yaklaşımın nihai faydasını araştırmak için yarı deneysel bir çalışma yürütülmüştür. Deneye 84 ikinci sınıf öğrencisi katılmıştır. Etkinlikler Code.org'dan alınan bir dizi faaliyet üzerine tasarlanarak uygulanmıştır. Bu etkinlikler iki aşamadan oluşmaktadır. İlk aşamada bir grup bilgisayarsız aktivitelerle, diğer grup ise bilgisayarlı aktivitelerle çalışmış ve ikinci aşamada ise her iki grup da bilgisayarlı aktivitelerle çalışmıştır. Bilgi işlemsel düşünme becerilerinin gelişimi, katılımcıların önerilen öğretime yönelik motivasyonları ve önceki iki alanda öğrencilerin cinsiyetinin etkisi öğretim öncesinde, arasında ve sonrasında yapılan testler aracılığıyla analiz edildiğinde, bilgi işlemsel düşünme, motivasyon ve cinsiyet dikkate alındığında, eğitime bilgisayarsız etkinliklerin dahil edilmesinin faydalı görüldüğü sonucuna varılmıştır.

Eryılmaz ve Deniz (2021) çalışmalarında, Tinkercad'ın programlama eğitiminde öğrencilerin bilgi işlemsel düşünme becerileri ve algıları üzerindeki etkisini değerlendirmektedir. Araştırmanın örneklem grubu, 2019-2020 eğitim öğretim yılında 5. 6. 7. ve 8. sınıf düzeyinde okuyan 583 ortaokul öğrencisinden ve 11 bilişim teknolojileri öğretmeninden oluşmaktadır. Araştırmalarında, nicel ve nitel araştırma desenlerini bir arada kullanarak karma yöntemle gerçekleştirmişlerdir. Araştırmanın sonuçlarına göre, öğrencilerin teknoloji yetkinliklerinin başlangıç seviyesinde olduğu belirlenmiş ve %70'den fazlasının çevrimiçi kurslar almadığı tespit edilmiştir. Cinsiyet faktörüne bağlı olarak öğrencilerin Tinkercad motivasyonu, kullanışlılık ve kullanım kolaylığı algıları arasında anlamlı bir farklılık bulunmamıştır. Ancak, sınıf düzeyi faktörüne göre motivasyon ve kullanışlılık algılarında anlamlı bir farklılık tespit edilmiştir. Öğrencilerin bilgi işlemsel düşünme algılarının orta düzeyde olduğu belirlenmiştir. Cinsiyete bağlı olarak bilgi işlemsel düşünme becerileri arasında anlamlı bir farklılık bulunmazken, 5. sınıf düzeyi ile 7. sınıf düzeyi arasında bilgi işlemsel düşünme becerilerinde anlamlı bir farklılık saptanmıştır. Öğrencilerin Tinkercad kullanım sıklığı ile Tinkercad algıları arasında pozitif yönlü düşük bir ilişki bulunurken, Tinkercad algıları ile bilgi işlemsel düşünme becerileri arasında pozitif yönlü orta düzey bir ilişki olduğu sonucuna varılmıştır.

Deng ve diğerleri (2020), çalışmalarında Pencil Code (blok tabanlı) uygulamasının k-12 eğitim düzeyinde, öğrencilerin bilgi işlemsel düşünme becerilerini geliştirmenin önemli bir parçası olarak ifade etmektedirler. Bu çalışmada, aynı öğretim içeriği ile farklı programlama araçlarının öğrencilerin bilgi işlemsel düşünme becerileri ve bilgisayar öğrenme tutumları üzerindeki etkilerini sınıf ortamında test etmeyi amaçlamıştır. Deneysel uygulama sürecinde 99 katılımcı, 8 haftalık Visual Basic (metin tabanlı) ve Pencil Code (blok tabanlı) programlama öğretimine tabi tutulmuştur. Çalışma sonucunda blok tabanlı Pencil Code programlama ortamında öğrenen öğrencilerin bilgi işlemsel düşünme becerilerini önemli ölçüde geliştirdiği, aynı zamanda daha özgüvenli hale geldikleri ifade edilmiştir. Ayrıca, 8 haftalık uygulama sürecinden sonra yapılan görüşmelerde, öğrencilerin Pencil Code ile öğrenirken daha fazla iletişim ve etkileşim davranışı sergilediklerini belirtmişlerdir.

3. YÖNTEM

Bu bölümde, öncelikle araştırma modelini, daha sonra deney desenini, çalışmanın hedef kitlesini ve örneklemini oluşturan çalışma grubunu, deneysel uygulama sürecini, veri toplama araçlarını, toplanan verilerin nasıl yorumlandığını ve hangi istatistiksel yöntemlerin kullanıldığının anlatıldığı veri analizlerine yer verilmektedir.

3.1. Araştırma Modeli

Bu çalışmada, Toxicode ve Google Blockly uygulamalarının öğrencilerin bilgi işlemsel düşünme becerisine ve algoritma geliştirmeye ilişkin öz yeterlik algısına olan etkisini incelemek amacıyla deneysel desenlerden Solomon dört gruplu model kullanılmıştır. Bu model kontrol grupları ve deney grupları arasında seçkisiz olarak belirlenen yarı deneysel bir yapıya dayanmaktadır. Bu çalışmada, Toxicode ve Google Blockly programlama öğretim platformları bağımsız değişken, bilgi işlemsel düşünme becerisi ve algoritma geliştirmeye ilişkin öz yeterlik algıları ise bağımlı değişkenler olarak değerlendirilmiştir.

3.2. Solomon Deneysel Modeli (Dört Gruplu)

Solomon desen modelinde, olası ön-test etkileri giderilmeye çalışılır. Bu modelde iki gruba ön test uygulanırken diğer iki gruba ise ön test uygulanmaz. Dört gruptan ön test uygulanan ile ön test uygulanmayan birer grup deney grubu olarak tanımlanır. Süreç sonunda tüm gruplara son test uygulanır. Deney_1 ve Kontrol_1 grupları, ön test-son test kontrol grubu modelini temsil ederken, Deney_2 ve Kontrol_2 grupları ise son test kontrol grubu modelini temsil etmektedir (Fraenkel ve Wallen, 2008).

Solomon Dört Grup Deseni, hem iç hem de dış geçerliliği birlikte koruyan en sağlam deney modelidir (Karasar, 2006). Bu deneysel model, araştırmacıya iç ve dış geçerliği tehdit eden faktörleri kontrol etme imkanı sağlar. Özellikle iç geçerliliği etkileyebilecek ölçüm, zamanlama ve olgunluk etkilerini kontrol etmek ve sonuçların genelleştirilebilirliğini artırmak amacıyla kullanılır (Kirk, 2009). Bu model "ön test duyarlılığı" olarak adlandırılan ön testin son test puanlarındaki değişime olan etkisini test etme imkanı sağlar. Ayrıca, bu model, ön test-son test kontrol grubu ve son test kontrol grubu olan iki farklı deney grubunun

kullanılması sayesinde olası ölçüm etkilerini kontrol edebilir ve zamanlama ile olgunluk etkilerini azaltabilir. İç ve dış geçerliği artırarak, araştırmacılara daha güvenilir ve geçerli sonuçlar elde etme imkanı tanır. Ayrıca, uygulanabilirliği ve pratikliği sayesinde deneysel grupları oluşturma ve veri toplama sürecini daha esnek bir şekilde planlama olanağı sunar (Campbell ve Stanley, 1963). Solomon dört grup deneysel modele ait uygulama modeli çizelge 3.1’de sunulmuştur.

Çizelge 3.1. Solomon deneysel uygulama modeli (dört gruplu) (Braver ve Braver, 1988)

Grup		Ön test	İşlem	Son test
Deney_1	S	Test1	Deneysel İşlem	Test3
Kontrol_1	S	Test2	-	Test4
Deney_2	S	-	Deneysel İşlem	Test5
Kontrol_2	S	-	-	Test6

Çizelge 3.1'deki Deney_1 ve Kontrol_1 grupları, ön test-son test kontrol grubu desenini yansıtırken, Deney_2 ve Kontrol_2 grupları ise son test kontrol grubu desenini temsil etmektedir. Deney_1 ve Kontrol_1 gruplarına ön test uygulanırken, ön test yapılmayan Deney_2 grubu aracılığıyla da ön test-deney etkileşiminin kontrolü sağlanabilmektedir. Seçkisiz (S) atama ile oluşturulan Solomon Modeli'nde, toplam altı ölçüm yapılmaktadır: İki ön test ve dört son test ölçümü.

Seçkisiz atama, deneklerin iç geçerliği tehdit eden unsurları kontrol etmek için kullanılabilir. Bu nedenle, bilimsel çalışmalarda seçkisiz atama etkili ve güçlü bir yöntem olarak tercih edilmektedir (Fraenkel, Wallen ve Hyun, 2012). Bu deneysel çalışmada gruplar oluşturulurken, sınıflar seçkisiz atama ile oluşturulmuştur.

Çalışmanın bağımlı değişkenleri öğrencilerin “Bilgisayarca Düşünme Beceri Düzeyleri (BDÖ)” ve “Algoritma Geliştirmeye İlişkin Öz Yeterlik Algı (AGYÖ)” ölçeklerini kullanarak elde edilen verilerdir. Bağımsız değişken ise Toxicode deneysel işlemidir.

Solomon deneysel modele (dört gruplu) göre yapılan çalışmanın simgeleştirilmiş hali çizelge 3.2’de sunulmuştur.

Çizelge 3.2. Araştırmada kullanılan Solomon deneysel desen (dört gruplu)

Grup		Ön test	Deneysel İşlem	Son test
Deney_1	S	BDÖ+AGYÖ	Toxicode	BDÖ+AGYÖ
Kontrol_1	S	BDÖ+AGYÖ	-	BDÖ+AGYÖ
Deney_2	S	-	Toxicode	BDÖ+AGYÖ
Kontrol_2	S	-	-	BDÖ+AGYÖ

AGYÖ: Algoritma Geliştirmeye İlişkin Öz Yeterlik Algısı Ölçeği; BDÖ: Bilgisayarca Düşünme Beceri Düzeyleri Ölçeği

Çizelge 3.2’de ifade edildiği gibi çalışmada deney grupları ve kontrol grupları olmak üzere ikişer adet grup bulunmaktadır. Deney_1 ve Deney_2 gruplarında, etkinliklerin uygulama aşamasında Toxicode platformu kullanılırken, kontrol gruplarında BTYD 5. sınıf kitabında kullanılması önerilen Google Blockly platformu uygulama etkinlikleri tercih edilmiştir (MEB, 2018). Deney_1 ve Kontrol_1 gruplarına deneysel uygulama öncesinde BDÖ ve AYGÖ ön test olarak uygulanmıştır. Son test aşamasında, aynı testler tüm gruplara tekrar uygulanmıştır.

3.3. Çalışma Grubu

Araştırmadaki çalışma grubu, 2022-2023 eğitim-öğretim yılında Samsun ilindeki bir devlet okulunda 5. sınıfta öğrenim gören toplam 102 öğrenciden oluşmaktadır. Uygulama aşamasında kullanılan çalışma gruplarının dağılımı çizelge 3.3’te verilmiştir.

Çizelge 3.3. Çalışma grubuna ait dağılım

Grup	Sınıflar	Uygulanan Yöntem	Kız n	Erkek n	Toplam n	%
Deney_1	5-D	Toxicode	13	13	26	25,5
Kontrol_1	5-F	Blockly	15	12	27	26,5
Deney_2	5-M	Toxicode	10	18	28	27,5
Kontrol_2	5-B	Blockly	10	11	21	20,6
		Toplam	48	54	102	100

Çizelge 3.3 incelendiğinde seçkisiz atama yöntemi ile belirlenen gruplardaki öğrenci dağılımlarının birbirlerine yakın değerlerde olduğu görülmektedir. Çalışmadaki uygulamalar ve etkinlikler araştırmacı ve okul bilişim teknolojileri öğretmeni ile birlikte yürütülmüştür. Deneysel uygulama sürecinde tüm gruplara halihazırda derslerine giren öğretmende herhangi bir değişiklik olmamıştır. Ayrıca uygulama sürecinde gözlemci etkisinden (Roethlisberger ve Dickson, 1939) kaçınmak için öğrencilerin, birilerinin onları gözlemlediğini hissettirmemek için azami özen gösterilmiştir.

3.4. Uygulama Süreci

Araştırmanın yürütüleceği ortaokulun 5. sınıf öğrencileriyle bilişim teknolojileri dersinde uygulama yapabilmek için gerekli olan etik kurul ve MEB uygulama izinleri deneysel süreçler öncesinden alınmıştır (Ek16). Bu araştırma dört farklı 5. sınıf ve bir bilişim teknolojileri öğretmeni ile birlikte 06 - 31 Mart 2023 tarihleri arasında dört haftalık süreç içerisinde yürütülmüştür. Deneysel uygulama sürecinde devamsızlık yaparak sürece aktif olarak katılmayan öğrencilerin verileri veri setinden çıkarılarak analizde dikkate alınmamıştır.

3.5. Deneysel Uygulama

Deney grubundaki öğrencilere yönelik, algoritma geliştirmeye ilişkin öz yeterlik algısı ve bilgi işlemsel düşünme becerilerinin BTYD kapsamındaki kazanımlara uygun uzman (iki farklı BTYD öğretmeni) görüşleri ve dönütleri dikkate alınarak Toxicode uygulama

etkinlikleri hazırlanmıştır. Kontrol grubunda ise var olan öğretim programında yer alan Google Blockly uygulamasına ait ders planları hazırlanmış ve uygulanmıştır. İki gruba da bu kazanımlar için uygun çalışma planları ve etkinlikler hazırlanmıştır (Ek1, Ek2, Ek3...Ek13). Uygulama süreci 4 hafta sürmüştür.

3.6. Veri Toplanması

Verilerin toplama sürecinde Korkmaz, Çakır ve Özden'in (2015) geliştirmiş olduğu BDÖ ile Bakırcı ve Akgün'ün (2019) geliştirmiş olduğu AGYÖ ön test-son test olacak şekilde 2022-2023 bahar döneminde uygulanmıştır. Çalışmada kullanmak üzere yazarlardan ölçek kullanımı için gerekli izinler alınmıştır. Ayrıca bir devlet üniversitesinden etik kurul izni ve MEB'den uygulama izni de alınmıştır.

3.6.1. Bilgisayarca düşünme beceri düzeyleri ölçeği (BDÖ)

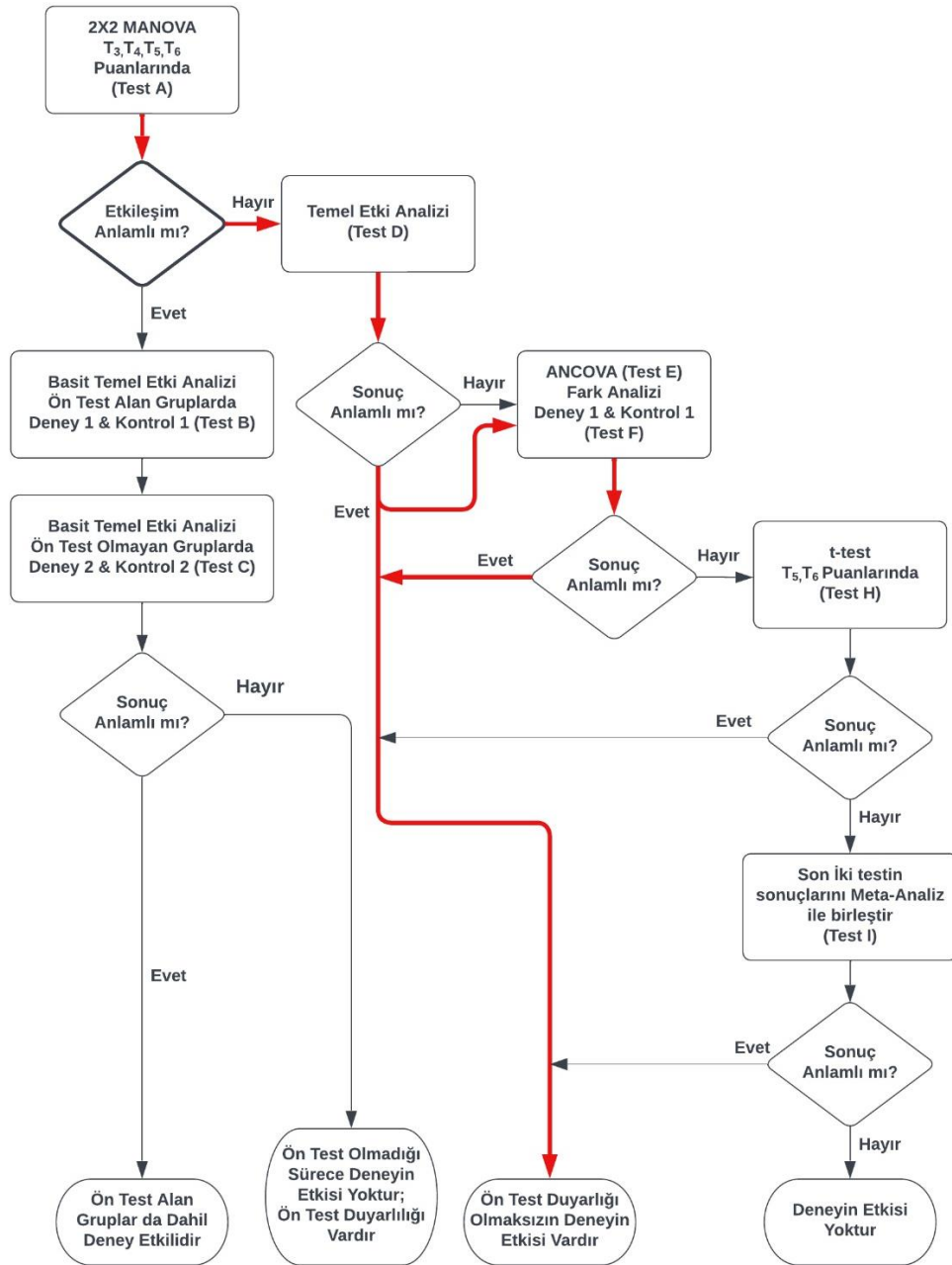
BDÖ, ortaokul düzeyindeki öğrencilerin bilgi işlemsel düşünme beceri seviyelerini anlamak amacıyla kullanılmıştır. Bu ölçek yaratıcı düşünme, eleştirel düşünme, algoritmik düşünme, işbirlikli düşünme ve problem çözme olmak üzere 5 faktöre dayanan 22 maddelik likert tipi (5'li) ölçektir. Ölçek geliştirilirken yazarlar tarafından yapılan faktör analizleri sonucunda maddelerin korelasyon katsayı değer aralığının 0.6555 ile 0.862, regresyon değer aralığının ise 0.507 ile 0.872 değiştiği; Cronbach Alpha güvenirlik katsayısının da 0.809 olduğu belirtilmiştir (Korkmaz, Çakır ve Özden, 2015).

3.6.2. Algoritma geliştirmeye ilişkin öz yeterlik algı ölçeği (AGYÖ)

AGYÖ, Bakırcı ve Akgün (2019) tarafından ortaokul düzeyindeki öğrencilerin algoritma geliştirmeye ilişkin öz yeterlik algılarını değerlendirmek amacıyla geliştirilmiştir. 16 maddelik ölçek çalışmasında yapılan değişken (faktör) analizi sonucunda toplam varyansın %91,929'unu açıklayan tek faktörlü bir yapının olduğu görülmüştür. İç tutarlık belirlemek için kullanılan Cronbach's Alpha analiz testi sonucunda $\alpha=0,994$ olarak bulunmuştur. Madde analizleri ve anahtarlanmış cevaplara yönelik yapılan değerlendirmeler sonucunda testin geçerliği bakımından yeterli olarak kabul edilmiştir (Bakırcı ve Akgün, 2019).

3.7. Verilerin Analizi

Deneysel sürecin sonunda, deney ve kontrol grupları arası yapılan karşılaştırmalarda Braver ve Braver'ın (1988) önerdiği meta-analitik yaklaşımından faydalınılmıştır. Bu yaklaşımın en önemli tarafı, deneysel sürecin etkilerini, ön test alan grupların sonuçları ile son test alan grupların sonuçlarını değerlendirmesini mümkün kılmasıdır. Bu modele özgü veri analizi akış diyagramı Şekil 3.1'de sunulmaktadır.



Şekil 3.1. Solomon meta-analitik basamakları (Braver and Braver, 1988).

Braver and Braver (1988) ön test duyarlılığını test etmek amacıyla önerdiği akış diyagramında ANOVA (2x2 varyans analizi) ile başlatılması gerekirken, ilgili diyagram bu çalışma için uyarlandığında ve solomon dört grup deneysel modeli dikkate alındığında, Şekil 8'deki gösterimde belirtilen adımlar arasında yer alan analiz süreci, çalışmada kullanılan iki farklı ölçme aracı olan BDÖ ve AGYÖ'den elde edilen veriler analiz sürecine dahil edilmesiyle; bağımlı değişkenlerin aynı anda analiz edilebilmesini sağlayan çok değişkenli varyans analiz yöntemi olan MANOVA (Test A) ile devam edilmesine karar verilmiştir. MANOVA analizi, gruplara ait son test puanlarının, deneysel süreç öncesinde yapılan ön testten etkilenip etkilenmediğini belirlemek amacı ile kullanılmıştır. Sonrasında istatistiksel olarak anlamlı bir fark olmadığı için yöntemin etkisine bakmak amacıyla “Temel Etki Analizi (Test D)” yapılmıştır. Bu analiz, deney ve kontrol gruplarının deneysel işlem üzerindeki etkilerini değerlendirirken, gruplara ön test uygulanıp uygulanmadığına bakılmaksızın kıyaslamalar yapılmasını sağlar. Bu kıyaslama sonucuna göre yöntemin istatistiksel olarak anlamlı bir etkisi olduğu ortaya çıkmıştır. Ancak temel etki analizi, ön test uygulanan grupların ön test değerlerini değerlendirmeye almadığı için, sonuç istatistiksel açıdan anlamlı olsa bile, Braver ve Braver'ın (1988) analizlere devam etme önerisi nedeniyle, Şekil 8'deki bir sonraki analiz basamağına (Test E) geçilmiştir. Bu adımda (Test E) ön test uygulanan gruplarda ön test verilerini kontrol ederek deneyin son test puanları üzerindeki etkisi ortaya konulmaya çalışılmaktadır (Büyüköztürk, 2014, s.121). Bu amaçla, Deney_1 ve Kontrol_1 grupları arasındaki farkı ortaya çıkarmak için kovaryans analiz yöntemi olan ANCOVA yapılmıştır. Bu analiz sonucunda istatistiksel olarak anlamlı sonuca ulaşıldığı için gruplar arasındaki değişimi görebilmek için iki bağımlı değişkenin öntest ve son test puan farklarına bağımsız örneklem t-testi uygulanmıştır. Sonraki adımda fark analizi sonucuna göre anlamlı farklılığın görüldüğü AGYÖ verilerine göre Deney_1 ve Kontrol_1 gruplarına tekrarlayan ölçümler için ANOVA analizi yapılmıştır.

Veri analizleri öncesi araştırmanın bağımlı değişkenleri olan BİD ve AGYÖ testlerine ait ön test - son test verilerinin normal dağılım olup olmadığını belirlemek için, çarpıklık (skewness) ve basıklık (kurtosis) değerleri, Kolmogorov-Smirnov normallik testi, histogram, kutu-bıyık grafikleri incelenmiştir. Kolmogorov-Smirnov testinde anlamlılık düzeyi ,05 seviyesinin üzerinde olan veri setlerinin normal dağılım sergilediği kabul edilmektedir (Özdamar, 2013). Çarpıklık ve basıklık değerlerinin her ikisi de $\pm 1,5$ değerleri arasında olması verilerin kabul edilebilir aralık içerisinde yer aldığını ve normal dağıldığını

göstermektedir (Tabachnick ve Fidell, 2013). Buna göre araştırmamızda ön test uygulanan Deney_1 ve Kontrol_1 gruplarının BİD ve AGYÖ ölçeklerine ait ön test verilerinin Kolmogorov-Smirnov testi için anlamlılık düzeyi olarak kabul edilen ,05'in üzerinde olduğu görülmüştür. Aynı veri setine ait çarpıklık ve basıklık değerlerinin ise -1,018 ile -,017 aralığında değiştiği belirlenmiştir. Her iki inceleme sonucuna göre, BİD ve AGYÖ ölçeklerinin ön test puanlarına ait verilerin normal dağılım gösterdiği sonucuna ulaşılmıştır. Tüm deney ve kontrol gruplarının BİD ve AGYÖ son test verilerinin analiz sonuçları Kolmogorov-Smirnov testi anlamlılık düzeyi olan ,05'in üzerindedir. Yine aynı veri setine ait çarpıklık-basıklık değerleri -,367 ile ,149 aralığında değiştiği görülmüştür. Kabul edilebilir değerler aralığında yer alan bu sonuçlara göre BİD ve AGYÖ ait son test verilerinin analiz sonuçlarında verilerin normal dağılım sergilediği sonucuna ulaşılmıştır.

4. BULGULAR

Araştırmanın bu bölümünde varsayımları test etmek üzere, deneysel uygulama sonrası toplanan tüm verilerin analizleri neticesinde ortaya çıkan bulgular ve yorumlar çizelgeler halinde sunulmuştur.

Grupların ön test - son test verilerine ait tanımlayıcı istatistiklere ait değerler çizelge 4.1’de verilmiştir.

Çizelge 4.1. BDÖ ve AYGÖ tanımlayıcı sonuçları

Test	Grup	n	$\bar{X}$	S
AGYÖ	Ön	Deney_1	26	2,94
		Kontrol_1	27	3,10
	Son	Deney_1	26	3,40
		Kontrol_1	27	3,80
		Deney_2	28	3,44
		Kontrol_2	21	3,63
BDÖ	Ön	Deney_1	26	3,10
		Kontrol_1	27	3,32
	Son	Deney_1	26	3,49
		Kontrol_1	27	3,86
		Deney_2	28	3,47
		Kontrol_2	21	3,61

Çizelge 4.1’e göre, AGYÖ ön test ortalaması en yüksek grup Kontrol_1 ($\bar{X}$ =3,10) iken son test ortalaması en yüksek grup yine Kontrol_1 ($\bar{X}$ =3,80)’dir. BDÖ ön test ortalaması en yüksek grup Kontrol_1 ($\bar{X}$ =3,32) iken son test ortalaması en yüksek grup yine Kontrol_1 ($\bar{X}$ =3,86)’dir.

Ön test uygulanan Deney_1 ve Kontrol_1 gruplarının deneysel uygulama öncesi denliğini görebilmek için sınıfların karşılaştırılmasına yönelik yapılan ilişkisiz örneklem t-testi sonuçları çizelge 4.2’de verilmiştir.

Çizelge 4.2. Ön test uygulanan sınıfların karşılaştırılmasına ilişkin ilişkisiz örneklem t-testi sonuçları

Ön Test	Grup	Sınıf	n	$\bar{X}$	S	t	sd	p
AGYÖ	Deney_1	5-D	26	2,94	0,29	-1,84	51	0,07
	Kontrol_1	5-F	27	3,10	0,34			
BDÖ	Deney_1	5-D	26	3,10	0,42	-1,73	51	0,08
	Kontrol_1	5-F	27	3,31	0,46			

Çizelge 4.2 incelendiğinde, BDÖ Deney_1 grubu ortalama puanı ($\bar{X} = 3,10$) ile Kontrol_1 grubu ortalama puanı ($\bar{X} = 3,31$) arasında istatistiksel olarak anlamlı bir farklılık olmadığı ($t(99) = -1,73, p > .05$). AGYÖ Deney_1 grubu ortalama puanı ($\bar{X} = 2,94$) ile Kontrol_1 grubu ortalama puanı ($\bar{X} = 3,10$) arasında istatistiksel olarak anlamlı bir farklılık olmadığı görülmüştür ($t(99) = -1,84, p > .05$). Bu doğrultuda, Deney_1 ve Kontrol_1'in başlangıçta birbirine benzer özelliklere sahip oldukları söylenebilir.

Grupların denkliliğinin tespit edilmesinden sonra, analizin ilk basamağında ön test duyarlılığının var olup olmadığı test edilmeye çalışılmıştır. Bu amaçla, tüm grupların son test puanlarına yönelik bir MANOVA (2x2) analizi gerçekleştirilmiş ve tüm grupların son test puanlarının ön teste bağlı olarak değişip değişmediği incelenmiştir. Bu analize ait desen, aşağıdaki çizelge 4.3'te gösterilmiştir.

Çizelge 4.3. MANOVA (2X2) analiz deseni

Ön Test	Deneysel İşlem	
	Var	Yok
Var	D_1	K_1
Yok	D_2	K_2

Ön şartların sağlanması sonucu yapılan MANOVA (2X2) analiz sonuçları Çizelge 4.4'de yer almaktadır.

Çizelge 4.4. Ön Test Bağımsız Değişkenine Göre BDÖ ve AYGÖ son test puanlarında MANOVA (2x2) sonuçları

Varyans Kaynağı	Wilks'λ	F	df1	df2	p	ηp ²
Ön Test (Var/Yok)	,984	1,08	3	225	,441	,016

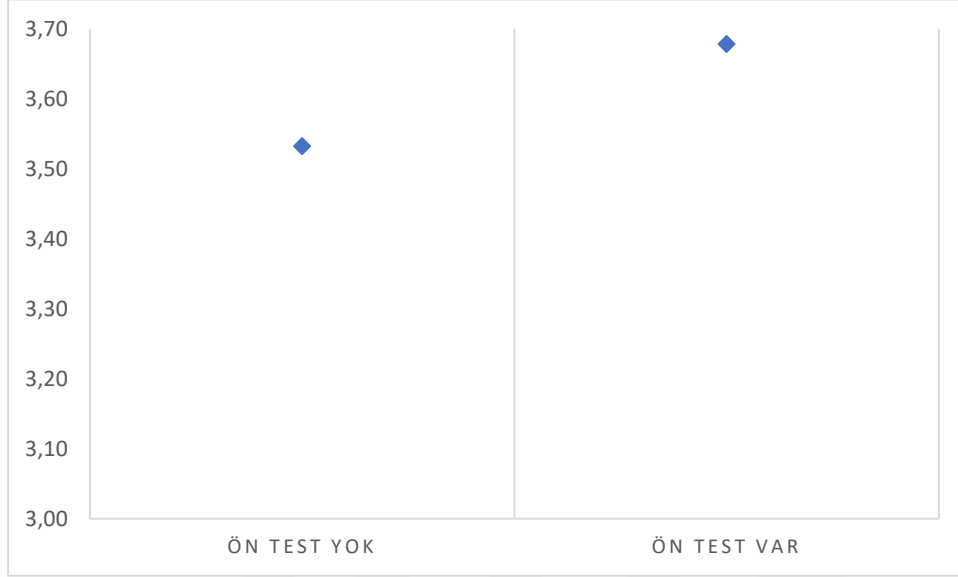
Çizelge 4.4’de son test puanlarından oluşturulan bağımlı birleşik değişkene göre gruplar arasında istatistiksel olarak anlamlı düzeyde fark görülmemiştir ($F(3, 225) = 1,08$; $p > .05$; Wilks’λ = ,984; $\eta p^2 = ,016$). Bu bulgu ile, BDÖ ve AYGÖ son test puanlarından elde edilen grup ortalamalarının ön test alma durumuna bağlı olarak değişiklik göstermediği görülmüştür. Dolayısıyla, bu deneysel çalışmada yapılan analizlere dayanarak, ön test duyarlılığının mevcut olmadığı sonucuna ulaşılmıştır. Grupların ön test alıp almadıklarına bağlı olarak AGYÖ ve BDÖ son test puanlarını gösteren grafikler, Şekil 4.1 ve Şekil 4.2’de gösterilmiştir.



Şekil 4.1. Ön test alma durumlarına göre grupların AYGÖ son test puanları

Şekil 4.1’de AGYÖ ait son test puanlarının ön_test bağımsız değişkene göre nasıl değiştiği gösterilmektedir. Deney grubunda ön test uygulanma durumuna bağlı olarak AGYÖ son test puanlarında bir fark oluşurken ($\bar{X}_{\text{ÖnTestVar}} = 3,60$; $\bar{X}_{\text{ÖnTestYok}} = 3,52$; $p > ,05$); yapılan

MANOVA (2X2) analizi sonuçlarına göre bu farkın istatistiksel olarak anlamlı olmadığı görülmüştür.



Şekil 4.2. Ön test alma durumlarına göre grupların BDÖ son test puanları

Şekil 4.2’de BDÖ son test puanlarının ön_test bağımsız değişkene göre nasıl değiştiği gösterilmektedir. Deney grubunda ön test uygulanma durumuna bağlı olarak BDÖ son test puanlarında bir fark oluşurken ($\bar{X}_{\text{ÖnTestVar}} = 3,68$; $\bar{X}_{\text{ÖnTestYok}} = 3,53$; $p > ,05$); yapılan MANOVA (2X2) analizi sonuçlarına göre bu farkın istatistiksel olarak anlamlı düzeyde olmadığı görülmüştür.

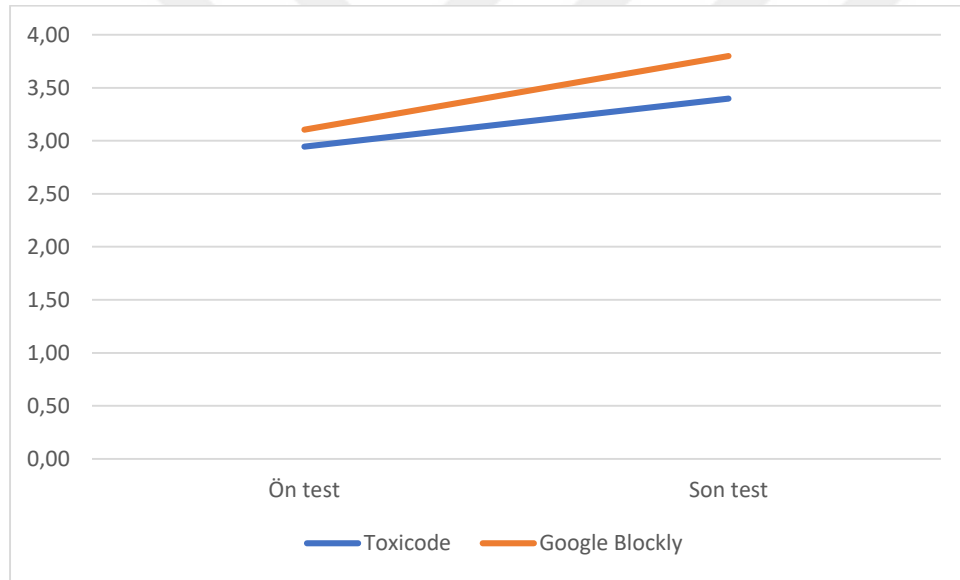
Çizelge 4.5’de grupların son test puanlarında yapılan MANOVA (2X2) analizi yöntem bağımsız değişkenine ait veriler bulunmaktadır. Bu çizelgede grupların son test puanlarının yöntemle bağılı olarak değişip değişmediği incelenmektedir. Ön şartların sağlanması ile yapılan MANOVA (2X2) analizine ait değerler çizelge 4.5’de gösterilmektedir.

Çizelge 4.5. Yöntem Bağımsız Değişkenine Göre BDÖ ve AYGÖ son test puanlarında MANOVA (2X2) sonuçları

Varyans Kaynağı	Wilks’ λ	F	df1	df2	p	ηp^2
Yöntem	0,922	1,12	3	329	,018	,078

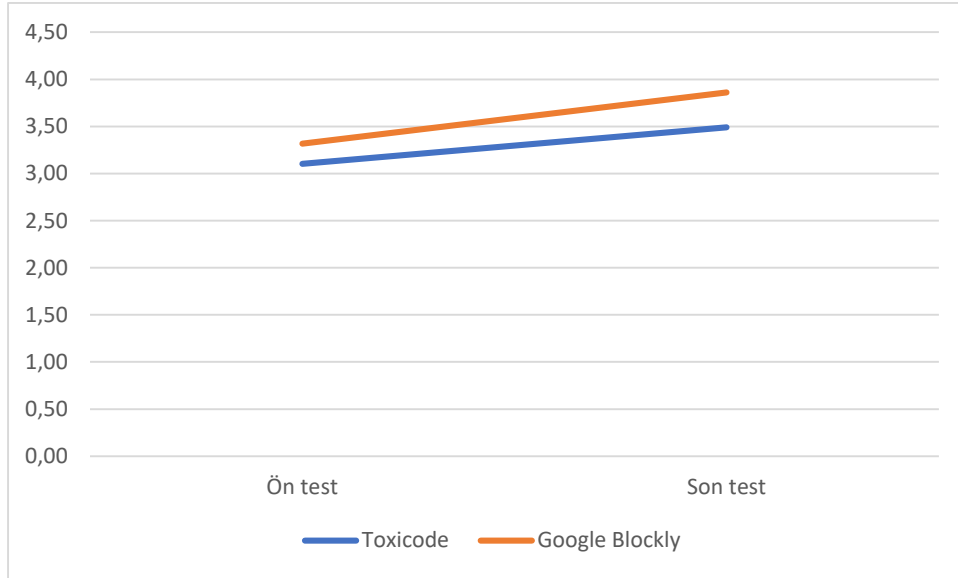
Çizelge 4.5 incelendiğinde, son test puanları üzerinden oluşturulan bağımlı birleşik değişkene göre gruplar arasında istatistiksel olarak anlamlı düzeyde bir farklılık görülmüştür ($F(3, 329) = 1,12$; $p < .05$; Wilks' $\lambda = ,922$; $\eta^2 = ,078$). Bu sonuç BDÖ ve AYGÖ son test puanlarından oluşturulan grup ortalama puanlarının, yönteme bağlı olarak değiştiğini göstermektedir. Bu analizler sonucunda hipoteze ait olan her iki denencenin de Toxicode online platformundaki uygulamalar kullanılarak yürütülen bilişim teknolojileri dersinde, öğrencilerin bilgi işlemsel düşünme becerileri ve algoritma geliştirmeye ilişkin öz yeterlik algısının anlamlı düzeyde arttığı bulunmuştur.

Grupların, uygulanan yönteme göre AGYÖ ve BDÖ son test puanlarındaki değişimi ifade eden grafikler, Şekil 4.3 ve Şekil 4.4.'de gösterilmiştir.



Şekil 4.3. Uygulama yöntemine göre grupların AYGÖ son test puanlarındaki değişim

Şekil 4.3'de AGYÖ son test puanlarının yöntem bağımsız değişkenine bağlı olarak değişimi gösterilmektedir. Deney grubunda uygulanan yönteme göre AGYÖ son test puanlarında bir fark oluşurken ($\bar{X}_{\text{Toxicode}} = 3,42$; $\bar{X}_{\text{Google_Blockly}} = 3,73$; $p < ,05$); yapılan MANOVA analizi sonuçlarına göre bu farkın istatistiksel olarak anlamlı olduğu bulunmuştur.



Şekil 4.4. Uygulama yöntemine göre grupların BDÖ son test puanlarındaki değişim

Şekil 4.4’de BDÖ son test puanlarının yöntem bağımsız değişkene bağlı olarak değişimi gösterilmektedir. Deney gruplarında uygulanan yönteme göre BDÖ son test puanlarında bir fark oluşurken ($\bar{X}_{\text{Toxicode}} = 3,48$; $\bar{X}_{\text{Google_Blockly}} = 3,75$; $p > ,05$); yapılan MANOVA analizi sonuçlarına göre bu farkın istatistiksel olarak anlamlı olduğu tespit edilmiştir.

Meta-anatilik yaklaşımın bir sonraki adımına bağlı olarak, deneyin etkisini belirlemek için ön test puanlarının kontrol edilmesi ve tüm grupların son test ortalama puan farkları alınarak oluşturulan puanlar arasında ANCOVA analizi (Test E-Fark Analizi) yapılmıştır. Gruplar arası ön test puanlarına dayalı olarak düzeltilmiş son test puanları çizelge 4.6’da sunulmuştur

Çizelge 4.6. BDÖ ve AGYÖ için grupların düzeltilmiş son test puanları

Gruplar	AGYÖ		BDÖ		Düzeltilmiş		Düzeltilmiş	
	Son Test		Son Test		AGYÖ Son Test		BDÖ Son Test	
	Ortalaması		Ortalaması		Ortalaması		Ortalaması	
	$\bar{X}$	SD	$\bar{X}$	SD	$\bar{X}$	SD	$\bar{X}$	SD
Toxicode	3,41	0,58	3,48	0,57	3,42	0,77	3,48	0,77
Google Blockly	3,72	0,52	3,75	0,054	3,72	0,82	3,74	0,81

Çizelge 4.6’da yer alan AGYÖ düzeltilmiş son test puanlarına göre Google Blockly grubu ortalama puanının ($\bar{X}_{\text{Google_Blockly}} = 3,72$) Toxicode grubu ortalama puanından ($\bar{X}_{\text{Toxicode}} = 3,42$) daha büyük olduğu görülmektedir. Çizelge 4.6’da yer alan BDÖ düzeltilmiş son test puanlarına göre Google Blockly grubu ortalama puanının ($\bar{X}_{\text{Google_Blockly}} = 3,74$) Toxicode grubu ortalama puanından ($\bar{X}_{\text{Toxicode}} = 3,48$) daha büyük olduğu görülmektedir. Bu farkların istatistiksel olarak anlamlılığını incelemek için çizelge 4.7’de bulunan ANCOVA (kovaryans analizi) sonuçlarından faydalanılmıştır.

Tüm grupların ön-son test puanları arasında ANCOVA analizi (Test E) sonuçları çizelge 4.7’de gösterilmektedir.

Çizelge 4.7. BDÖ ve AGYÖ için ön test-son test puanlarının kontrol edildiği ANCOVA (Test E) analizi

Varyansın Kaynağı		KT	Sd	KO	F	p	η^2
Yöntem	AGYÖ	2,321	1	2,321	7,291	,008	,069
	BDÖ	1,704	1	1,704	5,402	,022	,052

Çizelge 4.7’de yer alan ANCOVA analizi sonucuna göre, AGYÖ yönetime göre ön test son test puanları incelendiğinde, gruplar arasında istatistiksel olarak anlamlı bir fark olduğu görülmüştür ($F(1, 99) = 7,291$, $p < 0.05$; $\eta^2 = 0.069$). Bu farkın, düzeltilmiş ortalama puanlarına göre kontrol grubu lehine olduğu ($\bar{X}_{\text{Google_Blockly}} = 3,72$; $\bar{X}_{\text{Toxicode}} = 3,42$) görülmektedir. BDÖ ön test son test puanları kontrol edildiğinde, gruplar arasında istatistiksel olarak anlamlı düzeyde fark olduğu görülmüştür ($F(1, 99) = 5,402$, $p < 0.05$; $\eta^2 = 0.052$). Bu farkın, düzeltilmiş ortalama puanlarına göre kontrol grubu lehine olduğu ($\bar{X}_{\text{Google_Blockly}} = 3,74$; $\bar{X}_{\text{Toxicode}} = 3,48$) görülmektedir.

Deney_1 ve Kontrol_1 gruplarının BDÖ ve AGYÖ ön test ile son test puanları arasındaki fark puan analizi (Test F) için yapılan bağımsız örneklem t testi sonuçları, çizelge 4.8’de gösterilmiştir.

Çizelge 4.8. Deney_1 ve Kontrol_1 Grupları Bağımsız Örneklem t-Testi Fark Puan Analizi Sonuçları

Test	Uygulama	Sınıf	n	$\bar{X}$	S	t	sd	p
AGYÖ	Deney_1	5-D	26	0,45	0,35	-2,14	51	0,03
	Kontrol_1	5-F	27	0,69	0,46			
BDÖ	Deney_1	5-D	26	0,39	0,25	-1,56	51	0,12
	Kontrol_1	5-F	27	0,54	0,45			

Çizelge 4.8’de görüldüğü gibi; BDÖ’de kontrol grubunun fark puan ortalaması yüksek olmasına rağmen gruplar arasında istatistiki olarak anlamlı düzeyde bir farklılık görülmemektedir ($t(51)=-1,56;p>0.05$). AGYÖ’de ise kontrol ve deney grubu arasındaki fark kontrol grubu lehine istatistiksel açıdan bir anlam ifade etmektedir ($t(51)=-2,14;p<0.05$).

5. SONUÇ VE ÖNERİLER

Araştırmanın bu bölümünde “Toxicode ve google blockly uygulamalarının öğrencilerin bilgi işlemsel düşünme becerisine ve algoritma geliştirmeye ilişkin öz yeterlik algısına etkisi?” sorusuna ait alt araştırma sorularına yönelik elde edilen bulgular ışığında ortaya çıkan sonuçlara yer verilmiştir.

5.1. Sonuçlar

Blok tabanlı kodlama eğitiminde kullanılan online platform olan Toxicode uygulamasının 5. sınıf öğrencilerinin algoritma geliştirmeye ilişkin öz yeterlik algılarına ve bilgi işlemsel düşünme becerilerine olan etkisini belirlemek için Solomon dört gruplu deneysel modeli ile yapılan bu çalışmada kazandırılması amaçlanan bu becerileri ölçmek için AGYÖ ve BDÖ ön test - son test olarak uygulanmıştır. Deneysel süreçlerin sonunda elde edilen veriler Solomon deneysel model (dört gruplu) analizleri için, Braver ve Braver'ın (1988) önerdiği meta-analitik yaklaşımı kullanılarak değerlendirilmiştir.

Varsayım sonucunda yansız atama yöntemi ile belirlenen sınıflardan Kontrol_1 ve Deney_1 gruplarına ön test uygulanmış, test sonrası bağımsız t-testi analizi yapılarak gruplar arasında istatistiksel açıdan anlamlı düzeyde bir fark olmadığı görülmüş ve sonrasında deneysel uygulama sürecinde devamsızlık yaparak sürece aktif olarak katılmayan öğrencilerin verileri deneysel uygulamanın gerçek etkisinin belirlenmesi basamağında analizde dikkate alınmamıştır.

Meta-analitik yaklaşımın ilk basamağında ön test etkisini görebilmek için yapılan MANOVA (2x2) ön test etkisi analiz (Test A) sonucunda deneysel uygulamanın ön test duyarlılığı olmaksızın etkisinin olduğu sonucuna varılmıştır. Bu sonuca dayalı olarak Toxicode uygulamasının etkisini görebilmek için MANOVA (2x2) temel etki analizi (Test D) yapılmış ve BDÖ ve AYGÖ son test puanları bileşeninden elde edilen grup ortalama puanlarının, yöntem etkisi ile anlamlı olarak farklılaştığı görülmüştür. Ancak “Temel Etki Analizi”, ön test alan grupların ön test bilgilerini dikkate almadığından dolayı, analizin sonucu istatistiksel olarak anlamlı çıksa bile analizlere devam edilmesi (Braver ve Braver, 1988) önerildiği için bir sonraki analiz adımına ANCOVA (Test E) geçilmiş ve analiz

sonuçlarına göre öğrencilerin algoritma geliştirmeye ilişkin öz yeterlik algısı ve bilgi işlemsel düşünme becerilerini anlamlı düzeyde arttırdığı ancak bu farkın kontrol grubu lehine daha fazla olduğu tespit edilmiştir. Bir sonraki adımda bağımsız örneklem t-testi ile yapılan fark puan analizi (Test F) sonucuna göre ise öğrencilerin algoritma geliştirmeye ilişkin öz yeterlik algısı ve bilgi işlemsel düşünme becerilerinin ortalamaları artarken bu artışın istatistiksel olarak sadece algoritma geliştirmeye ilişkin öz yeterlik algısında kontrol grubu lehine anlamlı bir fark olduğu sonucuna ulaşılmıştır. Buna göre Toxicode ve Google Blockly uygulamalarının öğrencilerin bilgi işlemsel düşünme becerilerini anlamlı düzeyde artırdığı söylenebilir. Totan (2021) da çalışmasında benzer şekilde blocky ile blok tabanlı kodlama eğitimi sonrasında bilgi işlemsel düşünme becerisine yönelik öz yeterlik algılarının arttığı sonucuna ulaşmıştır. Benzer şekilde Ceylan (2020) senaryo uygulamasına dayalı Scratch öğretiminin bilgi işlemsel düşünme becerileri ile problem çözme becerilerine etkisine ilişkin yaptığı çalışmada öğrencilerin akademik başarı ve kalıcılık puanlarına bu eğitimin olumlu etkisi olduğunu göstermiştir. Atiker (2019) çalışmasında Scratch uygulaması üzerinden yapılan kodlama öğretiminin bilgi işlemsel düşünme becerilerine olumlu yönde etkisi olduğunu gözlemlemiştir. Ünsal (2020) ise diğer bir benzer platform olan code.org'u kullanarak blok tabanlı kodlama uygulamalarının ilkökul 2. sınıf düzeyindeki öğrencilerin bilgi işlemsel düşünme becerilerine ve etkinlik algısına olumlu etkisi olduğunu belirtmiştir. Ulucan (2022) da okul öncesi öğrencileriyle yaptığı çalışmada uygulanan kodlama ve robotik etkinliklerinin bilgi işlemsel düşünme ve problem çözme becerisine anlamlı bir etkisi olduğu bulmuştur. Oluk ve diğerleri (2018) ile Yünkül ve diğerleri (2017) de çalışmalarında benzer sonuçlar elde etmişlerdir. Bu sonuçlar bizim araştırmamız ile benzerlik göstermektedir.

Öğrencilerin eğitim süreçleri içerisinde edindiği bilgilerin ve yaptıkları uygulamaların onların öz yeterlilik becerilerini arttırdığını vurgulandığı araştırmalarda ortaya konmuştur (Toytok ve Çetin, 2015). Bilgin (2022) bilgisayarsız kodlamanın etkilerini araştırdığı çalışmasında öğrencilerin BİD becerilerinde ve öz yeterlik algılarında istatistiksel olarak anlamlı bir artış olduğu sonucuna ulaşmıştır. Bir diğer çalışmada, fiziksel programlama öğretim araçlarından biri olan mBlock ile yapılan etkinliklerin öğrencilerin algoritma geliştirmeye yönelik öz-yeterlik algılarını ve tutumlarını önemli ölçüde arttırdığı sonucuna ulaşılmıştır (Gunbatar ve Karalar, 2018).

Ulusal literatürde belirtilen bulguların yanı sıra uluslararası literatürde de benzer sonuçlara ulaşılmıştır. Örneğin, Papert (1980) yaptığı çalışmalarda, kodlama ve bilgi işlemsel düşünmenin çocukların matematik ve problem çözme becerilerini geliştirdiğini vurgulamıştır. Resnick ve arkadaşları (2009), Scratch kodlama platformunun çocukların yaratıcı düşünme yeteneklerini artırdığını belirtmiştir. Grover ve Pea (2013) ise bilgi işlemsel düşünmenin sadece kodlama becerisi olmadığını, aynı zamanda problem çözme, sistematik düşünme ve analitik düşünme yeteneklerini de içerdiğini vurgulamışlardır. Bu uluslararası çalışmalar, ulusal literatürde elde edilen bulguların evrenselliğini teyit eden niteliktedir. Ancak her iki literatürde de belirtilen önemli nokta, kodlama eğitiminin etkilerinin kültürel ve eğitim sistemine bağlı olarak değişiklik gösterebileceğidir.

Genç ve Karakuş (2011), Vatansever (2018) ve Bakırcı (2019) çalışmalarında Scratch ile yapılan programlama öğretimi etkinliklerinin öğrencilerin algoritma geliştirmeye ilişkin öz yeterlik algı düzeylerini artırdığı sonucuna ulaşmışlardır. Ozoran ve diğerleri (2012) de üniversite öğrencileriyle yaptığı çalışmada benzer sonuçlara ulaşmıştır.

Alan yazında, farklı öğrenme ortamlarının öğrencilerin Bilgi İşlemsel Düşünme (BİD) becerileri üzerindeki etkisine dair birçok araştırma bulunmaktadır. Bu araştırmalarda, bazı öğrenme ortamlarının diğerlerine göre daha etkili olduğu görülmüştür. Örneğin, Blockly ile öğrenim gören öğrencilerin daha yüksek ortalamalara sahip oldukları tespit edilmiştir (Ertürk ve Öztürk, 2019). Bununla birlikte, bu farklılığın nedenleri üzerine literatürde çeşitli yorumlar ve çıkarımlar bulunmaktadır (Brown, 2017).

Alan yazında yapılan taramalar sonucunda araştırmayı destekler nitelikte birçok bulgunun bulunmasının yanı sıra Atman Uslu ve diğerleri (2018), Scratch uygulama etkinliklerinin öğrencilerin BİD becerilerini etkilemediği sonucuna varmışlardır. Benzer bir çalışmada Alagöz'ün (2022) daha önce kodlama dersi almanın kodlama dersi almama durumuna göre BİD becerilerinde anlamlı derecede bir farklılık göstermediği sonucuna ulaştığı bir çalışmaya da rastlanmıştır.

Alan yazı incelemelerinde; blok tabanlı kodlama uygulamalarının soyut düşünmeyi kolaylaştırdığı, hataları azalttığı, algoritma ve mantık becerilerini geliştirdiği, yaratıcılığı teşvik ettiği, işbirliği ve paylaşımı desteklediği, ayrıca başlangıç için düşük eşik sağlayarak

daha fazla kişinin programlamaya ilgi duymasını sağladığı bilgilerine ulaşılmıştır (Pinto ve Escudeiro, 2014; Resnick ve diğerleri, 2009). Ayrıca algoritma ve kodlama eğitiminde önerilen Scratch uygulaması ile yapılabilecek yazılım, oyun ve hikâyelerin metin, grafik, ses, video gibi öğelerle desteklenebilmesi, anlaşılması güç olan soyut kavramların somutlaştırılarak daha anlaşılabilir hale gelmesini sağlayabileceğine dair bulgulara ulaşılmıştır. Elde edilen bulgulara yönelik olarak, derslerde metin, grafik, ses, video gibi çoklu ortam öğelerin kullanımına yönelik yapılan birçok alan yazı çalışmasında, öğrencilerin daha başarılı olduğu ifade edilmiştir (Klee ve Bayrhuber, 2002; Yaman, 2005).

Bu çalışma her ne kadar nicel bir model kullanılarak desenlense de, deneysel süreçte bilişim teknolojileri dersi öğretmeninin gözlemlerinden hareketle, öğrencilerin Toxicode uygulamasının arayüzünün sade, anlaşılır, kolay ve keyifli olduğu, tekrar etme, geri dönüt alma ve kendi öğrenme hızlarına uygun yapıda bulunduğu söylenebilir. Blockly platformunda kod bloklarını kullanarak görev tamamlamak için analiz ve sentez düzeyinde beceriler gerekirken Toxicode platformunda ise verilen ifadelere göre adım adım ilerleyerek sonuca ulaşması için daha alt düzey becerileri gerektirmektedir. Blok tabanlı programlama araçları kullanılarak farklı disiplin alanlarında da öğrencilerin derslere yönelik tutumlarını ve ders başarılarını olumlu yönde artırılabilirdiği alanyazın çalışmalarına da yansımıştır (Başaran, 2018; Ke, 2014; Secer, 2020; Ünal, 2019).

5.2. Öneriler

Araştırmanın bu bölümünde “Toxicode ve google blockly uygulamalarının öğrencilerin bilgi işlemsel düşünme becerisine ve algoritma geliştirmeye ilişkin öz yeterlik algısına etkisi?” sorusuna ait alt araştırma sorularına yönelik elde edilen bulgular ışığında gelecekteki araştırmalara yönelik önerilere yer verilmiştir.

5.2.1. Araştırmacılar için öneriler

- Ülkenin farklı kültürel ve sosyoekonomik düzeydeki ailelerin çocuklarıyla birlikte çalışarak genişletilmiş örneklem gruplarıyla çalışmalar yapılabilir.

- Alan yazına paralel olarak Toxicode uygulamaları ile yaratıcılık, iş birliği, sosyal gelişim, bilişsel gelişim başta olmak üzere diğer gelişim alanlarının ilişkisini inceleyen yeni araştırmaların yapılabilir.
- Fiziksel programlama araçları ile farklı programlama yaklaşımlarının öğrencilerin bilgi işlemsel düşünme becerisi ve algoritma geliştirmeye ilişkin öz yeterlilik algısı üzerindeki etkilerini anlamaya yönelik çalışmalar yapılabilir.
- Fiziksel programlama araçlarının yanı sıra diğer blok tabanlı ve metin tabanlı programlama araçlarıyla da çarpaz karşılaştırmalar yapılabilir. Bu şekilde, farklı programlama yaklaşımlarının öğrencilerin bilgi işlemsel düşünme becerisi ve algoritma geliştirmeye ilişkin öz yeterlilik algısı üzerindeki etkilerini anlamak daha mümkün olabilir.
- Araştırma, fiziksel programlama araçlarının uzun dönemli etkisini değerlendirmek için izleme çalışmalarını içerebilir. Öğrencilerin bilgi işlemsel düşünme becerilerinin ve algoritma geliştirmeye ilişkin öz yeterlilik algılarının zamanla nasıl değiştiği ve geliştiği hakkında bir çalışma yapılabilir.
- Araştırmacılar, çalışma grubunu ve katılımcıları seçerken, farklı yaş grupları, eğitim düzeyleri ve programlama deneyimleri dikkate alabilirler. Örneğin, ilkokul, ortaokul ve lise düzeyindeki öğrencilerden farklı gruplar oluşturulabilir. Bu şekilde elde edilen sonuçlar, farklı yaş gruplarının programlama araçlarına tepkilerini ve etkinliklerden elde ettikleri faydayı gösterme açısından daha kapsamlı olabilir.
- Bilgi işlemsel düşünme becerisi ve algoritma geliştirme ile ilgili öz yeterlilik algısını değerlendirmek için daha farklı testler, anketler veya gözlem yöntemleri, öğrencilerin beceri düzeylerini ve algılarını değerlendirmek için kullanılabilir.
- Araştırma Solomon dört grup modeline göre yapılmış olup, istenirse derinlemesine anlayış elde etmenize ve katılımcıların deneyimlerini, algılarını ve düşüncelerini daha iyi anlamınıza yardımcı olabilecek yüzyüze görüşmeler yapılabilir. Böylelikle nitel boyutuda dikkate alan karma yöntemlere dayalı benzer çalışmalar yapılabilir.
- Araştırmacılar, farklı programlama platformlarının kullanıcı dostu olup olmama durumunu değerlendirmek amacıyla kullanılabilirlik çalışmaları yapabilirler.

5.2.2. Uygulayıcılar için öneriler

- Bu araştırma kapsamında hazırlanan ve kullanılan etkinliklerin, öğrencilerin farklı beceri düzeylerine ve ilgi alanlarına hitap edecek şekilde çeşitlendirilmesi önemlidir. Çocuklar, öğrenmeye karşı farklı düzeylerde ilgi gösterebilir, bu nedenle etkinliklerin daha ilgi çekici ve eğlenceli olmasına özen gösterilebilir.
- Öğrenciler etkinliklerini gerçekleştirirken düzenli geri bildirim vermek ve değerlendirme yapmak önemlidir. Geri bildirimler, etkinliklerin geliştirilmesine ve uygulayıcıların deneyimlerini daha iyi anlamasına yardımcı olur.
- Toxicode ile yalnızca bilgi işlemsel düşünme ve algoritma geliştirmeye ilişkin öz yeterlilik algısını artırmak için değil, beraberinde yaratıcı düşünme ve problem çözme gibi pek çok programlama becerilerinin de kazandırılması için kullanılabilir.

KAYNAKLAR

- Akçay, A. ve Çoklar, A. N. (2016). Bilişsel Becerilerin Gelişimine Yönelik Bir Öneri: Programlama Eğitimi. A. İşman, H. F. Odabaşı ve B. Akkoyunlu (Eds.), *Eğitim Teknolojileri Okumaları 2016* (s. 121-139). Ankara: The Turkish Online Journal of Educational Technology (TOJET).
- Akçayır, M. ve Akçayır, G. (2016). Üniversite öğrencilerinin yabancı dil eğitiminde artırılmış gerçeklik teknolojisi kullanımına yönelik görüşleri. *Erzincan Üniversitesi Eğitim Fakültesi Dergisi*, 18 (2), 1169-1186, Doi: 10.17556/jef.86406.
- Akgün, E., Özdemir, Ş. ve Karaoğlu Yılmaz, F. G. (2023). Bilgisayar Mühendisliği Öğrencilerinin Algoritma Sorusu Çözümlerinin Göz İzleme Yöntemi ile İncelenmesi ve Öğrenci Görüşleri . *Karaelmas Eğitim Bilimleri Dergisi*, 11 (1), 13-31 , Doi: 10.58638/kebd.1092963.
- Akoğlu, G. (2014). *Dil ve Konuşma Bozuklukları. Hasta ve Yaşlı Hizmetleri: Özel Gereksinimi Olan (Engelli) Bireyler ve Eğitimi*. 43-67. Ankara: Vize.
- Alagöz, S. (2022). *Ortaokul öğrencilerinin bilgi `işlemsel düşünmesine etki eden faktörler*, (Yayınlanmamış Yüksek Lisans Tezi), Abant İzzet Baysal Üniversitesi/Lisansüstü Eğitim Enstitüsü, Bolu.
- Aldağ, H. ve Tekdal, M. (2015). *Bilgisayar kullanımı ve programlama öğretiminde cinsiyet farklılıkları*. Proceeding of 1. Uluslararası Çukurova Kadın Çalışmaları Kongresi, 236–243.
- Alp, Y. (2019). *Blok Tabanlı Programlama Öğretiminin Ortaokul Öğrencilerinin Problem Çözme Becerisine ve Bilgisayara Yönelik Tutumuna Etkisi*, Yayınlanmamış Yüksek Lisans Tezi, İnönü Üniversitesi Eğitim Bilimleri, Malatya

- Atman Uslu, N., Mumcu, F. ve Eğin, F. (2018). Görsel programlama etkinliklerinin ortaokul öğrencilerinin bilgi-işlemsel düşünme becerilerine etkisi. *Ege Eğitim Teknolojileri Dergisi*, 2 (1), 19-31
- Atiker, B. (2019). *Programlama Öğretiminde Ortaokul Öğrencilerinin Bilgi İşlemsel Düşünme Becerilerinin Başarıya Etkileri*, Yayınlanmamış Doktora Tezi, İstanbul Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.
- Atmatzidou, S. and Demetriadis, S. (2016). Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems*, 75, 661–670, Doi: 10.1016/j.robot.2015.10.008.
- Aytekin, A., Sönmez-Çakır, F., Yücel, Y. B. ve Kulaöz, İ. (2018). Algoritmaların hayatımızdaki yeri ve önemi. *Avrasya Sosyal ve Ekonomi Araştırmaları Dergisi (Asead)*, 5(7), 151-162. URL: <https://dergipark.org.tr/tr/pub/asead/issue/41013/495619>
- Bakırcı, F. (2019). *Blok tabanlı programlama aracının 6. Sınıf öğrencilerinin programlama başarısı, algoritma geliştirme öz yeterlikleri ve güdülenmelerine etkisi*. (Yayınlanmamış Yüksek Lisans Tezi), Sakarya Üniversitesi/Eğitim Bilimleri Enstitüsü, Sakarya.
- Bandura, A. (1986). *Social Foundations of Thought and Action: A Social Cognitive Theory*. Englewood Cliffs, NJ: Prentice Hall.
- Bandura, A. (1997). *Self-efficacy: The exercise of control*. W H Freeman/Times Books/ Henry Holt and Co.
- Barr, D., Harrison, J. and Conery, L. (2011). *Computational Thinking: A Dijital Age Skill For Everyone*. URL: <http://Files.Eric.Ed.Gov/Fulltext/Ej918910.Pdf>. Son Erişim Tarihi: 18.06.2022

- Barr, B. V. and Stephenson, C. (2011). Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community? *ACM Inroads*, 2(1), 48-54, Doi: 10.1145/1929887.1929905.
- Başaran, B. (2018). *Arduino'nun elektrik deneylerine entegre edilmesinin ve deney raporlarının poster şeklinde hazırlanmasının, fen bilgisi öğretmen adaylarının fizik laboratuvarlarına, teknolojiye ve bilgi ve iletişim teknolojilerine yönelik tutumlarına etkisinin incelenmesi*, (Yayınlanmamış Yüksek Lisans Tezi), Kocaeli Üniversitesi/Fen Bilimleri Enstitüsü, Kocaeli.
- BattelleforKids (2023). *21st century learning experience*. URL: <https://www.battelleforkids.org/networks/p21/frameworks-resources> Son Erişim Tarihi: 18.05.2023
- Bilgin, E. Ç. (2022). *Video konferans aracılığıyla sunulan bilgisayarsız ve blok temelli etkinliklerin ortaokul öğrencilerinin bilgi işlemsel düşünme becerilerini geliştirmek için kullanımı*, Yayınlanmamış Yüksek Lisans Tezi , Boğaziçi Üniversitesi Sosyal Bilimler Enstitüsü, İstanbul.
- Blockly Games. (2023). Yarının programcıları için oyunlar. <https://blockly.games>, Son Erişim Tarihi: 21.08.2023.
- Braver, M. W. and Braver, S. L. (1988). Statistical treatment of the Solomon four-group design: A meta-analytic approach. *Psychological Bulletin*, 104(1), 150-154, Doi: 10.1037/0033-2909.104.1.150.
- Brennan, K. and Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In F. J. Levine (Ed.), *Proceedings of the 2012 Annual Meeting of the American Educational Research Association* (pp. 1-25). The University of British Columbia. <http://scratched.gse.harvard.edu/ct/files/AERA2012.pdf>, Son Erişim Tarihi: 22.11.2022.
- Brown, L. (2017). *Öğrenme ortamlarının etkisi*. Eğitim Yayıncılığı.

- Boulay, B. D. (1986). Some difficulties of learning to program. *Journal of Educational Computing Research*, 2(1), 57-73, Doi: 10.2190/3LFX-9RRF-67T8-UVK9.
- Brown, W. (2015). *Introduction to Algorithmic Thinking*. Available, URL: www.cs4fn.com/algorithmicthinking.php, Son Erişim Tarihi: 22.11.2021.
- Buitrago Flórez, F., Casallas, R., Hernández, M., Reyes, A., Restrepo, S. and Danies, G. (2017). Changing a generation's way of thinking: Teaching computational thinking through programming. *Review of Educational Research*, 87(4), 834-860 Doi: 10.3102/0034654317710096.
- Campbell, D. T. and Stanley, J. C. (1963). *Experimental and quasi-experimental designs for research*. Chicago: Rand McNally and Company.
- Cevahir, H. ve Özdemir, M. (2017). *Programlama öğretiminde karşılaşılan zorluklara yönelik öğretmen görüşleri ve çözüm önerileri*. Uluslararası Bilgisayar ve Öğretim Teknolojileri Sempozyumu.
- Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C. and Woollard, J. (2015), *Computational Thinking A Guide For Teachers*, URL: <https://community.computingatschool.org.uk/files/8550/original.pdf>, Son Erişim Tarihi: 22.12.2021.
- Cuny, J., Snyder, L. and Wing, J. M. (2010). *Demystifying computational thinking for non-computer scientists*. Unpublished manuscript, referenced in <http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf>. Son Erişim Tarihi: 22.12.2022.
- Çağıltay, N. E. ve Fal, M. (2014). *Scratch İle Programlama Öğreniyorum*, Ankara: ODTÜ Yayıncılık.
- Çakıroğlu, Ü., Çevik, İ., Köşeli, E. ve Aydın, M. (2021). Understanding students' abstractions in block-based programming environments: A performance based evaluation. *Thinking Skills and Creativity*, 41, Doi: 10.1016/j.tsc.2021.100888.

- Deng, W., Pi, Z., Lei, W., Zhou, Q. and Zhang, W. (2020). Pencil code improves learners' computational thinking and computer learning attitude. *Computer Applications in Engineering Education*, 28, 90- 104. Doi: 10.1002/cae.22177.
- del Olmo-Muno, J., Cozar-Gutierrez, R. and Gonzalez-Calero, J. A. (2020). Computational thinking through unplugged activities in early years of primary education. *Computers & Education*, 150, 68-94. Doi: 10.1016/j.compedu.2020.103832.
- Demir, Ö. ve Seferoglu, S. S. (2017). *Yeni Kavramlar, Farklı Kullanımlar: Bilgi-İşlemsel Düşünmeyle İlgili Bir Değerlendirme*, URL: https://www.academia.edu/34131104/Yeni_Kavramlar_Farkl%C4%B1_Kullan%C4%B1mlar_Bilgi_%C4%B0%C5%9Fl emsel_D%C3%BC%C5%9F%C3%BCnmeyle_%C4%B0lgili_Bir_De%C4%9Ferl endirme. Son Erişim Tarihi: 22.03.2023.
- Demir, Ü. ve Cevahir, H. (2020). Algoritmik Düşünme Yeterliliği ile Problem Çözme Becerisi Arasındaki İlişkinin İncelenmesi: Mesleki ve Teknik Anadolu Lisesi Örneği . *Kastamonu Eğitim Dergisi*, 28(4), 1610-1619, Doi: 10.24106/kefdergi.4179.
- Denning, P. J. (2003). Great principles of computing. *Communications of the ACM*, 46(11), 15-20, Doi: 10.1145/948383.948400.
- Doğan, D., Çınar, M., Bilgiç, H. G. ve Tüzün, H. (2015). Sarmal eğitsel oyun tasarımı modeline göre dijital oyun geliştirme süreci: <e-adventure> örneği. *Proceedings of International Play and Toy Congress*, 442-452. URL: https://www.researchgate.net/publication/309329510_Sarmal_egitsel_oyun_tasarim_i_modeline_gore_dijital_oyun_gelistirme_sureci_ornegi, Son Erişim Tarihi: 20.03.2023.
- Dönmez Usta, N. ve Turan Güntepe, E. (2019). Bilişim teknolojileri rehber öğretmenlerinin programlama araçlarına ilişkin deneyimlerinin incelenmesi. *Amasya Üniversitesi Eğitim Fakültesi Dergisi*, 8(2), 373-396. URL: <https://dergipark.org.tr/en/download/article-file/887176>

- Duncan, D. I., Bell, T. and Tanimoto, S. L. (2014). What's the big idea? Toward a theory of programming education. *Communications of the ACM*, 57(10), 44-52, Doi: 10.1145/2670757.2670774.
- EARGED. (2011). *MEB 21. Yüzyıl Öğrenci Profili*. URL: https://www.meb.gov.tr/earged/earged/21.%20yy_og_pro.pdf, Son Erişim Tarihi: 21.11.2022.
- Erol, O. ve Kurt, A. A. (2017). The effects of teaching programming with scratch on pre-service information technology teachers' motivation and achievement. *Computers in Human Behavior*, 77, 11-18. Doi: 10.1016/j.chb.2017.08.017.
- Eryilmaz, S. ve Deniz, G. (2021). *Programlama eğitiminde tinkercad kullanımının Öğrencilerin bilgi işlemsel düşünme becerisine ve algılarına etkisi*, Yayınlanmamış Yüksek Lisans Tezi, Gazi Üniversitesi Eğitim Bilimler Enstitüsü, Ankara.
- Feijóo-García, P. G., Kapoor, A., Gardner-McCune, C. and Ragan, E. (2021). Effects of a block-based scaffolded tool on students' introduction to hierarchical data structures. *IEEE transactions on education*, 1-9, Doi: 10.1109/TE.2021.3109604.
- Fessakis, G., Komis, V., Dimitracopoulou, A. and Prantsoudi, S. (2019). Overview of the computer programming learning environments for primary education. *Review of Science, Mathematics and ICT Education*, 13(1), 7-33. <https://pasithee.library.upatras.gr/review/article/view/3140/3434>, Son Erişim Tarihi: 21.05.2023.
- Fraenkel, J. R. and Wallen, N. E. (2008). *How to design and evaluate research in education. (8th ed.)*. New York: McGraw-Hill.
- Fraenkel, J. R., Wallen, N. E. and Hyun, H. H. (2012). *How to design and evaluate research in education (8th ed.)*. New York: McGraw-Hill.

- Fraser, N. (2015). Ten things we've learned from Blockly. In F. Turbak, D. Bau, J. Gray, C. Kelleher and J. Sheldon (Eds.), *Proceedings of the 2015 IEEE Blocks and Beyond Workshop* 49-50. Doi: 10.1109/BLOCKS.2015.7369000
- Futschek, G. (2006). Algorithmic thinking: the key for understanding computer science. In *International Conference on Informatics in Secondary Schools-Evolution and Perspectives*. 159-168. URL: https://link.springer.com/chapter/10.1007/11915355_15, Son Erişim Tarihi: 11.04.2023.
- Genç, Z. ve Karakuş, S. (2011). Tasarımla öğrenme: eğitsel bilgisayar oyunları tasarımında scratch kullanımı. *5th International Computer & Instructional Technologies Symposium*, 981-987. URL: <http://medya.beu.edu.tr/YazarMedya/385/Tasarimla%20Öğrenme%20-%20Eğitsel%20Bilgisayar%20Oyunlari%20Tasariminda%20Scratch%20Kullanimi.pdf>. Son Erişim Tarihi: 13.09.2022.
- Gouws, L. A., Bradshaw, K. and Wentworth, P. (2013). Computational thinking in educational activities: an evaluation of the educational game light-bot. *Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education*, 10-15, Doi:10.1145/2462476.2466518.
- Grover, S. and Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. *Educational Researcher*, 42(1), 38-43, Doi: 10.3102/ 0013189X12463051.
- Grover, S. and Pea, R. (2018). Computational Thinking: A Competency Whose Time Has Come. In S. Sentance, E. Barendsen, and C. Schulte (Eds.), *Computer Science Education: Perspectives on Teaching and Learning in School* (p. 20-38). URL: https://www.researchgate.net/publication/322104135_Computational_Thinking_A_Competency_Whose_Time_Has_Come. Son Erişim Tarihi: 21.10.2022.
- Guadalupe, R. and Fumarola, F. (2015). Teaching coding in schools: A new era for computer science education. *Informatics In Education*, 14(1), 1-16. URL: <https://files.eric.ed.gov/fulltext/EJ1064282.pdf>. Son Erişim Tarihi: 23.06.2022.

- Glbahar, Y., Kert, S. B. ve Kalelioėlu, F. (2019). Bilgi iřlemisel dřnme becerisine ynelik z yeterlik algısı leėi: geerlik ve gvenirlik alıřması. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)* , 10 (1) , 1-29, Doi: 10.16949/turkbilmat.385097.
- Glbahar, Y., Kalelioėlu, F., Kert, S. B., Kaplan, A., Koak, B., İliř, B. E., Karaosmanoėlu, G., Demirci, O. G., Kse, T. ve zgr, Y. K. (2018). Biliřim Teknolojileri ve Yazılım Dersi ėretmen Rehberi, Ankara: MEB.
- Gnbatar, M. S. ve Karalar, H. (2018). Gender differences in middle school students' attitudes and selfefficacy perceptions towards mBlock programming. *European Journal of Educational Research*, 7(4), 925–933. Doi:10.12973/eu-jer.7.4.925.
- Hsu, T. C., Chang, S. C. and Hung, Y. T. (2018). How To Learn And How To Teach Computational Thinking: Suggestions Based On A Review Of The Literature. *Computers & Education*, (126), 296-310. URL: <https://www.Learntechlib.org/p/200629/>, Son Eriřim Tarihi: 21.07.2022.
- Israel, M., Wherfel, Q. M., Pearson, J., Shehab, S. and Tapia, T. (2015). Empowering k–12 students with disabilities to learn computational thinking and computer Programming. *Teaching Exceptional Children*, 48(1), 45–53, Doi: 10.1177/0040059915594790.
- Jonsson, M., Bommarco, R., Ekbom, B., Smith, H. G., Bengtsson, J., Caballero-Lopez, B., Winqvist, C., Olsson, O. (2014). Ecological production functions for biological control services in agricultural landscapes. *Method. Ecol. Evol.* 5, 243–252, Doi: 10.1111/2041-210X.12149.
- Kalelioėlu, F. ve Glbahar, 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. URL: <https://infedu.vu.lt/journal/INFEDU/article/417/info>, Son Eriřim Tarihi: 21.04.2022.

- Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code.org. *Computers in Human Behavior*, 52, 200-210, Doi: 10.1016/j.chb.2015.05.047.
- Karasar, N. (2012). *Bilimsel Araştırma Yöntemi*. Ankara: Bilim Kitap Kirtasiye Yayınevi.
- Ke, F. (2014). An implementation of design-based learning through creating educational computer games: A case study on mathematics learning during design and computing. *Computers & Education*, 73, 26-39, Doi: 10.1016/j.compedu.2013.12.010.
- Kert, S. ve Uğraş, T. (2009). *Programlama eğitiminde sadelik ve eğlence: scratch örneği*. The First International Congress Of Educational Research. URL: <https://avesis.yildiz.edu.tr/yayin/a6b0507b-c955-4cd1-bcae-ab0a433c1ee6/programlama-egitiminde-sadelik-ve-eglence-scratch-ornegi>, Son Erişim Tarihi: 11.09.2022.
- Kert, S. B. (2017). Bilgisayar Bilimi Eğitime Giriş, Gülbahar, Y. (Ed.). *Bilgi işlemsel düşünmeden programlamaya*, Ankara: Pegem Akademi.
- Kirk, R. E. (2009). Experimental design. In Millsap, R. and Maydeu-Olivares, A. (Eds.), *Sage Handbook Of Quantitative Methods in Psychology*. Thousand Oaks, CA: Sage.
- Klee, R. and Bayrhuber, H. (2002) Multimedia im ökologieunterricht: lernprozesse und programmgestaltung. *Lehr - und Lernforschung in der Biologiedidaktik*, 187-198. URL: http://www.multimedialernen.at/materialien/IML_2004.pdf, Son Erişim Tarihi: 23.11.2022.
- Knuth, D. E. (1969) *The Art Of Computer Programming, Volume-1/The Fundamental Algorithms*, Series in Computer Science and Information Processing. URL: https://www.haio.ir/app/uploads/2022/01/The-Art-of-Computer-Programming-Vol.-1-Fundamental-Algorithms-3rd-Edition-by-Donald-E.-Knuth-z-lib.org_.pdf, Son Erişim Tarihi: 22.12.2022.

- Korkmaz, Ö., Çakır, R. ve Özden, M. Y. (2015). Bilgisayarca düşünme beceri düzeyleri ölçeğinin (BDBD) ortaokul düzeyine uyarlanması. *Gazi Eğitim Bilimleri Dergisi*, 1(2), 67-86. URL: <https://toad.halileksi.net/olcek/bilgisayarca-dusunme-beceri-duzeyleri-olcegi-orta-okullar-icin/> Son Erişim Tarihi: 11.07.2021.
- Korkmaz, Ö., Çakır, R., Özden, M. Y., Oluk, A. ve Sarıoğlu, S. (2015). Bireylerin bilgisayarca düşünme becerilerinin farklı değişkenler açısından incelenmesi. *Ondokuz Mayıs Üniversitesi Eğitim Fakültesi Dergisi*, 34(2), 68–87, Doi: 10.7822/omuefd.34.2.5.
- Korkmaz, Ö., Çakır, R. ve Özden, M. Y. (2017). A validity and reliability study of the Computational Thinking Scales (CTS). *Computers in Human Behavior*, 72, 558-569, Doi: 10.1016/j.chb.2017.01.005.
- Korkmaz, Ö., Balcı, H., Çakır, R. ve Uğur Erdoğan, F. (2021). Görsel programlama ortamlarında yapılan oyun geliştirme etkinliklerinin etkililiği. *Mehmet Akif Ersoy Üniversitesi Eğitim Fakültesi Dergisi* , (57) , 52-73. URL: <https://dergipark.org.tr/en/pub/maeuefd/issue/60050/589111>, Son Erişim Tarihi: 30.03.2022.
- Lye, S. Y. and 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, Doi: 10.1016/j.chb.2014.09.012.
- Ma, H., Zhao, M., Wang, H., Wan, X., Cavanaugh, T. W. and Liu, J. (2021). Promoting pupils' computational thinking skills and self-efficacy: A problem-solving instructional approach. *Education Tech Research Dev* 69, 1599–1616, Doi: 10.1007/s11423-021-10016-5.
- Marcelino, M. J., Pessoa, T., Vieira, C., Salvador, T. and Mendes, A. J. (2017). Learning computational thinking and scratch at distance. *Computers in Human Behavior*, 80 (C), 470-477, Doi: 10.1016/j.chb.2017.09.025.

- MEB. (2018). *Bilişim teknolojileri ve yazılım dersi öğretim programı (Ortaokul 5 ve 6. Sınıflar)*. URL: <http://mufredat.meb.gov.tr/ProgramDetay.aspx?PID=374>, Son Erişim Tarihi: 17.10.2022.
- Morelli, R., De Lanerolle, T., Lake, P., Limardo, N., Tamotsu, E. and Uche, C. (2011). Can Android app inventor bring computational thinking to K-12. *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education* (pp. 1-6). Association for Computing Machinery. http://hfoss.org/uploads/docs/appinventor_manuscript.pdf, Son Erişim Tarihi: 21.10.2022.
- Oluk, A. ve Korkmaz, Ö. (2016). Comparing students' scratch skills with their computational thinking skills in terms of different variables. *I. J. Modern Education and Computer Science*, 11, 1-7, Doi: 10.5815/ijmecs.2016.11.01.
- Oluk, A., Korkmaz, Ö. ve Oluk, H. A. (2018). Scratch'ın 5. Sınıf öğrencilerinin algoritma geliştirme ve bilgi işlemsel düşünme becerilerine etkisi. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)* , 9 (1), 54-71, Doi: 10.16949/turkbilmat.399588.
- Özdamar, K. (2013). *Paket Programları İle İstatistiksel Veri Analizi*, Ankara: Pelikan Yayıncılık.
- Ozoran, D., Çağıltay, N. ve Topalli, D. (2012, November). Using Scratch in introduction to programing course for engineering students. *Paper presented at 2nd International Engineering Education Conference (IEEC2012)*, Atılım Üniversitesi, Antalya.
- Özel, O. (2019). *Programlama yöntemlerinin ortaokul öğrencilerinin bilgi işlemsel düşünme becerisine yönelik öz yeterlik algısına ve programlama başarısına etkisi*, (Yayınlanmamış Yüksek Lisan Tezi), Marmara Üniversitesi/Eğitim Bilimleri Enstitüsü, İstanbul.
- Pajares, F. (2002). *Overview of Social Cognitive Theory and Self-Efficacy*. URL: <http://www.emory.edu/EDUCATION/mfp/eff.html>, Son Erişim Tarihi: 21.10.2022.

- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Retrieved, URL: <https://dl.acm.org/doi/pdf/10.5555/1095592>, Son Erişim Tarihi: 11.04.2023.
- Papadakis, S., Kalogiannakis, M. and Zaranis, N. (2016). ScratchJr as an educational tool for the development of computational thinking in preschoolers. *Education and Information Technologies*, 21(6), 1515-1529. URL: <https://dergipark.org.tr/en/pub/jiet/issue/42071/482727>, Son Erişim Tarihi: 01.04.2022.
- Partnership for 21st Century Skills. (2018). *21. Yüzyıl Öğrenme Çerçevesi*. URL: <https://www.battelleforkids.org/networks/p21>, Son Erişim Tarihi: 11.03.2023.
- Pinto, A. and Escudeiro, P. (2014). *The use of Scratch for the development of 21st century learning skills in ICT*. In 2014 9th Iberian Conference on Information Systems and Technologies (CISTI) (1–4), Doi: 10.1109/CISTI.2014.6877061.
- Pinto-Llorente, A. M., Casillas-Martín, S., Cabezas-González, M. and García-Peñalvo, F. J. (2018). Building, coding and programming 3D models via a visual programming environment. *Quality ve Quantity*, 52(6), 2455–2468, Doi: 10.1007/s11135-017-0509-4.
- Polat, E. ve Hopcan, S. (2019). Öğretmenlerin eğitsel mobil uygulama geliştirme aracı olarak mit app inventor'ı kabulü. *Kastamonu Eğitim Dergisi*, 27 (6) , 2459-2466, Doi: 10.24106/kefdergi.3300.
- Ramazanoğlu, M. (2021). Robotik kodlama uygulamalarının ortaokul öğrencilerinin bilgisayara yönelik tutumlarına ve bilgi işlemsel düşünme becerisine yönelik öz yeterlilik algılarına etkisi. *Türkiye Sosyal Araştırmalar Dergisi*, 25 (1), 163-174. URL: <https://dergipark.org.tr/tr/pub/tsadergisi/issue/61177/736602>, Son Erişim Tarihi: 11.03.2023.

- Ramalingam, V., LaBelle, D. and Wiedenbeck, S. (2004). Self-efficacy and mental models in learning to program. *Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education*, 171-175, Doi: 10.1145/1007996.1008042
- Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K. and Kafai, Y. (2009). Scratch: programming for all. *Communications of the ACM*, 52(11), 60-67, Doi: 10.1145/1592761.1592779.
- Roethlisberger, F. J. and Dickson, W. J. (1939). *Management and the Worker*. Harvard University Press.
- Rodríguez-Martínez, J. A., González-Calero, J. A. and Sáez-López, J. M. (2019). Computational thinking and mathematics using Scratch: an experiment with sixth-grade students. *Interactive Learning Environments*, 1-12, Doi:10.1080/10494820.2019.1612448.
- Sayın, Z. ve Seferoğlu, S. S. (2016). *Yeni Bir 21. Yüzyıl Becerisi Olarak Kodlama Eğitimi ve Kodlamanın Eğitim Politikalarına Etkisi*. XVIII. Akademik Bilişim Konferansı, Adnan Menderes Üniversitesi, Aydın.
- Schneider, G. M. and Gersting, J. (2018). *Invitation to computer science*. Boston, MA: Cengage Learning.
- Schunk, D. H. (1995). Self-efficacy, motivation, and performance. *Journal of Applied Sport Psychology*, 7(2), 112-137, Doi: 10.1007/978-1-4419-6868-5_10.
- Sedgewick, R. and Wayne, K. (2011). *Algorithms*. Princeton University: Pearson Education.
- Secer, M. (2020). *Bilişim teknolojileri ve yazılım dersinde Arduino kodlama ile kâğıt-kalem kodlama uygulamalarının öğrencilerin bilgi işlemsel düşünme becerileri, problem çözme becerileri ve STEM tutumları üzerine etkisi*, (Yayınlanmamış Yüksek Lisans Tezi), Mersin Üniversitesi/Eğitim Bilimleri Enstitüsü, Mersin.

- Selby, C. C., (2013). *Computational thinking: the developing definition*, ITiSCE Conference, 1-3 July 2013, England.
- Selby, C. C., (2014), *How can the teaching of programming be used to enhance computational thinking skills?* Thesis for the Degree of Doctor of Philosophy, University of Southampton.
- Sünger, İ., Çankaya, S. ve Durak, G. (2022). Artırılmış gerçeklik: lisansüstü tezlerin içerik analizi. *International Journal of Computers in Education*, 5(1), 31-48, Doi: 10.5281/zenodo.7504724 f.
- Syslo, M. M. and Kwiatkowska, A. B. (2013). *Informatics for All High School Students: a computational thinking approach*. In Proceedings of the 6th international conference on Informatics in Schools: Situation, Evolution, and Perspectives, Doi: 10.1007/978-3-642-36617-8_4.
- Şahiner, A. ve Kert, S. B. (2016). Komputasyonel düşünme kavramı ile ilgili 2006-2015 yılları arasındaki çalışmaların incelenmesi. *Avrupa Bilim ve Teknoloji Dergisi*, 5(9), 38-43. URL: <https://dergipark.org.tr/tr/pub/ejosat/issue/45153/565045>, Son Erişim Tarihi: 16.12.2022.
- Tabachnick, B.G. and Fidell, L.S. (2013). *Using Multivariate Statistics (6th Ed.)*. Boston: Allyn & Bacon.
- Tang, X., Yin, Y., Lin, Q., Hadad, R. and Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers & Education*, 148, 1-22, Doi: 10.1016/j.compedu.2019.103798.
- Taşdöndüren, T. ve Korucu, A. T. (2022). The Effect of Secondary School Students' Perceptions of Computing Technologies and Self-Efficiency on Attitudes Towards Coding. *Journal of Learning and Teaching in Digital Age*, 7(2), 200-209, Doi: 10.53850/joltida.1035448.

Tebliğler Dergisi (1997-2013). T.C. MEB Tebliğler Dergisi. URL: <http://tebligler.meb.gov.tr/>, Son Erişim Tarihi: 21.10.2022.

Toxicode (2023). About silent teacher. <https://silentteacher.toxicode.fr/about>, Son Erişim Tarihi: 21.08.2023.

Totan, H. N. (2021). *Blok tabanlı kodlama eğitiminin ortaokul öğrencilerinin bilgi işlemsel düşünme becerileri ve kodlama öğrenimine yönelik tutumlarına etkisi: blocky örneği*, Yayınlanmamış Yüksek Lisans Tezi, Necmettin Erbakan Üniversitesi Eğitim Bilimleri Enstitüsü, Konya.

Toytok, E. H. ve Çetin, A. (2015). The effects of pedagogical training program on students' professional attitudes and self efficacy levels. *The Turkish Online Journal of Educational Technology*, 607-611.

Turan, F. ve Akoğlu, G. (2011). İletişim, Dil ve Konuşma Bozuklukları Olan Çocuklar ve Eğitimleri. Baykoç, N. (Ed.). *Özel Gereksinimli Çocuklar ve Özel Eğitim*. Ankara: Eğiten Kitap Yayınevi.

Ulucan, J. (2022). *Kodlama ve robotik etkinliklerinin bilgi-işlemsel düşünme ve problem çözme becerilerine etkisi: meta-analiz çalışması*, Yayınlanmamış Yüksek Lisans Tezi , Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.

Ünal, A. (2019). *Dalı Sosyal Bilimler Lisesi öğrencilerine blok tabanlı programlama öğretiminin kaygı, bilişsel yük ve başarıya etkisi*, Yayınlanmamış Yüksek Lisans Tezi , Atatürk Üniversitesi Eğitim Bilimleri Enstitüsü, Erzurum.

Üzümcü, Ö. ve Bay, E. (2018). Eğitimde Yeni 21. Yüzyıl Becerisi: Bilgi-İşlemsel Düşünme, *Uluslararası Türk Kültür Coğrafyasında Sosyal Bilimler Dergisi*, 3(2): 1-16. URL: <https://dergipark.org.tr/tr/pub/turksosbilder/issue/46760/491067>, Son Erişim Tarihi: 11.06.2022.

- Vatansever, Ö. (2018). *Scratch ile programlama öğretiminin ortaokul 5. ve 6. Sınıf öğrencilerinin problem çözme becerisi üzerindeki etkisinin incelenmesi*. Yayınlanmamış Yüksek Lisans Tezi, Uludağ Üniversitesi Eğitim Bilimleri Enstitüsü, Bursa.
- Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., and Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, 25(1), 127-147, Doi: 10.1007/s10956-015-9581-5.
- Wei, W., Zhang, H., Cheng, Z. and Xiong, S. (2021). The effect of paired programming on computational thinking and self-efficacy of elementary school students. *Computers & Education*, 168, 104-215, Doi: 10.1016/j.compedu.2020.104023.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35. Doi: 10.1145/1118178.1118215.
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Phil. Trans. R. Soc. A*, (366) 3717–3725, Doi: 10.1098/rsta.2008.0118.
- Yadav, A., Mayfield, C., Zhou, N., Hambruch, S. and Korb, J. (2014). Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education*, 14(1), 1-5, Doi: 10.1145/2576872.
- Yaman, M. (2005). Solunum zinciri konusunda simülasyonla desteklenmiş bir bilgisayar programının öğrenme ve ilgiye etkisi. *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi*, 29, 222-228. URL: <http://dergipark.ulakbim.gov.tr/hunefd/article/download/5000048687/5000046007>, Son Erişim Tarihi: 22.10.2022.
- Yang, K., Tsai, C., Lee, L. and Chen, C. (2016). *The employment of online self-learning coding course to enhance logical reasoning ability for fifth and sixth grader students*. International Conference on Applied System Innovation, 26-30 May 2016 Okinawa, Japan, 616-620.

- Yüksel, S. (2017). *Scratch programı öğretiminde ayrılıp birleşme tekniği kullanımının öğrencilerin derse yönelik tutumuna akademik başarısına ve kalıcılığa etkisi* (Yayınlanmamış Yüksek Lisans Tezi). Adnan Menderes Üniversitesi/Eğitim Bilimleri Anabilim Dalı, Aydın
- Yükseltürk E. ve Altıok, S. (2015). Bilişim Teknolojileri Öğretmen Adaylarının Bilgisayar Programlama Öğretimine Yönelik Görüşleri. *Amasya Üniversitesi Eğitim Fakültesi Dergisi*, 4(1), 50-65. URL: <http://dergi.amasya.edu.tr/article/view/5000087734>, Son Erişim Tarihi: 21.12.2022.
- Yünkül, E., Durak, G., Çankaya, S. ve Abidin, Z. (2017). The Effects of Scratch Software on Students' Computational Thinking Skills. *Necatibey Eğitim Fakültesi Elektronik Fen ve Matematik Eğitimi Dergisi (EFMED)*, 11(2), 502-517. URL: <https://dergipark.org.tr/tr/download/article-file/396510>, Son Erişim Tarihi: 08.10.2022.
- Zsako, L. and Szlavi, P. (2012). ICT competences: Algorithmic thinking. *Acta Didactica Napocensia*, 5(2), 49-58 URL: https://www.researchgate.net/publication/235724026_ICT_COMPETENCES_ALGORITHMIC_THINKING, Son Erişim Tarihi: 26.10.2022.
- Zimmerman, B. J., Bandura, A. and Martinez-Pons, M. (1992). Self-motivation for academic attainment: The role of self-efficacy beliefs and personal goal setting. *American Educational Research Journal*, 29(3), 663-676, Doi: 10.2307/1163261.



EKLER

EK-1. Haftalık Çalışma Planı

Haftalar	Ders Saatleri	Deney Grubu		Kontrol Grubu	
		Konu Başlıkları	Etkinlikler	Konu Başlıkları	Etkinlikler
1.Hafta	2	- Ön Test - Bilgisayarca Düşünme Becerileri Düzeyleri Ölçeği - Algoritma Geliştirmeye İlişkin Öz Yeterlik Algısı Ölçeği - Programlamayla ilgili temel kavramlar - Blok tabanlı programlama aracının arayüzü ve özellikleri (http://www.toxicode.fr/learn) - Algoritma oluşturma - Öğrendiklerimi Uyguluyorum (https://little-dot.toxicode.fr/ 0-5)	Toxicode İle Tanışyorum Etkinliği (Ek_C)	- Ön Test - Bilgisayarca Düşünme Becerileri Düzeyleri Ölçeği - Algoritma Geliştirmeye İlişkin Öz Yeterlik Algısı Ölçeği- - Programlamayla ilgili temel kavramlar - Blok tabanlı programlama aracının arayüzü ve özellikleri (https://blockly.games) - Algoritma oluşturma - Öğrendiklerimi Uyguluyorum (https://blockly.games/puzzle?lang=tr)	Google Blockly İle Tanışyorum Etkinliği (Ek_D)
2.Hafta	2	- Doğrusal mantık yapısı - Doğrusal mantık yapısını kullanan algoritmalar geliştirme - Döngü yapısı ve işlevleri - Öğrendiklerimi Uyguluyorum (https://compute-it.toxicode.fr/ 1-9) - Karar yapısı ve işlevleri - Öğrendiklerimi Uyguluyorum (https://compute-it.toxicode.fr/ 10-35)	Toxicode Doğrusal Yapılar ve Döngüler Etkinliği (Ek_E) Toxicode Karar Yapıları Başlangıç Seviyesi Etkinliği (Ek_F)	- Doğrusal mantık yapısı - Doğrusal mantık yapısını kullanan algoritmalar geliştirme - Döngü yapısı ve işlevleri - Öğrendiklerimi Uyguluyorum (https://blockly.games/maze?lang=tr 1-5) - Karar yapısı ve işlevleri - Öğrendiklerimi Uyguluyorum (https://blockly.games/maze?lang=tr 6-10)	Blockly Doğrusal Yapılar ve Döngüler Etkinliği (Ek_E) Blockly Karar Yapıları Başlangıç Seviyesi Etkinliği (Ek_F)

EK-1. Haftalık Çalışma Planı (Devam)

3.Hafta 2	- Karar yapılarını içeren algoritmalar geliştirme - Öğrendiklerimi Uyguluyorum (https://little-dot.toxicode.fr/ 6-20)	Karar yapılarını içeren algoritmalar geliştirme - Öğrendiklerimi Uyguluyorum (https://blockly.games/bird?lang=tr)	Toxicode Karar Yapıları Orta Seviye Etkinliği (Ek_I)
4.Hafta 2	- Öğrendiklerimi Uyguluyorum (https://little-dot.toxicode.fr/ 21-30) - Son Test Bilgisayarca Düşünme Becerileri Düzeyleri Ölçeği Algoritma Geliştirmeye İlişkin Öz Yeterlik Algısı Ölçeği	- Öğrendiklerimi Uyguluyorum (https://blockly.games/movie?lang=tr) - Son Test Bilgisayarca Düşünme Becerileri Düzeyleri Ölçeği Algoritma Geliştirmeye İlişkin Öz Yeterlik Algısı Ölçeği	Toxicode Karar Yapıları İleri Seviye Etkinliği (Ek_K)
			Blockly Karar Yapıları Orta Seviye Etkinliği (Ek_J)
			Blockly Karar Yapıları İleri Seviye Etkinliği (Ek_L)

EK-2. Programlama öğretimi konu başlıkları

Haftalar	Ders Saatleri	Konu Başlıkları
1. Hafta	2	Ön Test - Bilgisayarca Düşünme Becerileri Düzeyleri Ölçeği - Algoritma Geliştirmeye İlişkin Öz Yeterlik Algısı Ölçeği Programlama - Programlamayla ilgili temel kavramlar - Blok tabanlı programlama aracının arayüzü ve özellikleri - Öğrendiklerimi Uyguluyorum (http://www.toxicode.fr/learn) – (https://blockly.games) - Algoritma oluşturma - Öğrendiklerimi Uyguluyorum (https://little-dot.toxicode.fr/0-5) – (https://blockly.games/puzzle?lang=tr)
2. Hafta	2	- Doğrusal mantık yapısı - Doğrusal mantık yapısını kullanan algoritmalar geliştirme - Döngü yapısı ve işlevleri - Öğrendiklerimi Uyguluyorum (https://compute-it.toxicode.fr/0-9) – (https://blockly.games/maze?lang=tr 1-5) - Karar yapısı ve işlevleri - Öğrendiklerimi Uyguluyorum (https://compute-it.toxicode.fr/10-35) - (https://blockly.games/maze?lang=tr 6-10)
3. Hafta	2	- Karar yapılarını içeren algoritmalar geliştirme - Öğrendiklerimi Uyguluyorum (https://little-dot.toxicode.fr/6-20) - (https://blockly.games/bird?lang=tr)

EK-2. Programlama öğretimi konu başlıkları (Devam)

4. Hafta	2	- Öğrendiklerimi Uyguluyorum (https://little-dot.toxicode.fr/ 21-30) - (https://blockly.games/movie?lang=tr) - Son Test Bilgisayarca Düşünme Becerileri Düzeyleri Ölçeği Algoritma Geliştirmeye İlişkin Öz Yeterlik Algısı Ölçeği
-----------------	----------	---



EK-3. Bilgi İşlemsel Düşünme Becerilerinin ve Algoritma Geliştirmeye İlişkin Ders Kazanımlarına Entegre Edildiği Ünite Planı

<i>Hafta</i>	<i>Ders Saati</i>	<i>Kazanımlar</i>	<i>Konular</i>	<i>Yöntem ve Teknik</i>
1	2	5.5.2.1. Programlamayla ilgili temel kavramları açıklar. 5.5.2.2. Blok tabanlı programlama aracının arayüzünü ve özelliklerini tanır. 5.5.2.3. Blok tabanlı programlama ortamında sunulan hedeflere ulaşmak için doğru algoritmayı oluşturur.	1. Bilgisayarca Düşünme Becerileri Düzeyleri Ölçeği Uygulaması 2. Algoritma Geliştirmeye İlişkin Öz Yeterlik Algısı Ölçeği Uygulaması 3. Programlamayla ilgili temel kavramlar 4. Blok tabanlı programlama aracının arayüzü ve özellikleri 5. Algoritma oluşturma	Anlatım, Soru-Cevap, Problem Çözme
Etkinlikler		• Öğrencilere ön testler uygulanır. • Bilgisayarların nasıl çalıştığı sorusu öğrencilere yöneltilir. Ardından öğrencilere bilgisayarın tanımı yapılarak komut kavramından bahsedilir. Bilgisayara komutların kim tarafından nasıl girildiği sorusu sorularak öğrencilerin cevaplama istenir. • Öğrencilere bilgisayar programlarının nasıl hazırlandığı sorusu yöneltilir? Algoritma kavramı üzerinde durulur. Teneffüste kantine gitme algoritması sizce nasıldır sorusu öğrencilere yönlendirilerek algoritmalar oluşturma istenir. • Bilgisayar programının ne olduğu ve kimler tarafından nasıl oluşturulabileceği üzerinde durularak öğrencilere sizler de yazdığınız komutları çalıştıran bir program yapmak ister miydiniz sorusu yöneltilir. • Öğrencilere; bilgisayar programlama dillerinin bir yabancı dil gibi öğrenilmesi zaman alan ve ilk bakışta karmaşık gelen diller olduğu, blok tabanlı programlama araçları ile belirli bir program dilinde kullanılan komutun işlevini anlatan bloklara çevrildiği ve bu şekilde iletişimin kolaylaştığı söylenir.		

EK-3. Bilgi İşlemsel Düşünme Becerilerinin ve Algoritma Geliştirmeye İlişkin Ders Kazanımlarına Entegre Edildiği Ünite Planı (Devam)

Etkinlikler		● Deney gurubuna: ▪ “Bunlar arasında en yaygın olanları Google Blockly, Scratch ve Toxicode platformlarıdır. Biz derslerimizde Toxicode platformunu kullanacağız” denir. ▪ “Toxicode, bilgisayar programlama konusunda daha önce deneyimi olmayan çocuklara, programlama eğitimi vermek üzere hazırlanmış, eğitici oyunlardır” denilerek platform (http://www.toxicode.fr/learn) öğrencilere tanıtılarak öğrencilerden Toxicode’la tanışıyorum etkinliğini (ek4) yapması istenir. ● Kontrol gurubuna: ▪ “Bunlar arasında en yaygın olanları Google Blockly, Scratch ve Toxicode platformlarıdır. Biz derslerimizde Google Blockly platformunu kullanacağız” denir. ▪ “Blockly Oyunlar, bilgisayar programlama konusunda daha önce deneyimi olmayan çocuklara, programlama eğitimi vermek üzere hazırlanmış, eğitici oyunlardır” denilerek platform (https://blockly.games) öğrencilere tanıtılarak öğrencilerden Google Blockly ile tanışıyorum etkinliğini (ek5) yapması istenir.		
Hafta	Ders Saati	Kazanımlar	Konular	Yöntem ve Teknik
2	2	5.5.2.4. Doğrusal mantık yapısını açıklar. 5.5.2.5. Doğrusal mantık yapısını kullanan algoritmalar geliştirir. 5.5.2.8. Döngü yapısını ve işlevlerini açıklar. 5.5.2.6. Karar yapısını ve işlevlerini açıklar.	1. Doğrusal mantık yapısı 2. Doğrusal mantık yapısını kullanan algoritmalar geliştirme 3. Döngü yapısı ve işlevleri 4. Karar yapısı ve işlevleri	Anlatım, Soru-Cevap, Gösterip Yaptırma, Problem Çözme

EK-3. Bilgi İşlemsel Düşünme Becerilerinin ve Algoritma Geliştirmeye İlişkin Ders Kazanımlarına Entegre Edildiği Ünite Planı (Devam)

Etkinlikler	• İlk olarak öğrencilere, algoritma kavramının ne olduğunu hatırlayıp hatırlamadıkları sorulur ve öğrencilerden yanıtlar alındıktan sonra algoritma tanımı ve önceki hafta yapılan teneffüste kantine gitme algoritması öğrencilere hatırlatılır. • Birbirini tekrar eden işlemler için komutların gerçek hayatta nasıl verilebileceği sorusu öğrencilere yöneltilir. Mesela bir yönlendirme oyununda “sağa dön, sağa dön, sağa dön, sağa dön” şeklinde ayrı ayrı 4 komut vermek yerine “4 defa sağa dön” komutu vermenin daha doğru olacağı söylenerek bilgisayarda da tekrar eden komutlar için döngü yapılarının oluşturulabileceği söylenerek döngü kavramı açıklanır. • Deney gurubundan Toxicode doğrusal yapılar ve döngüler etkinliğini (ek6) yapması istenir. Bu etkinliğin kuralları açıklandıktan sonra öğrencilerden etkinliğin görevlerini yapması beklenir. • Kontrol gurubundan Blockly doğrusal yapılar ve döngüler etkinliğini (ek7) yapması istenir. Bu etkinliğin kuralları açıklandıktan sonra öğrencilerden etkinliğin görevlerini yapması beklenir. • Öğrencilere ne zaman su içtikleri sorularak sadece belli şartlar altında belli görevleri yapmakta olduğumuz açıklanır. Bilgisayarda da belli şartlar altında belli görevlerin gerçekleşmesini sağlayabilecek yapılar olduğu söylenerek bilgisayarlarda kullanılan koşul yapıları anlatılarak mantıksal operatörler kavramı öğrencilere açıklanır. • Deney gurubundan Toxicode karar yapıları başlangıç seviyesi etkinliğini (ek8) yapması istenir. Bu etkinliğin kuralları açıklandıktan sonra öğrencilerden etkinliğin görevlerini yapması beklenir. • Kontrol gurubundan Blockly karar yapıları başlangıç seviyesi etkinliğini (ek9) yapması istenir. Bu etkinliğin kuralları açıklandıktan sonra öğrencilerden etkinliğin görevlerini yapması beklenir.
---------------------------	--

EK-3. Bilgi İşlemsel Düşünme Becerilerinin ve Algoritma Geliştirmeye İlişkin Ders Kazanımlarına Entegre Edildiği Ünite Planı (Devam)

<i>Hafta</i>	<i>Ders Saati</i>	<i>Kazanımlar</i>	<i>Konular</i>	<i>Yöntem ve Teknik</i>
3	2	5.5.2.7. Karar yapılarını içeren algoritmalar geliştirir.	1. Karar yapılarını içeren algoritmalar geliştirme	Anlatım, Soru-Cevap, Gösterip Yaptırma, Problem Çözme
Etkinlikler		• İlk olarak öğrencilere, önceki hafta neler yaptıklarını hatırlayıp hatırlamadıkları sorusu yöneltilir ve yanıtlar alındıktan sonra aşamaları nasıl geçtikleri ve bu süreçte hangi kavramları kullandıkları hatırlatılır. • Öğrencilere bu hafta yapılacak çalışmanın daha önce yaptıkları farklı uygulamalara benzer olduğu söylenir. • Deney gurubundan Toxicode karar yapıları orta seviye etkinliğini (ek10) yapması istenir. Bu etkinliğin kuralları açıklandıktan sonra öğrencilerden etkinliğin görevlerini yapması beklenir. • Kontrol gurubundan Blockly karar yapıları orta seviye etkinliğini (ek11) yapması istenir. Bu etkinliğin kuralları açıklandıktan sonra öğrencilerden etkinliğin görevlerini yapması beklenir.		

EK-3. Bilgi İşlemsel Düşünme Becerilerinin ve Algoritma Geliştirmeye İlişkin Ders Kazanımlarına Entegre Edildiği Ünite Planı (Devam)

<i>Hafta</i>	<i>Ders Saati</i>	<i>Kazanımlar</i>	<i>Konular</i>	<i>Yöntem ve Teknik</i>
4	2	5.5.2.7. Karar yapılarını içeren algoritmalar geliştirir.	1. Karar yapılarını içeren algoritmalar geliştirme 2. Bilgisayarca Düşünme Becerileri Düzeyleri Ölçeği 3. Problem Çözme Başarı Testi	
Etkinlikler		• Daha önceki haftalarda öğrenilen bilgiler tekrar edilir. • Deney gurubundan Toxicode karar yapıları ileri seviye etkinliğini (ek12) yapması istenir. Bu etkinliğin kuralları açıklandıktan sonra öğrencilerden etkinliğin görevlerini yapması beklenir. • Kontrol gurubundan Blockly karar yapıları ileri seviye etkinliğini (ek13) yapması istenir. Bu etkinliğin kuralları açıklandıktan sonra öğrencilerden etkinliğin görevlerini yapması beklenir. • Öğrencilere son testler uygulanır.		

EK-4. Toxicode’la Tanışıyorum Etkinliği

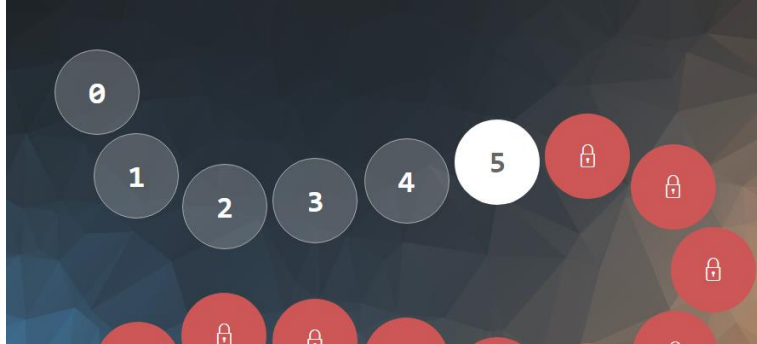
Etkinlik adı: Toxicode’la Tanışıyorum	Etkinlik süresi: 20 dk.
1- http://www.toxicode.fr/learn adresinden toxicode çevrimiçi kodlama ortamı açılır. 2- Ana sayfadan Toxicode etkinlikleri sırasıyla ne amaçla kullanıldığı nasıl başlatıldığı vb. gösterilerek tanıtılır.  COMPUTE IT Let's switch roles : this time YOU are the computer ! Read and interpret the programs to find the right trajectory and win the challenges. You will have to focus and use your intuitive abilities to understand some core concepts of programming. learn more PLAY  LITTLE DOT ADVENTURE Little Dot Adventure mixes differents aspects of Compute It and Paint Duel. Like in Compute It you have to execute yourself the right code to solve different levels. Only this time you will have to choose correctly your moves at each turn. You will have to anticipate your moves and discover different beginners coding mechanics. learn more PLAY 3- Ana sayfadan Little Dot Adventure kodlama oyunu başlatılır. 4- Aşağıda gösterilen uygulama başladığında çıkan resim tıklanır.	

EK-4. Toxicode’la Tanışıyorum Etkinliği (Devam)

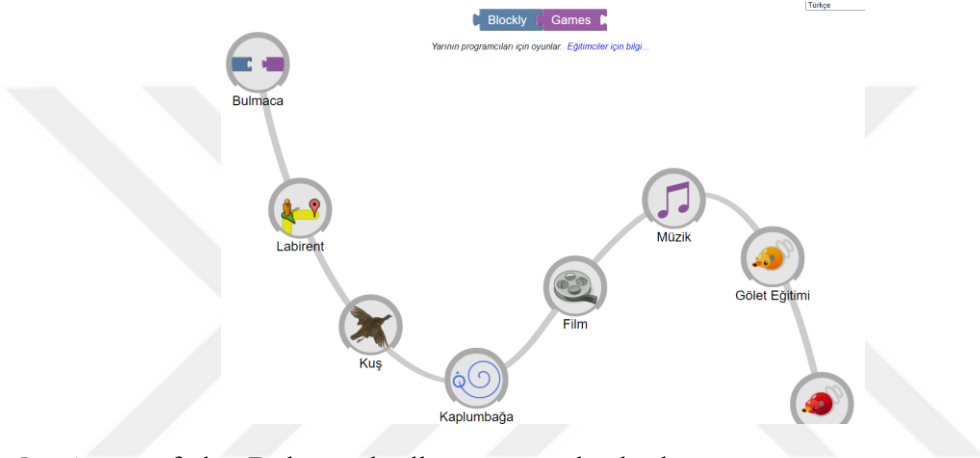
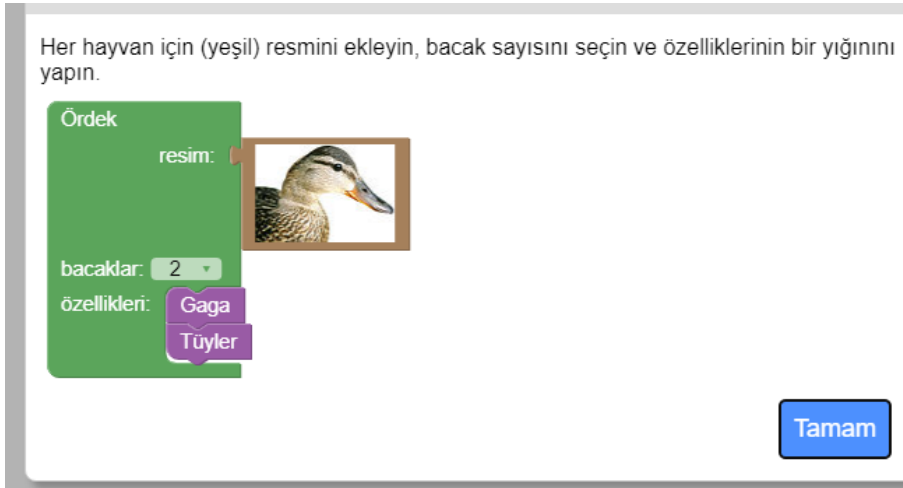
- 5- Aşağıdaki resimlerde sırasıyla gösterilen gelen ekranlarda bizden ne yapmamız istenildiğinin öğrenciler tarafından bulunması sağlanır.



- 6- Kart seçme ve algoritma oluşturma işlemlerinin nasıl yapılacağının öğrenciler tarafından keşfedilmesinin sağlanmasının ardından öğrencilerden toxicode çevrimiçi kodlama ortamını açıp 5. aşamaya kadar görevleri yapması istenir.



EK-5. Google Blockly İle Tanışyorum Etkinliği

Etkinlik adı: Google Blockly İle Tanışyorum	Etkinlik süresi: 20 dk.
1- https://blockly.games adresinden Google Blockly çevrimiçi kodlama ortamı açılır.  <i>Yarının programcıları için oyunlar. Eğitimciler için bilgi...</i> 2- Ana sayfadan Google Blockly etkinlikleri sırasıyla ne amaçla kullanıldığı nasıl başlatıldığı vb. gösterilerek tanıtılır.  3- Ana sayfadan Bulmaca kodlama oyunu başlatılır. 4- Aşağıdaki resimde gösterilen bizden ne yapmamız istenildiğinin gösterildiği gelen ekran ile ilgili görsel üzerinden sonraki adımlarda ne yapılması gerektiğinin öğrenciler tarafından bulunması sağlanır.  5- Block yerleştirme ve algoritma oluşturma işlemlerinin nasıl yapılacağının öğrenciler tarafından keşfedilmesinin sağlanmasının ardından öğrencilerden Google Blockly çevrimiçi kodlama ortamını açıp Bulmaca oyununu yapması istenir.	

EK-6. Toxicode Doğrulsal Yapılar ve Döngüler Etkinliği

Etkinlik adı: Toxicode Doğrulsal Yapılar ve Döngüler

Etkinlik süresi: 20 dk.

1- <http://www.toxicode.fr/learn> adresinden toxicode çevrimiçi kodlama ortamı açılır.

2- Ana sayfadan Compute It kodlama oyunu başlatılır.

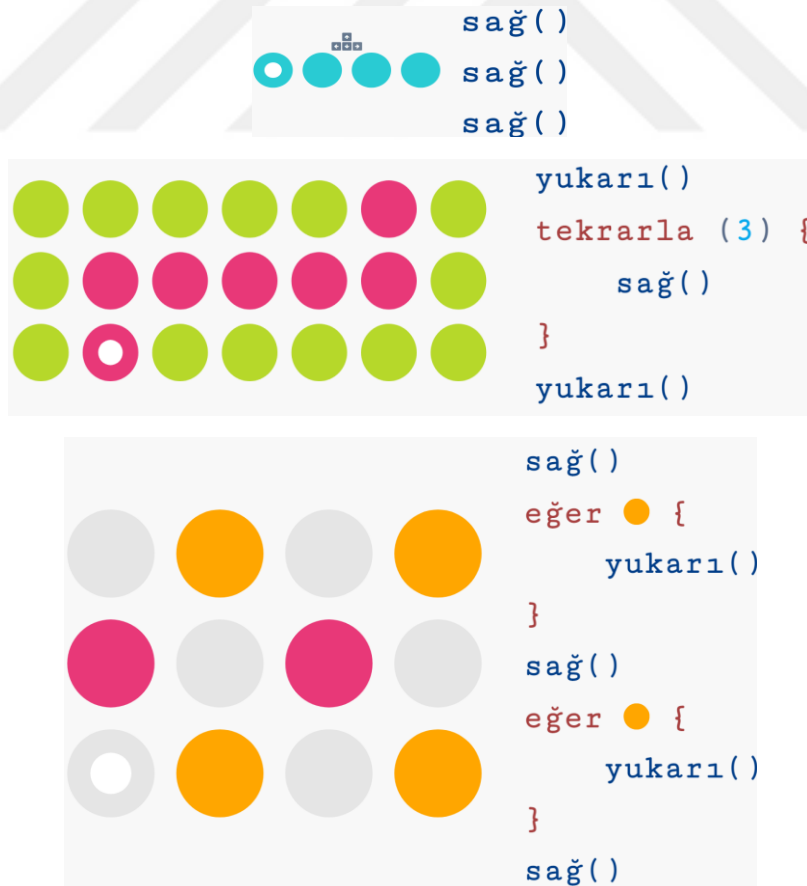
Not: Direkt olarak <https://compute-it.toxicode.fr/> adresi ile de uygulama açılabilir.

3- Aşağıda gösterilen uygulama başladığında çıkan resim tıklanır.



4- Öğrencilerden sırasıyla (ilk 10 uygulama) döngüleri öğretici görevleri yapması istenir.

Örnek ekran görüntüleri:



Etkinlik adı: Toxicode Karar Yapıları Başlangıç Seviyesi

Etkinlik süresi: 30 dk.

1- <http://www.toxicode.fr/learn> adresinden toxicode çevrimiçi kodlama ortamı açılır.

2- Ana sayfadan Compute It kodlama oyunu başlatılır.

Not: Direkt olarak <https://compute-it.toxicode.fr/> adresi ile de uygulama açılabilir.

3- Aşağıda gösterilen uygulama başladığında çıkan resim tıklanır.

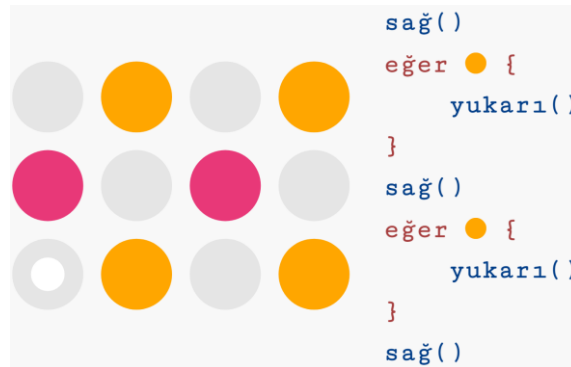


4- Ekranı gelen sayfada 11 sayısı tıklanır.



5- Öğrencilerden sırasıyla karar yapılarını öğretici görevleri yapması istenir.

Örnek ekran görüntüleri:



Etkinlik adı: Blockly Doğrulsal Yapılar ve Döngüler

Etkinlik süresi: 20 dk.

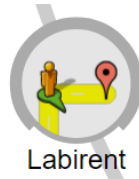
1- <https://blockly.games> adresinden Google Blockly çevrimiçi kodlama ortamı açılır.



Yarının programcıları için oyunlar. [Eğitimciler için bilgi...](#)

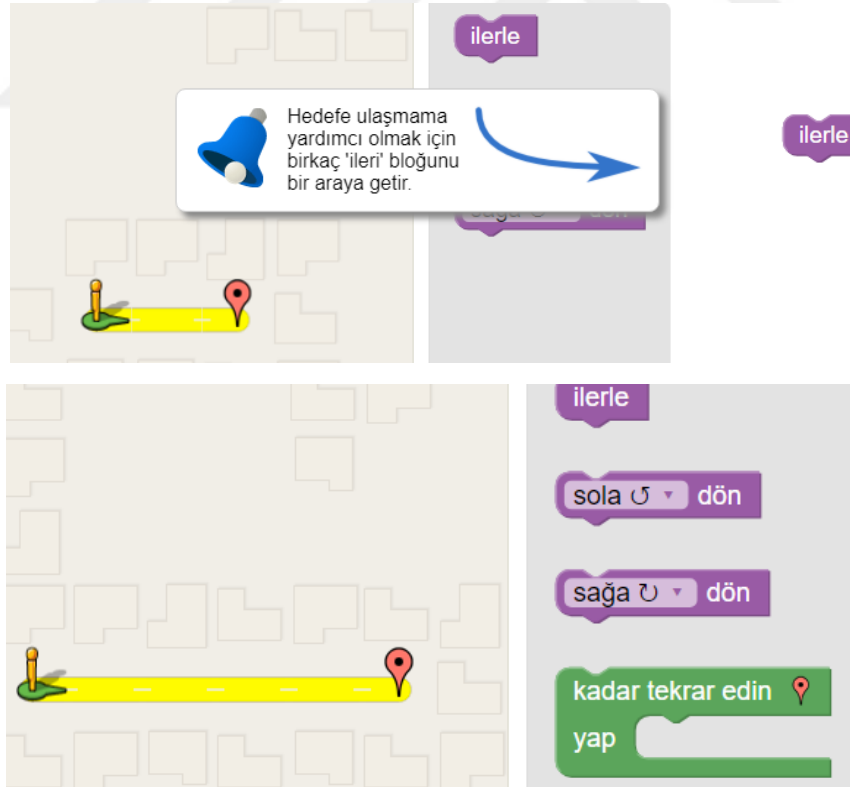
2- Ana sayfadan Labirent kodlama oyunu başlatılır.

Not: Direkt olarak <https://blockly.games/maze?lang=tr> adresi ile de uygulama açılabilir.



3- Öğrencilerden sırasıyla (ilk 5 uygulama) döngüleri öğreten görevleri yapması istenir.

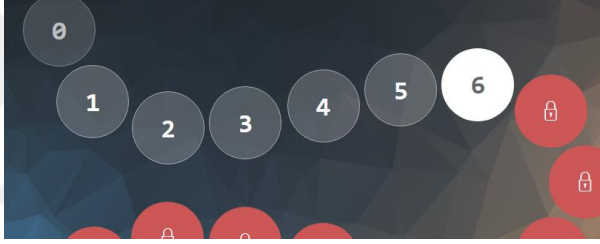
Örnek ekran görüntüleri:



EK-9. Blockly Karar Yapıları Başlangıç Seviyesi Etkinliği

Etkinlik adı: Blockly Karar Yapıları Başlangıç Seviyesi	Etkinlik süresi: 30 dk.
1- https://blockly.games adresinden Google Blockly çevrimiçi kodlama ortamı açılır. <i>Yarının programcıları için oyunlar: Eğitimciler için bilgi...</i> 2- Ana sayfadan Labirent kodlama oyunu başlatılır. Not: Direkt olarak https://blockly.games/maze?lang=tr adresi ile de uygulama açılabilir. 3- Öğrencilerden 6. oyundan itibaren sırasıyla karar yapılarını öğreten görevleri yapması istenir. Örnek ekran görüntüleri:	

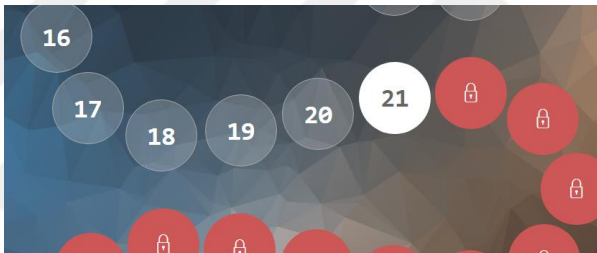
EK-10. Toxicode Karar Yapıları Orta Seviye Etkinliđi

Etkinlik adı: Toxicode Karar Yapıları Orta Seviye	Etkinlik süresi: 60 dk.
1- http://www.toxicode.fr/learn adresinden toxicode çevrimiçi kodlama ortamı açılır. 2- Ana sayfadan Little Dot Adventure kodlama oyunu başlatılır. Not: Direkt olarak https://little-dot.toxicode.fr/ adresi ile de uygulama açılabilir. 3- Aşağıda gösterilen uygulama başladığında çıkan resim tıklanır.	
	
4- Ekranı gelen sayfada 6 sayısı tıklanır.	
	
5- Öğrencilerden toxicode çevrimiçi kodlama ortamını açıp 20. aşamaya kadar görevleri yapması istenir. Örnek ekran görüntüleri:	
	

EK-11. Blockly Karar Yapıları Orta Seviye Etkinliği

Etkinlik adı: Blockly Karar Yapıları Orta Seviyesi	Etkinlik süresi: 60 dk.
1- https://blockly.games adresinden Google Blockly çevrimiçi kodlama ortamı açılır. <i>Yarının programcıları için oyunlar: Eğitimciler için bilgi...</i> 2- Ana sayfadan Kuş kodlama oyunu başlatılır. Not: Direkt olarak https://blockly.games/bird?lang=tr adresi ile de uygulama açılabilir. 3- Öğrencilerden 1. oyundan itibaren sırasıyla karar yapılarını öğretene görevleri açıp son göreve kadar yapması istenir. Örnek ekran görüntüleri:	

EK-12. Toxicode Karar Yapıları İleri Seviye Etkinliği

Etkinlik adı: Toxicode Karar Yapıları İleri Seviye	Etkinlik süresi: 30 dk.
1- http://www.toxicode.fr/learn adresinden toxicode çevrimiçi kodlama ortamı açılır. 2- Ana sayfadan Little Dot Adventure kodlama oyunu başlatılır. Not: Direkt olarak https://little-dot.toxicode.fr/ adresi ile de uygulama açılabilir. 3- Aşağıda gösterilen uygulama başladığında çıkan resim tıklanır.	
	
4- Ekranı gelen sayfada 21 sayısı tıklanır.	
	
5- Öğrencilerden toxicode çevrimiçi kodlama ortamını açıp son göreve kadar görevleri yapması istenir. Örnek ekran görüntüleri:	
	

EK-13. Blockly Karar Yapıları İleri Seviye Etkinliği

Etkinlik adı: Blockly Karar Yapıları İleri Seviyesi	Etkinlik süresi: 30 dk.
1- https://blockly.games adresinden Google Blockly çevrimiçi kodlama ortamı açılır. <i>Yarının programcıları için oyunlar. Eğitimciler için bilgi...</i> 2- Ana sayfadan Film kodlama oyunu başlatılır. Not: Direkt olarak https://blockly.games/movie?lang=tr adresi ile de uygulama açılabilir. 1- Öğrencilerden 1. oyundan itibaren sırasıyla karar yapılarını öğreten görevleri açıp son göreve kadar yapması istenir. Örnek ekran görüntüsü:	

EK-17. Deneysel Süreçlerde Kullanılan Ölçekler (AGYÖ ve BDÖ) (Devam)

Sevgili Öğrenciler

Aşağıdaki maddeler bilgisayarca düşünme becerilerini ölçmeye dönük hazırlanmış ve bir araştırmada kullanılacaktır. Araştırma dışında başka hiçbir amaçla kullanılmayacaktır. Lütfen her bir maddeyi dikkatle okuyup, sizi yansıtmaya düzeyini en olumludan (5) en olumsuz (1) doğru puanlayınız. Katılımınızdan dolayı şimdiden teşekkür ederiz.

1	Kararlarının çoğundan emin olan insanları severim	1	2	3	4	5
2	Yeni bir durumla karşılaştığımda ortaya çıkabilecek sorunları çözebileceğime inancım vardır.	1	2	3	4	5
3	Bir sorunumu çözmek üzere plan yaparken o planı yürütebileceğime güvenirim.	1	2	3	4	5
4	Bir sorunla karşılaştığımda, başka konuya geçmeden önce durur ve o sorun üzerinde düşünürüm.	1	2	3	4	5
5	Bir problemin çözümünü verecek denklemi hemen kurabilirim.	1	2	3	4	5
6	Matematiksel sembol ve kavramlar yardımıyla yapılan anlatımları daha kolay öğrendiğimi düşünürüm	1	2	3	4	5
7	Sayılar arasındaki ilişkileri kolaylıkla yakalayabildiğime inanırım	1	2	3	4	5
8	Sözel olarak ifade edilen bir matematik problemini sayısallaştırabilirim.	1	2	3	4	5
9	Grup arkadaşlarımla birlikte işbirlikli öğrenme deneyimleri yaşamaktan hoşlanırım.	1	2	3	4	5
10	İşbirlikli öğrenmede, grupla çalıştığım için daha başarılı sonuçlar elde ettiğimi/edeceğimi düşünüyorum.	1	2	3	4	5
11	İşbirlikli öğrenmede grup arkadaşlarımla birlikte grup projesi ile ilgili problemleri çözmekten hoşlanırım.	1	2	3	4	5
12	İşbirlikli öğrenmede daha çok fikir ortaya çıkıyor.	1	2	3	4	5
13	Karmaşık problemlerin çözümüne yönelik düzenli planlar geliştirmede iyiyimdir.	1	2	3	4	5
14	Karmaşık problemleri çözmeye çalışmak eğlencelidir.	1	2	3	4	5
15	Zorlayıcı şeyler öğrenmeye istekliyimdir.	1	2	3	4	5
16	Elimdeki seçenekleri karşılaştırırken ve karar verirken kullandığım sistematik bir yöntem vardır.	1	2	3	4	5
17	Problemin çözümünü zihnimde canlandırma konusunda sıkıntı yaşarım.	1	2	3	4	5
18	Problem çözümünde X, Y gibi değişkenleri nerede ve nasıl kullanmam gerektiği konusunda sıkıntı yaşarım.	1	2	3	4	5
19	Tasarladığım çözüm yollarını sırasıyla aşamalı bir şekilde uygulayamam	1	2	3	4	5
20	Bir soruna yönelik olası çözüm yollarını düşünürken çok fazla seçenek üretemem.	1	2	3	4	5
21	İşbirlikli öğrenme ortamında kendi düşüncelerimi geliştiremem.	1	2	3	4	5
22	İşbirlikli öğrenme grup arkadaşlarıma bir şeyler öğretmeye çalışmak beni yoruyor.	1	2	3	4	5

EK-18. Öğretmen Gözlem Notları

Hafte 1: Öğrencilere serbestçe ilgili bilgi verildi. En test uygulandı. Öğrencilerin heyecanlı oldukları gözlemlendi. Derney grubuna toxicode, kontrol grubuna google blockly arayüzleri gösterildi. Öğrencilerin arayüzü basit bir şekilde zorlanmada kullandıkları ve kayıtlı oldukları gözlemlendi.

Hafte 2: Öğrenciler kendi hürlerinde arayüzde ilerleme göstermekte ve aramaları kolay bir şekilde görmektedir. Basitçe zara giden etkinlikleri basitlikleri içinde mutlu oldukları gözlemlenmektedir.

Hafte 3: Toxicode etkinliklerinin öğrencilere tekrar etmesi için test vermesi ve kendi öğrenme hürlerinde ilerleme imkanı verdiği gözlemlenmektedir.

Hafte 4: Son test uygulanıyor, öğrencilerin kayıtlı oldukları ve geliştirilen süreçte gelişimlerine katkıları bulunduğları için mutlu oldukları gözlemlendi.

ÖZGEÇMİŞ

Adı-Soyadı : Bülent KOÇAK
 Yabancı Dil : İngilizce
 Eğitim Geçmişi : Hacettepe Üniversitesi, Eğitim Fakültesi, Bilgisayar ve Öğretim Teknolojileri Lisans Programı (1999-2003)

Mesleki Geçmişi:

2003-2008 , ARTVİN/HOPA/Kemalpaşa Pansiyonlu İlköğretim/Bilgisayar Formatör Öğretmeni
 2008-2009 , SAMSUN / İLKADIM / Atatürk Anadolu Lisesi / Bilgisayar Formatör Öğretmeni
 2009-2015 , SAMSUN / BAFRA / Mustafa Kemal Ortaokulu / Bilgisayar Formatör Öğretmeni
 2015-2016 , SAMSUN / BAFRA / İlçe Milli Eğitim Müdürlüğü / Eğitici Bilgisayar Formatör Öğretmeni
 2016-2017 , MEB / Yenilik ve Eğitim Teknolojileri Genel Müdürlüğü / Öğretmen
 2017-2018 , SAMSUN İl Milli Eğitim Müdürlüğü / Öğretmen
 2018-2019 , MEB / Öğretmen Yetiştirme ve Geliştirme Genel Müdürlüğü / Şube Müdürü
 2019-2020 , MEB / Özel Kalem Müdürlüğü / Öğretmen
 2020-2021 , MEB / Özel Kalem Müdürlüğü / Şube Müdürü
 2021-2023 , MEB / Temel Eğitim Genel Müdürlüğü / Şube Müdürü

Akademik Çalışmalar (Bildiriler) :

2015, *Board Unleashed For Fatih / Yazılım*, Eğitimde FATİH Projesi Eğitim Teknolojileri Zirvesi, Ankara
 2018, *Rekabet Değil Birlikte Gelişim / Yazılım*, Eğitimde FATİH Projesi Eğitim Teknolojileri Zirvesi, Ankara
 2023, *Toxicode Uygulamasının Bilgi İşlemsel Düşünme Becerisine ve Algoritma Geliştirmeye İlişkin Öz Yeterlik Algısına Etkisi*, International Eurasian Educational Research Congress. Ankara
 2023, *İlçe Milli Eğitim Müdürlüklerinin Yaptığı Toplantılar Hakkında Okul Müdürlerinin Görüşlerinin İncelenmesi: Nitel Bir Çalışma*, Edu Congress. Ankara