MANİSA CELAL BAYAR ÜNİVERSİTESİ HASAN FERDİ TURGUTLU TEKNOLOJİ FAKÜLTESİ

DFA(Deterministic Finite Automaton)

Similation

Biçimsel Diller ve Otomata

**Hazırlayanlar:**

**Yiğit Karakoyun– 152804012**

**Onat AKTAŞ – 152804004**

**Canberk ATEŞ - 162802043**

**5/14/2018**

Biçimsel diller ve otomata dersi için DFA similasyonu projesi ilgili veri yapılarının implementesini ve temel olaylarının tektiklenmesi karşısında gösterdikleri davranışların tanımlanması sonucu ortaya çıkartılan benzetime ait rapordur.

**İçindekiler**

[1. **Önsöz** 2](#_Toc26378)

[1.1. AMAÇ(PURPOSE) 2](#_Toc26379)

[1.2. KAPSAM(SCOPE) 2](#_Toc26380)

[1.3. HEDEF KİTLESİ(INTENDED AUDIENCE) 2](#_Toc26381)

[1.4. SÖZLÜK(GLOSSARY) 2](#_Toc26382)

[1.5. Teknolojiler 2](#_Toc26383)

[1.5.1. Paketler 3](#_Toc26384)

[1.6. Kaynaklar 3](#_Toc26385)

[1.7. Görev Paylaşımı 3](#_Toc26386)

[1.8. Doküman Genel Bakışı 3](#_Toc26387)

[2. **Gereksinimler** 3](#_Toc26388)

[3. **UML Diyagramları** 4](#_Toc26389)

[3.1 Genel Bakış: 4](#_Toc26390)

[3.2 Sınıflar, Arayüzler ve İlişkiler: 5](#_Toc26391)

[3.2.1 LinkedList Soyut Sınıfı Ve İlişkileri 5](#_Toc26392)

[3.2.2 Node Sınıfı ve İlişkileri 6](#_Toc26393)

[3.2.3 Car Sınıfı ve İlişkileri 6](#_Toc26394)

[3.2.4 FIFOList Sınıf ve İlişkileri 7](#_Toc26395)

[3.2.5 PQList Sınıfı ve İlişkileri 7](#_Toc26396)

[3.2.6 IQueue Arayüzü 7](#_Toc26397)

[4. **Ekran Görüntüleri** 8](#_Toc26398)

[4.1 Hoşgeldiniz! 8](#_Toc26399)

[4.2 Comparison Of Queue Types 8](#_Toc26400)

[4.2.1 FIFO ve PRIORITY Park 9](#_Toc26401)

[4.2.2 Kontrol Düğmeleri 9](#_Toc26402)

[4.2.3 Giren/Çıkan Araç Bilgileri 10](#_Toc26403)

[4.2.4 Bilgi Tablosu 10](#_Toc26404)

[5. **Spesifik Çözümler** 11](#_Toc26405)

[5.1 ImageList Control 11](#_Toc26406)

[5.2 ListView Control 11](#_Toc26407)

[5.3 PQList Insert() Metodu 12](#_Toc26408)

[5.4 Araçların Çıkış Sürelerinin Hesaplanması ve Eklenmesi 13](#_Toc26409)

[5.5 MaterialListView Control ve AracSil() Metodu 15](#_Toc26410)

# Önsöz

## AMAÇ(PURPOSE)

Veri Yapıları dersi için hazırlanan bu doküman dersi alan öğrencilerin Veri Yapılarının özelliklerini daha etkin ve efektif bir şekilde kavrayabilmesini amaçlamaktadır. Ayrıca bu doküman ilgili yazılımın meslektaşlar, kullanıcılar ve/veya jüriler tarafından anlaşılması için hazırlanmıştır.

## KAPSAM(SCOPE)

Bu doküman ilgili yazılım ele alındığında, geliştiriciler tarafından kullanılan ve açıklaması yapılan, mimari, alt sistemler, arayüzler, algoritma tasarımları ve UML diagramlarının açıklamalarını içerir.

## HEDEF KİTLESİ(INTENDED AUDIENCE)

Bu doküman başta Manisa Celal Bayar Üniversitesi’nin yazılımcılarına, teknik ve akademik kadrosuna ve öğrencilerine hitaben hazırlanmıştır.

## SÖZLÜK(GLOSSARY)

|  |  |
| --- | --- |
| **Terim(Term)** | **Açıklama(Description)** |
| Otomasyon | Otomasyon, bir işin insan ile makine arasında paylaşılmasıdır. |
| Otopark | Otopark, motorlu taşıtların belli bir süre için bırakılabildiği yer. |
| Araba | Araba, yolcu ve yük taşımaya uygun tekerlekli, motorlu veya motorsuz hareket edebilen her türlü kara ulaşım taşıtı. Motorsuz olanlar hayvanlarla ya da insanlar tarafından yürütülür. |
| Kuyruk(Queue) | Kuyruk, aynı çizgi üzerinde yan yana ya da art arda dizilmiş olan şeylerin ya da kimselerin tümü, ilk girenin ilk çıktığı veri yapısı türü. |
| Öncelikli Kuyruk  (Priority Queue) | Standart kuyruk yapısına nazaran, gelen verilerin sıralanırken bir öncelik değerine göre azdan çoka ya da çoktan aza olacak şeklinde oluşturulan kuyruk yapısına denir. |
| FIFO(First in First Out) | Kısaca Kuyruk yapısının temel ilkesidir. İlk giren verinin ilk çıkaması şeklinde süre gelen işlem yapısına denir. |
| Bağlantılı Liste (Linked List) | Diziler gibi Bağlantılı Liste, doğrusal bir veri yapısıdır. Dizilerin aksine, bağlantılı liste elemanları bitişik konumda depolanmaz; elemanlar işaretçiler kullanılarak bağlanır. |
| Düğüm(Node) | Bağlı listelerde bulunan her bir elemana verilen isim olmakla beraber genelde basit bir şekilde veri(data) bilgisi ve sonraki elemanın konumu(next) olmak üzere iki temel bilgiyi barındıran elemandır. |

## Teknolojiler

* Visual Studio Entriprise Edition 2017
* .NET Framwork 4.7.1
* Unity 2018

### Paketler

* Herhangi bir paket kullanılmadı.

## Kaynaklar

* <https://docs.unity3d.com>
* <https://answers.unity.com>
* <http://forum.unity3dtr.com>
* <https://stackoverflow.com>

## Görev Paylaşımı

Onat Aktaş – **152804004**

Back-End Developer, arayüz-sınıf implementasyonlarının tanımlanması ve işlevselliğinin eklenmesi görevlerini üstlenmiştir.

Front-End Developer, kullanıcı arayüz tasarımı, implementasyonu.

Proje dökümanının hazırlanması ve görev paylaşımı.

Canberk Ateş – **162802043**

Proje Dökümanın ve raporun hazırlanması.

Tester, Unit, Integration ve Acceptance Testlerinin gerçekleştiricisi

Yiğit Karakoyun – **152804012**

Back-End Developer, sınıfların tasarımı ve implementasyonu.

Tester, Unit, Integration ve Acceptance Testlerinin gerçekleştiricisi.

## Doküman Genel Bakışı

Dökümanda bulunan sonraki bölümde DFA Similasyonu üzerinde yapılacak işlemlerin analiz ve tasarım aşamalarını içerecektir. Analiz sonucunda sisteme uygulanacak ekran tasarımlarının hazırlanışı bulunmaktadır.

Üçüncü bölümde analiz aşaması sonucu ortaya çıkarılan taslak ve planların sisteme implemente edilişlerini bulabilirsiniz.

# Gereksinimler

**Proje Metni:**

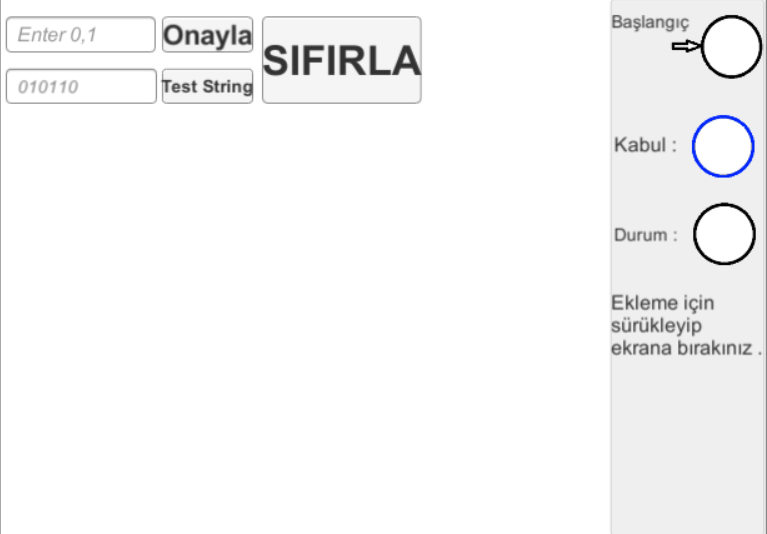
Belirli Sonlu Durum Otomatı Simülatörü

## Sınıflar ve İlişkiler:

### LinkedList Soyut Sınıfı Ve İlişkileri

# Ekran Görüntüleri

## Giriş Ekranı



Giriş ekranımız Defaul biçimde kullanıcıları karşılamaktadır.

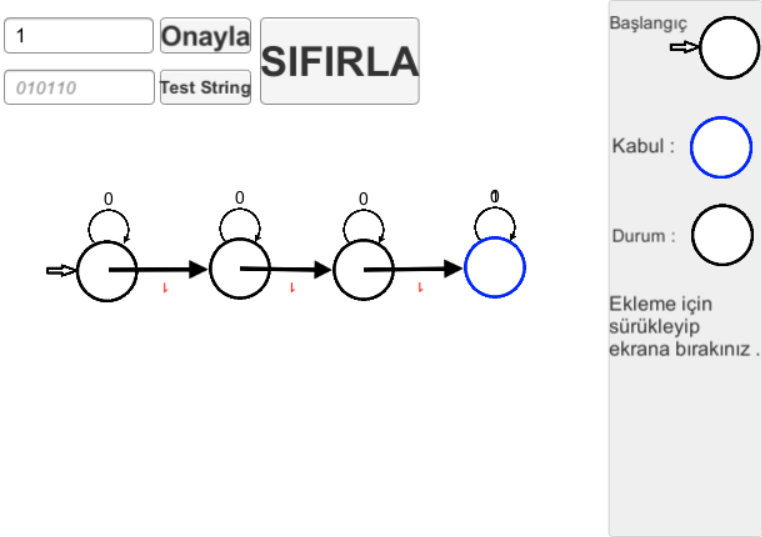
## Sağ Kontrol Paneli



Kontrol panelimizin kullanımı Drag and Drop yapısı ile çalışmaktadır panel içerisinde ki State’leri sürükle bırak ile DFA tasarımını oluşturabiliriz.

### Drag and Drop DFA Tasarımı

Sürükle bırak ile gerçekleştirdiğimiz bir DFA tasarımı.



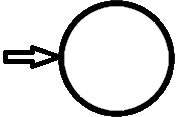
### DFA Durum Geçişleri



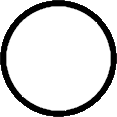
Durumlar arası Geçişler Click-Base ile yapılmaktadır durumları birbirine bağlamak için Sol üst paneled belirdiğimiz dil geçerlidir daha sonrasında First Click ile Second Click ile durumları birbirlerine ve kendisine bağlayabilmekteyiz.

### Kontrol Düğmeleri

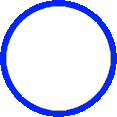
Başlangıç Durumunu simgelemektedir.



Default durumu simgelemektedir.



Kabul durumunu simgelemektedir.



Geçişlerdeki dili belirlemek için kullandığımız İnput.



Girilen String’I oluşturduğumuz DFA üzerinde Test etmemiz için kullanılan İnput.



Ekranımızı sıfırlamamıza yarayan butondur.



# Spesifik Çözümler

## Durum Drag And Drop

Proje içersinde ki durumların ekran üzerinde tasarımının serbest bir şekilde yapılması için Unity’nin using UnityEngine.EventSystems; kütüphanesini kullandık bununla beraber Drag and drop sistemi ile IBeginDragHandler, IDragHandler, IEndDragHandler interfacelerini kullanarak durumlarımızı ekranda istediğimiz şekilde konumlandırma sistemini yaptık.

public void OnBeginDrag(PointerEventData eventData)//Nesnemiz sürüklemeye başladığında

{

foreach (var prefab in manager.statePrefabs)

{

if (prefab.name.Equals(gameObject.name))

{

currentPrefab = Instantiate(prefab);

currentPrefab.transform.parent = manager.mainCanvas.transform;

}

}

if(currentPrefab == null)

{

return;

}

dragGameobject = gameObject;

startPosition = currentPrefab.transform.position;

}

public void OnDrag(PointerEventData eventData)//Nesnemiz sürüklenme halindeyken.

{

if (currentPrefab == null)

{

return;

}

currentPrefab.transform.position = Input.mousePosition;

}

public void OnEndDrag(PointerEventData eventData) // Sürükleme işlemi bittiğinde.

{

dragGameobject = null;

// currentPrefab.transform.position = startPosition;

}

## Durum geçişleri

Durum geçişlerini Click-base olarak yaptık rahat bir şekilde kullanıcı tarafından tasarlanması için bunun içinde unity’nin IPointerClickHandler interface’ni kullandık.

public void OnPointerClick(PointerEventData eventData)

{

if (manager.SecondClick != null)

{

manager.FirstClick = null;

manager.SecondClick = null;

}

if (manager.FirstClick == null)

{

manager.FirstClick = gameObject;

Debug.Log("First : "+ manager.FirstClick.name);

}

else

{

manager.SecondClick = gameObject;

if (manager.FirstClick.GetInstanceID() == manager.SecondClick.GetInstanceID())

{

Debug.Log("Aynı Stateler");

manager.kayitlarManager.EkleKayit(gameObject, manager.Language, gameObject);

DrawSelfArrow(gameObject);

return;

}

Debug.Log("Second :"+ manager.SecondClick.name);

DrawArrrow(manager.FirstClick, manager.SecondClick);

manager.kayitlarManager.EkleKayit(manager.FirstClick, manager.Language, manager.SecondClick);

}

}

Click event’ini control ettiğimiz yapımız.

Arrowları çizdirdiğimiz metotlar.

public void DrawArrrow(GameObject CenterObject,GameObject TargetObject)

{

Nesne CenterNesne = CenterObject.GetComponent<Nesne>();

Nesne TargetNesne = TargetObject.GetComponent<Nesne>();

float Distance = 0;

Distance = Vector2.Distance(CenterObject.GetComponent<RectTransform>().transform.localPosition, TargetObject.GetComponent<RectTransform>().transform.localPosition);

//Aralarında ki Mesafeyi hesaplama.

//AÇI Hesaplama

Vector3 dir = TargetObject.transform.position - CenterObject.transform.position;

dir = TargetObject.transform.InverseTransformDirection(dir);

float angle = Mathf.Atan2(dir.y, dir.x) \* Mathf.Rad2Deg;

currentArrowprefab = Instantiate(manager.arrowPrefab);

currentArrowprefab.transform.parent = CenterObject.transform;

currentArrowprefab.GetComponent<RectTransform>().transform.localPosition = new Vector3(0, 0, 0);//Posizyon ayarlama

currentArrowprefab.GetComponent<RectTransform>().transform.Rotate(new Vector3(0, 0, angle-90));//Okun Açısını ayarlama

currentArrowprefab.GetComponent<RectTransform>().sizeDelta = new Vector2(16f, Distance-25);//Okun Uzunluğunu ayarlama

currentArrowprefab.transform.GetChild(0).GetComponent<Text>().text = manager.Language;

}

public void DrawSelfArrow(GameObject CenterObject)

{

currentSelfArrow = Instantiate(manager.selfArrow);

currentSelfArrow.transform.parent = CenterObject.transform;

currentSelfArrow.GetComponent<RectTransform>().transform.localPosition = new Vector3(0,32, 0);

currentSelfArrow.transform.GetChild(0).GetComponent<Text>().text = manager.Language;

}

## PQList Insert() Metodu

## Araçların Çıkış Sürelerinin Hesaplanması ve Eklenmesi

## MaterialListView Control ve AracSil() Metodu