

PIND

Oyun Tasarım Belgesi

Prof. Dr. Hakan TÜZÜN

Proje Danışmanı

İrem Özden

2019

# İÇİNDEKİLER

[1. GİRİŞ](#_Toc396861082) 3

[1.1. Oyunun Kısa Özeti ve Konsepti](#_Toc396861082) 3

[1.2. Benzer Oyun Türleri](#_Toc396861082) 3

[2. Oynanış ve Mekanikler 4](#_Toc396861086)

[2.1. Oynanış 4](#_Toc396861087)

[2.2. Mekanikler 4](#_Toc396861090)

[2.2.1. Fizik 4](#_Toc396861091)

[2.3. Oyun Nesneleri](#_Toc396861094) 5

[2.3.1. Hacim Kutuları](#_Toc396861111) 5

[2.3.2.Oyun Modu Ayarları](#_Toc396861112) 5

[2.4. Karakterler](#_Toc396861116) 7

[2.4.1. Aktivist](#_Toc396861116) 8

[2.4.2. Ofis Çalışanı](#_Toc396861116) 8

[2.5. Korku Seviyesi Çubuğu 1](#_Toc396861116)0

[2.6. Tuzaklar 1](#_Toc396861116)1

[2.6.1. Water](#_Toc396861116) 11

2.6.2. [Explosion 1](#_Toc396861127)4

[2.6.3. Burn](#_Toc396861127) 17

3. Materyaller ve Yöntem 20

3.1. Modelleme ………………………………………………………………………………………………………………………………..20

3.2.Işık ve Kamera……………………………………………………………………………………………………………………………..20

3.3.Görsel Efekt………………………………………………………………………………………………………………………………. 21

3.4. Ses…………………………………………………………………………………………………………………………………………… 22

3.5. Kodlama…………………………………………………………………………………………………………………………………… 23

[4. İhtiyaç Duyulan Araçlar](#_Toc396861129) 23

1. GİRİŞ

## 1.1. Oyunun Kısa Özeti ve Konsepti

## Tek oyunculu 3 boyutlu bir oyundur. Oyunda, bir doğa aktivisti ünlü bir kozmetik firmasının merkez ofisine gidip basın organlarına sızdırmak için belgeler çalmak istiyor. Bunun için ofisin tatil olduğu günü seçiyor ancak bir ofis çalışanı izne çıkmamış olduğunu görüyor. Ofis çalışanını korkutup kaçırmak için ofiste çeşitli tuzaklar hazırlıyor.

## Düşük poligonlu modeller içeren ofis ortamında geçen bir oyundur. Ofis içerisinde çok fazla nesne bulunduğu için kalabalık ve yorucu bir görüntü oluşturmamak adına düşük poligonlu olması tercih edilmiştir. Çizgi film tarzı modeller kullanılmıştır.

## Ana karakterimiz oyundaki rolü gereği haylaz bir görünüme sahiptir. Ofis çalışanı için ise oldukça sıradan bir görünüm seçilmiştir.

## Tür: Macera-Aksiyon

## Hedef Kitle: 13+

## 1.2. Benzer Oyun Türleri

Bu oyunda “Neighbours from Hell” isimli oyundan esinlenilmiştir. Strateji-Macera türünde2003 yılında piyasaya çıkmış 2 boyutlu bir oyundur. Basit bir oynanışa, esprili bir tarza sahiptir.

## 1.3. Proje Kapsamı

Projenin ilk adımı olarak oyun fikri üzerinde araştırma yapılmıştır. Fikrin oluşması için çeşitli oyun türleri incelenmiştir. Oyun tasarımında kullanılacak modeller, sesler için *unreal engine marketplace* ve *freesound.org* kütüphanelerinin incelemesi yapılmıştır.

Oyun fikri netleştirildikten sonra fokus dokümanı hazırlanmıştır. Bu dokümanda oyununun hikayesi ve oynanışı üzerinde durulmuştur.

Focus dokümanı ile örtüşecek şekilde modeller ve oyun sahnesinde yer alacak objeler temin edilmiştir. Oyun geliştirme aracı olarak Unreal Oyun Motoru kullanılmasına karar verilmiştir. Bu oyun motorunun seçilmesindeki en önemli etken blueprint kullanımına olanak vermesidir.

# 2. Oynanış ve Mekanikler

## 2.1. Oynanış

### Oyuncu oyuna ofis ortasında başlar. Oyuncunun hareket alanı ofisle sınırlıdır. Kamera oyuncunun arkasında yer alır. Klavyedeki yön tuşları ile karakterin hareket komutları verilir. Farenin sol tuşuna basılı haldeyken fare hareket ettirilerek kameranın görüş açısı değiştirilebilir. Oyuncunun hareketiyle birlikte kamera da hareket eder. Oyunda ana karakterin haricinde bir de yapay zeka karakter (AI) bulunmaktadır. AI hareketleri oyuncu tarafından kontrol edilemez. AI belirli bir rota çizerek ofis içerisinde hareket eder. Bu rota sayesinde ofis çalışanının periyodik olarak belirli noktalara uğramasını sağlar. Oyuncu, ofis alışanının hareketlerini gözlemleyerek uğradığı noktaları seçer ve oraya giderek yerleştirilmiş tuzakları aktif hale getirir. Oyuncu tuzak olan noktaya geldiğinde oyun içi sesler yardımıyla gittiği noktada yapay zekaya karşı aktif edebileceği bir tuzak olup olmadığını fark eder. Örnek vermek gerekirse patlama tuzağı kuracağı fotokopi makinesinin önünde durduğunda fotokopi makinesinin sesi gelir. Bu sayede ofis çalışanının takip ettiği rotada bu noktalara geldiğinde oyuncu tarafından aktif hale getirilmiş tuzakla karşılaşır. Yapay zekanın aktif haldeki tuzak alanına girmesi ile alanda bulunan tetikleyici işlem gerçekleşmiş olur. Bu tetiklenme sonrasında ofis çalışanı çığlık atarak kaçmaya başlar ve ekranda sol üst köşede bulunan ofis çalışanına ait korku seviyesi 10 birim artar. Yapay zekanın korku seviyesi toplamda 100 birime ulaştığında oyuncu oyunu kazanmış olur.

## 2.2. Mekanikler

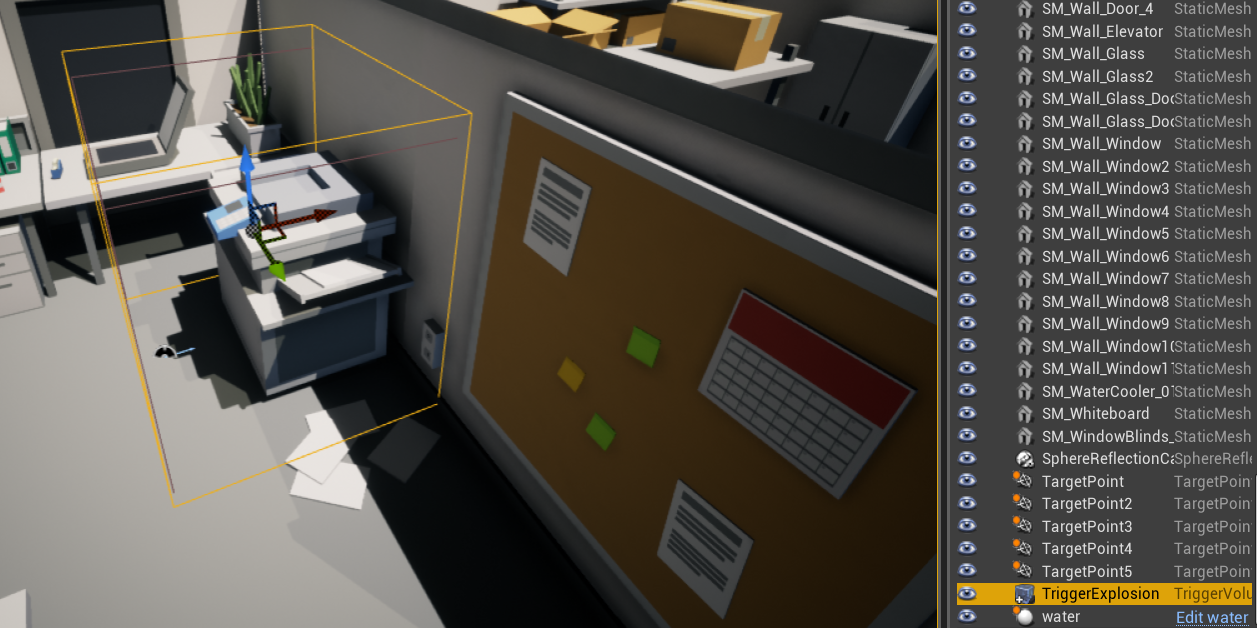
### 2.2.1. Fizik

Objelerin birbiri içerisinden geçmemeleri için unreal oyun motorunda yer alan çarpışma (collision) özellikleri incelendi. Bu doğrultuda sahne içinde kullanılacak hareketsiz nesnelerin kafeslerinin (static mesh) sınırlarına karar verildi. Hareketli nesneler olan oyun karakterlerinin çarpışma sınırlarını belirleyen kapsüllerin boyutlandırılması da nesnelerle temasları halinde iç içe geçmeyecek şekilde ayarlandı.

## 2.3. Oyun Nesneleri

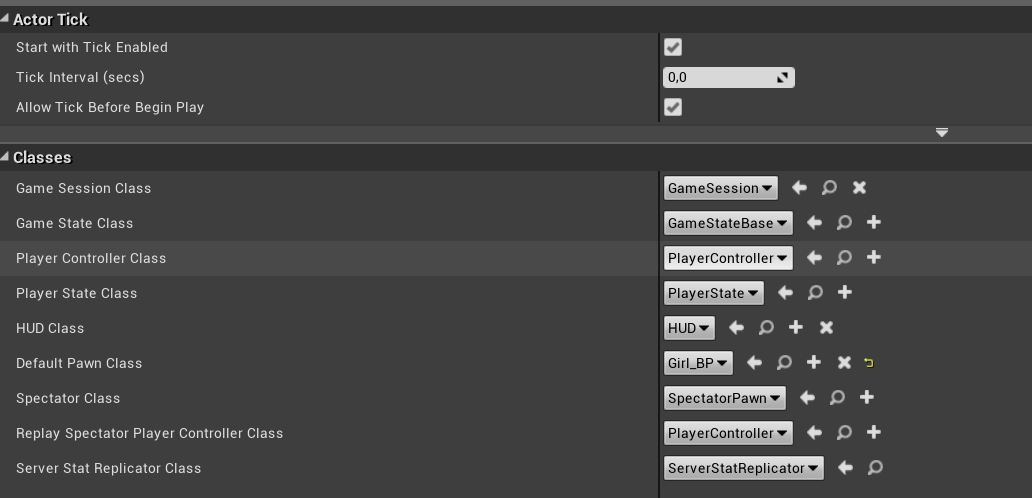
### 2.3.1 Hacim Kutuları

Oyundaki tuzakların oluşturulması için kullanılan en temel tetikleyici hacim kutularıdır. Bu hacim kutuları ile karakterlerin sahip olduğu kapsül alanlarının kesişmesi durumunda ilgili blueprint komutları gerçekleşir. Oyun karakterlerinden birisi hacim kutusu içerisinde iken blueprint komutunun çalışıp çalışmayacağı ise boolean olarak tanımlanmış değişkenlerin kontrolü ile sağlanır.



**2.3.2 Oyun Modu Ayarları**

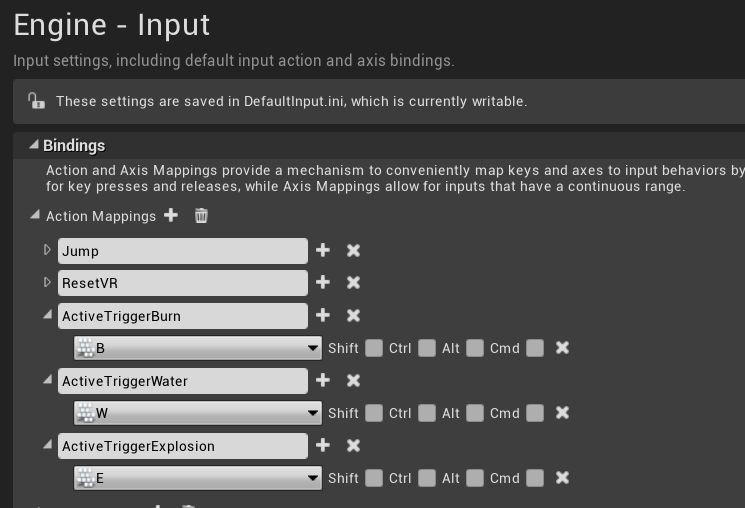
Oyuncunun hangi karakteri kontrol edeceğini belirlememiz gerekir. Oyun modu ayarları bu doğrultuda yapılmalıdır.



Oyunda karakterlerin özellikle de yapay zekanın hareket alanlarının belirlenebilmesi için işleme hacmi yaratılması gerekir.

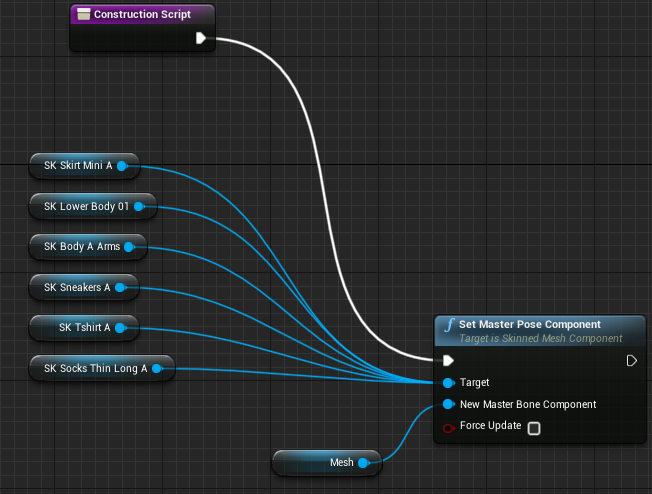


Oyunda oyuncunun girdi yapabileceği (input) tuşlar belirlenmelidir. Burada oyuncunun hareketini sağlayan sağ, sol, ileri yön tuşları tanımlanmalıdır. Ayrıca oyunun temelini oluşturan tuzakları aktif hale getirmesini sağlayan girdiler tanımlanmalıdır.



### 2.4. Karakterler

Oyundaki ana karakterin ve yapay zekanın özelliklerinin character class oluşturarak belirtilmesi gerekir. Burada karakter için kapsül oluşturularak hiyerarşide altında yer alacak şekilde karakterin mesh görüntüsü eklenir. Mesh ile birlikte karaktere ait diğer ögeler de (kıyafetler, saç vs.) yine bu hiyerarşi altına eklenir. Bu ögelerin ilişkisini ve yapısını gösteren kodlama (construction script) yapılır.



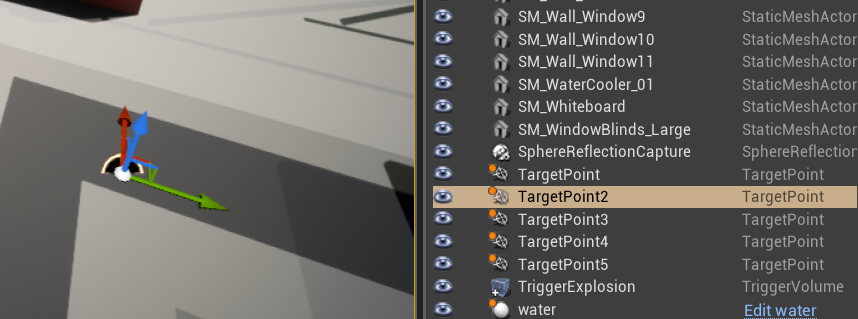
### 2.4.1 Aktivist

Kameranın karakteri arkadan takip edebilmesi için o da hiyerarşide kapsülün altında ye alacak şekilde blueprint objesine eklenmelidir.

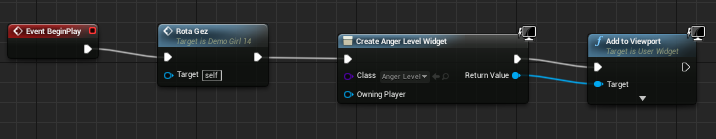


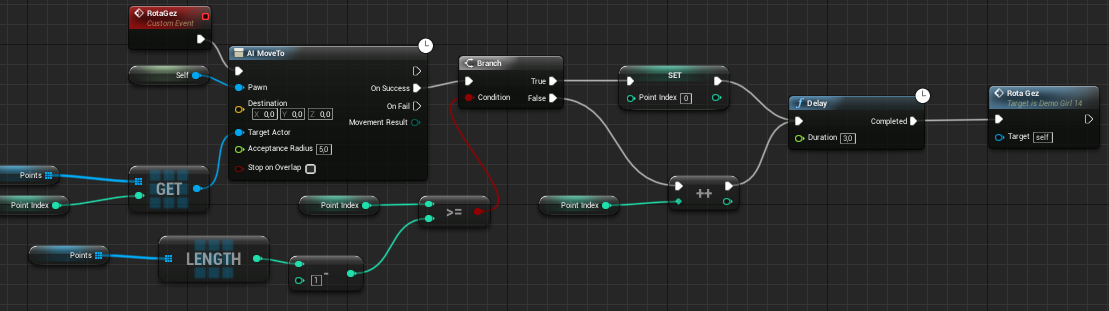
### 2.4.2 Ofis Çalışanı

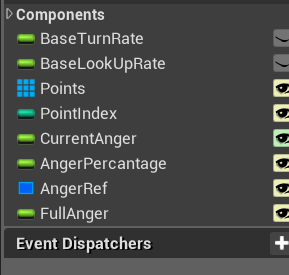
Ofis çalışanının odalar arasında belirli bir rotada gezebilmesi için öncelikle hedef noktalar seçilmelidir. Bunlar yapay zekanın varış noktaları olarak kullanılacaktır. Blueprint içinde oluşturulan olayda (RotaGez – Custom Event) yapay zekanın takip edeceği rota bu hedef noktalar arasında oluşturulacaktır. Bu noktalardan bir diziliş (array) oluşturulması, noktaların blueprint içerisinde kullanılabilmesine olanak sağlar.



Burada *Move To* fonsiyonu kullanılarak yapay zekanın yapacağı eylem belirlenir. *Get* fonksiyonu yardımıyla hedef lokasyon seçilir. Fonksyonda her defasında point indeksi 1 puan arttırılarak ofis çalışanının ilk noktadan son noktaya kadar sırayla dolaşması sağlanır. Ofis çalışanı uğradığı her durakta belli bir sürede bekler (Delay). Bir sonraki durağına doğru ancak bekleme süresi bittiği zaman ilerler.

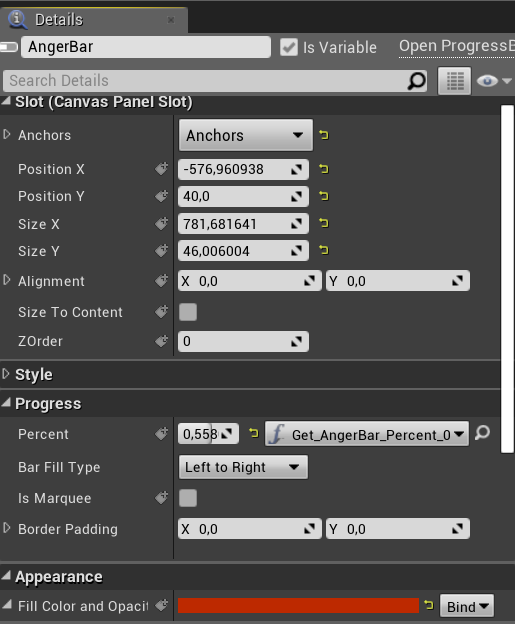
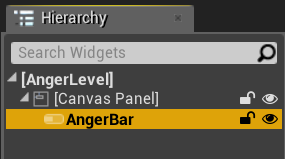






### 2.5. Korku Seviyesi Çubuğu

Ofis çalışanının korku seviyesinin arttığını oyuncunun görmesi gerekir. Bunun için ekranda görülecek bir ilerleme çubuğu kullanılır. Burada çubuk ekran üzerinde sabit olabilceği gibi ofis çalışanını temsil eden oyun ögesinin üzerinde hareketli halde de bulunabilir. Ben ekranda sabit olmasının görsel olarak daha az yorucu olduğunu düşündüğüm için bu şekilde kullanmaya karar verdim. Widget oluşturulup içerisine kanvas eklenir. Hiyerarşide bu kanvasın bir alt seviyesinde olacak şekilde bir progress bar (ilerleme çubuğu) oluşturulur. İlerleme çubuğunun özelliklerini oluşturan veriler girilir. Burada en mühim nokta fonksiyon olarak tanımlanmış olan yüzde değerinin ilerleme çubuğunda seçilmesidir.



### 2.6. Tuzaklar

Ofis çalışanının durak noktalarında 3 tuzak bulunur. Bunlara ait özellikler aşağıdaki tablo içerisinde belirtilmiştir. Tuzakların çalışma prensibi alt başlıklarda anlatılacaktır.

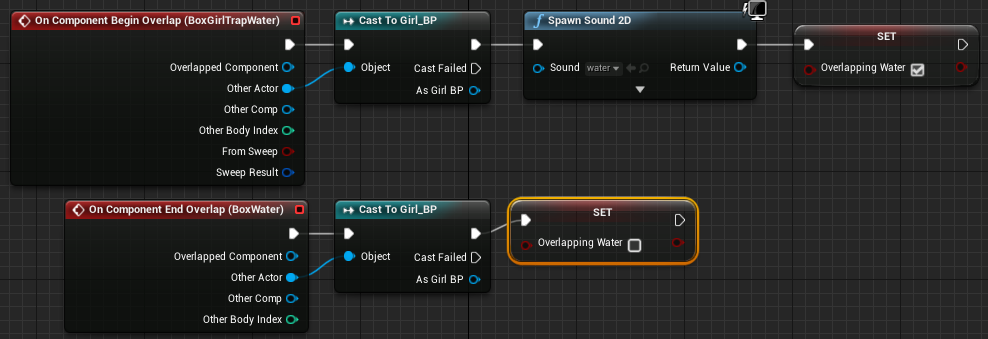
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tuzak Adı** | **Girdi** | **Yapay Zeka Hedef Nokta No** | **Renk** | **Değişkenler** | **Hacim Kutuları** | **Ses Dosyası** | **Görsel Efekt** |
| Water | W | 2 | mavi | TriggerOffWater overlappingWater | BoxWater BoxGirlTrapWater | water | p\_sparks |
| Explosion | E | 1 | kırmızı | TriggerOffExplosion overlappingExplosion | BoxExplosion BoxGirlTrapExplosion | copy-machine\_sound | p\_explosion |
| Burn | B | 3 | sarı | TriggerOffBurn overlappingBurn | BoxBurn BoxGirlTrapBurn | burning-paper | p\_fire |

### 2.6.1. Water

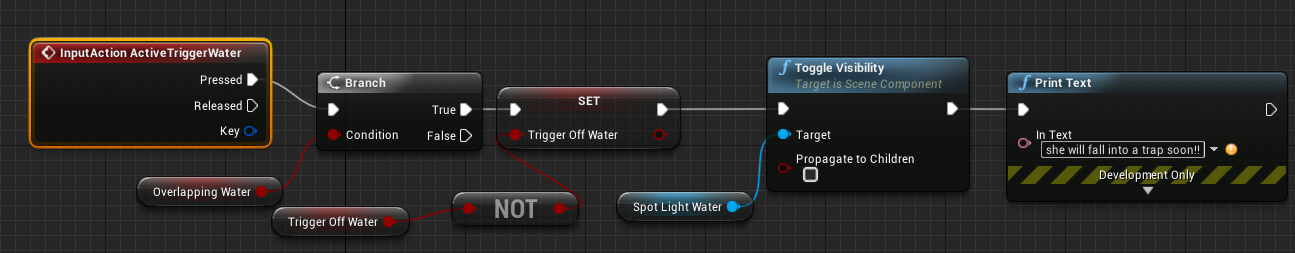


Bu tuzak ofis çalışanının ikinci durağı olan ofis tuvaletinde yer almaktadır. Oyuncu tuzağı aktif hale getirdiğinde üzerinde bulunan, oyun içerisinde görünmeyen spot ışığı mavi renkte yanmaktadır.

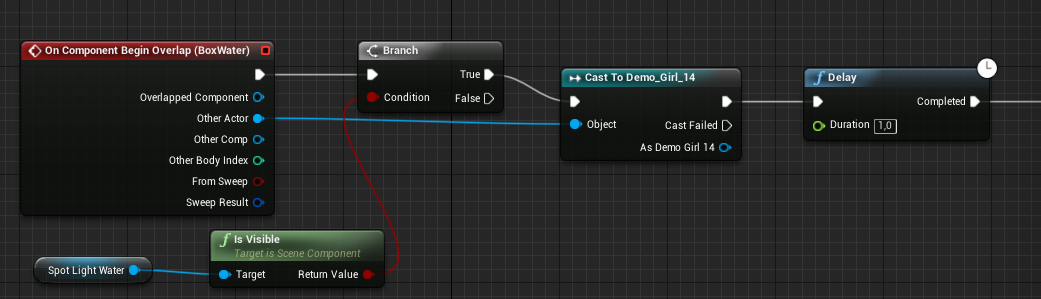
Tetikleyici hacim kutuları ile alan içerisinde oyuncunun olup olmadığı anlaşılır. Oyuncu alan içerisinde ise oyunda su sesi aktif hale gelir ve oyuncu burada tuzak olduğunu anlar. Bu esnada OverlappingWater isimli değişken yardımı ile oyuncunun alan içinde olup olmama durumu belirtilir.

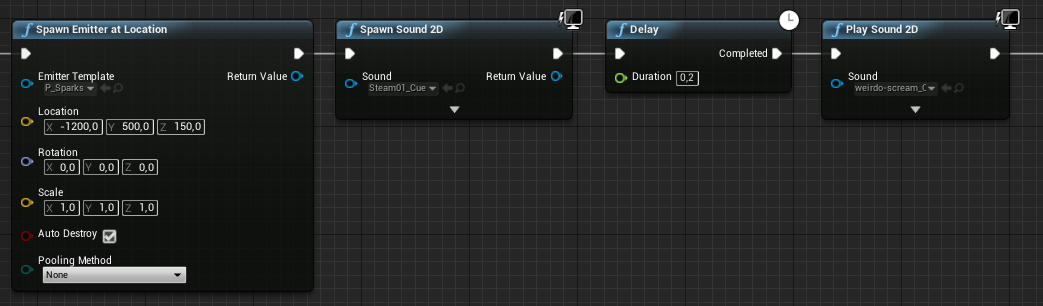


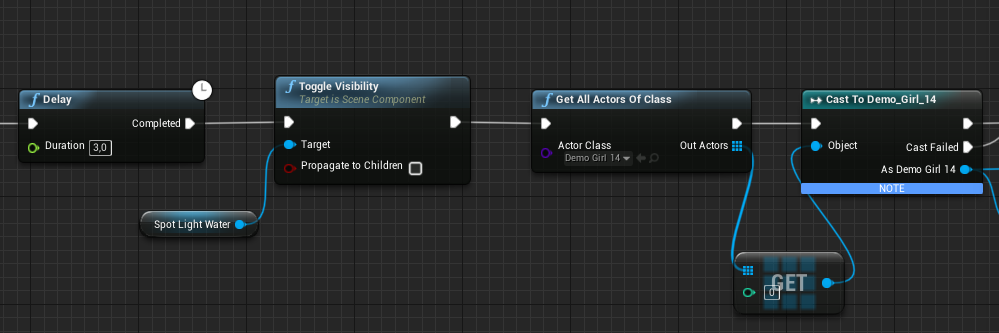
Oyuncu alan içerisinde ise OverlappingWater isimli değişken (boolean) doğru (1) olarak görülür. Doğru koşulu sağlanıyorsa oyuncu tuzağı aktif hale getirmek için W tuşu ile girdi sağlayabilir (inputActiveTriggerWater çalışır). Oyuncu tuzağı aktif hale getirdiğinde TriggerOffWater değişkeni yanlış (0) olarak işaretlenir. Bu sayede yeniden tuzak kurulacağı zaman mevcut değişkenin hali hazırda yanlış (0) olması sağlanır. Tuzağın aktif halde olduğunu oyuncuya hatırlatan mavi renkteki spot ışığı da görünür hale gelir.

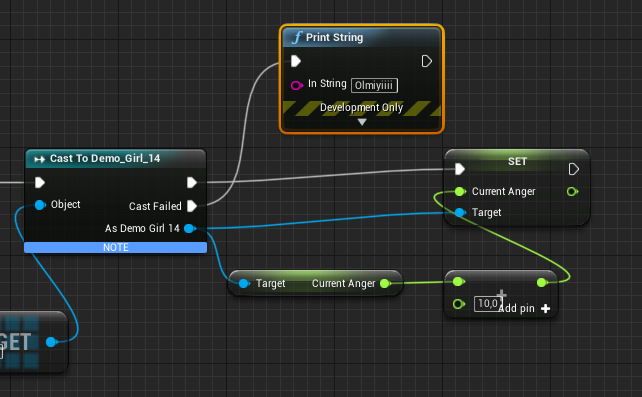


Ofis çalışanı (yapay zeka) BoxWater isimli hacim kutusunun kapsadığı alana girdiğinde aktif haldeki tuzak çalışır. Tuzağın aktif olup olmadığı kontrolü mavi renkteki spot ışığının görünür olup olmadığı durumuna bakılarak yapılır. Tuzak çalıştığında lavabodan sular fışkırdığını gösteren bir görsel efekt görürüz ve ofis çalışanı çığlık atar. Ofis çalışanının korku seviyesi 10 birim kadar artar.

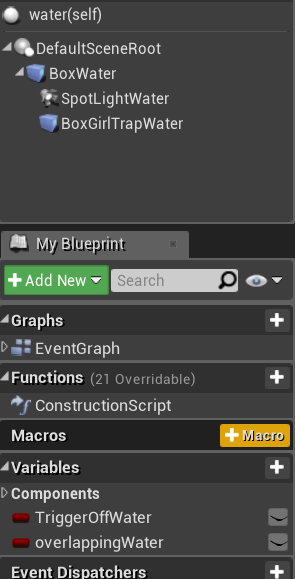








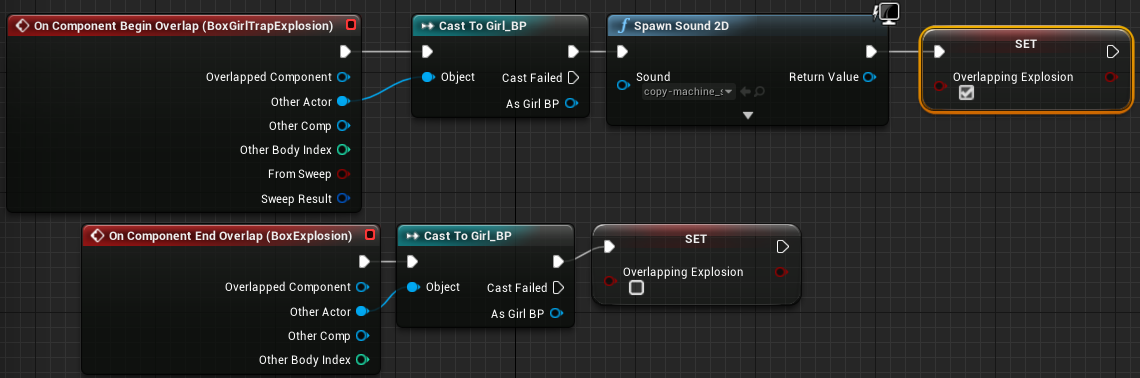
**Water Tuzak Değişkenleri:**



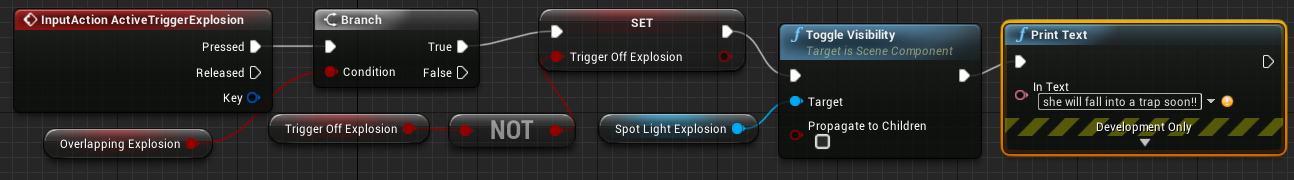
### 2.6.2. Explosion



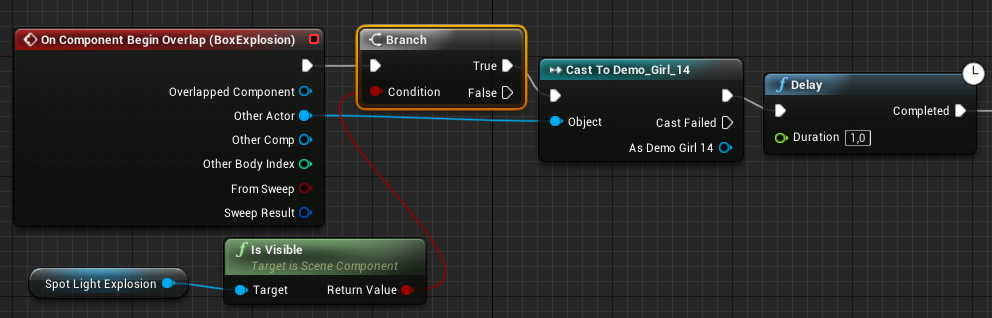
|  |
| --- |
| Bu tuzak ofis çalışanının ilk durağı olan açık ofis alanında yer almaktadır. Oyuncu tuzağı aktif hale getirdiğinde üzerinde bulunan, oyun içerisinde görünmeyen spot ışığı kırmızı renkte yanmaktadır. |
| Tetikleyici hacim kutuları ile alan içerisinde oyuncunun olup olmadığı anlaşılır. Oyuncu alan içerisinde ise oyunda fotokopi makinesi sesi aktif hale gelir ve oyuncu burada tuzak olduğunu anlar. Bu esnada OverlappingExplosion isimli değişken yardımı ile oyuncunun alan içinde olup olmama durumu belirtilir. |

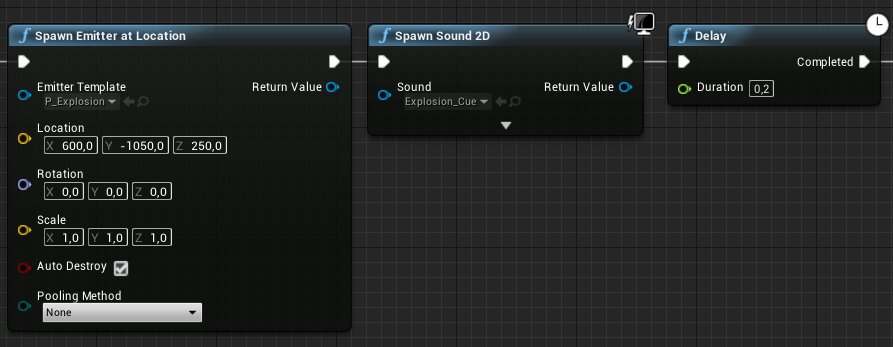


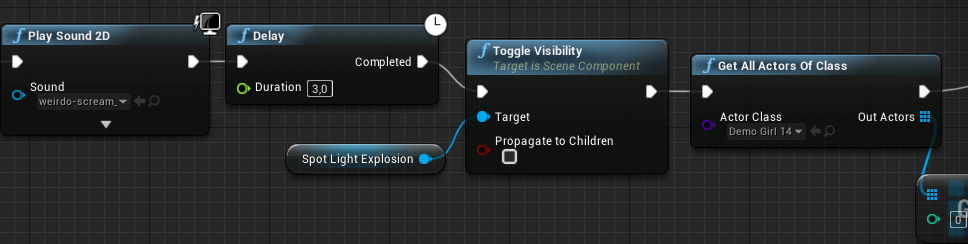
Oyuncu alan içerisinde ise OverlappingExplosion isimli değişken (boolean) doğru (1) olarak görülür. Doğru koşulu sağlanıyorsa oyuncu tuzağı aktif hale getirmek için E tuşu ile girdi sağlayabilir (inputActiveTriggerExplosion çalışır). Oyuncu tuzağı aktif hale getirdiğinde TriggerOffExplosion değişkeni yanlış (0) olarak işaretlenir. Bu sayede yeniden tuzak kurulacağı zaman mevcut değişkenin hali hazırda yanlış (0) olması sağlanır. Tuzağın aktif halde olduğunu oyuncuya hatırlatan kırmızı renkteki spot ışığı da görünür hale gelir.

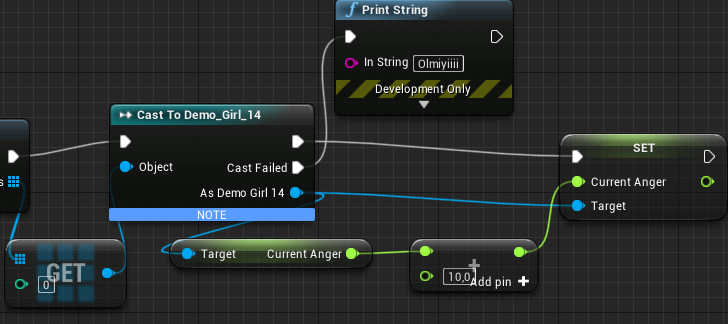


Ofis çalışanı (yapay zeka) BoxExplosion isimli hacim kutusunun kapsadığı alana girdiğinde aktif haldeki tuzak çalışır. Tuzağın aktif olup olmadığı kontrolü kırmızı renkteki spot ışığının görünür olup olmadığı durumuna bakılarak yapılır. Tuzak çalıştığında fotokopi makinesinin üst kısmının patladığını gösteren bir görsel efekt görürüz ve ofis çalışanı çığlık atar. Ofis çalışanının korku seviyesi 10 birim kadar artar.

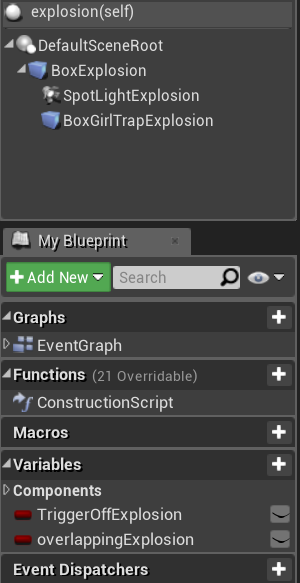








**Explosion Tuzak Değişkenleri:**

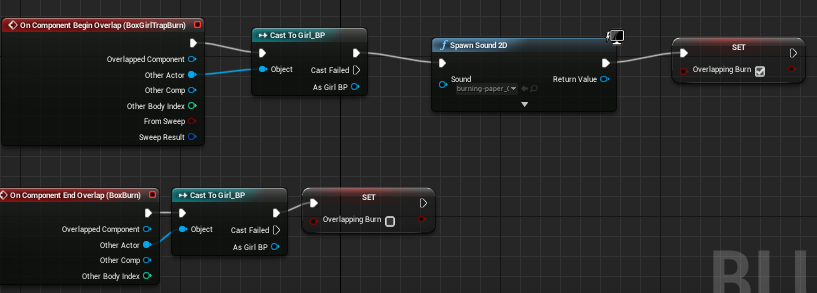


### 2.6.3. Burn

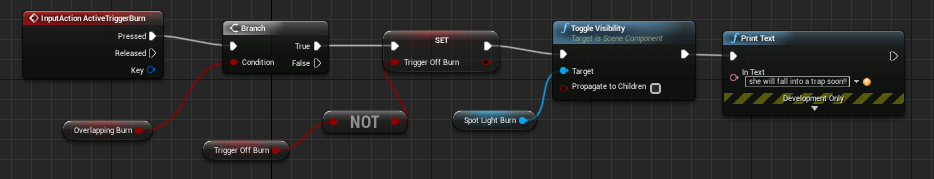


Bu tuzak ofis çalışanının üçüncü durağı olan toplantı odasında yer almaktadır. Oyuncu tuzağı aktif hale getirdiğinde üzerinde bulunan, oyun içerisinde görünmeyen spot ışığı sarı renkte yanmaktadır.

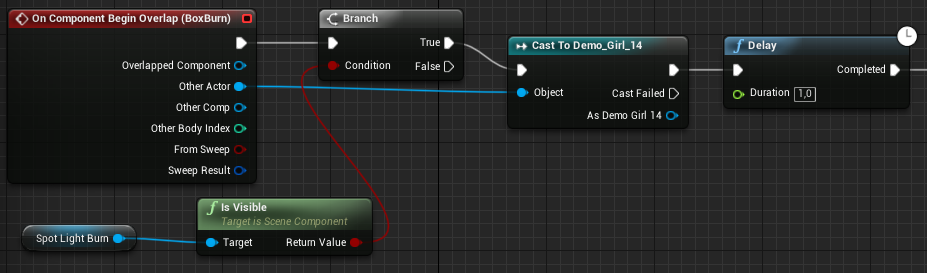
Tetikleyici hacim kutuları ile alan içerisinde oyuncunun olup olmadığı anlaşılır. Oyuncu alan içerisinde ise oyunda kağıt çıtırtısı sesi aktif hale gelir ve oyuncu burada tuzak olduğunu anlar. Bu esnada OverlappingBurn isimli değişken yardımı ile oyuncunun alan içinde olup olmama durumu belirtilir.

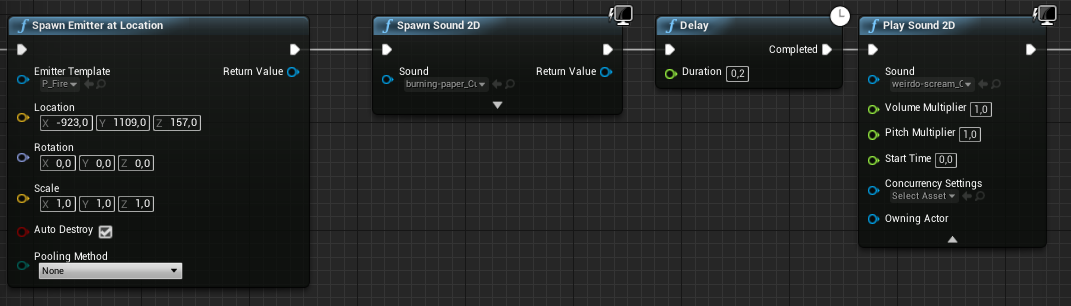


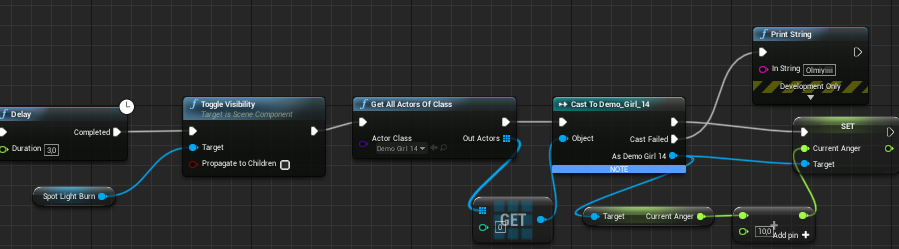
Oyuncu alan içerisinde ise OverlappingBurn isimli değişken (boolean) doğru (1) olarak görülür. Doğru koşulu sağlanıyorsa oyuncu tuzağı aktif hale getirmek için B tuşu ile girdi sağlayabilir (inputActiveTriggerBurn çalışır). Oyuncu tuzağı aktif hale getirdiğinde TriggerOffBurn değişkeni yanlış (0) olarak işaretlenir. Bu sayede yeniden tuzak kurulacağı zaman mevcut değişkenin hali hazırda yanlış (0) olması sağlanır. Tuzağın aktif halde olduğunu oyuncuya hatırlatan sarı renkteki spot ışığı da görünür hale gelir.



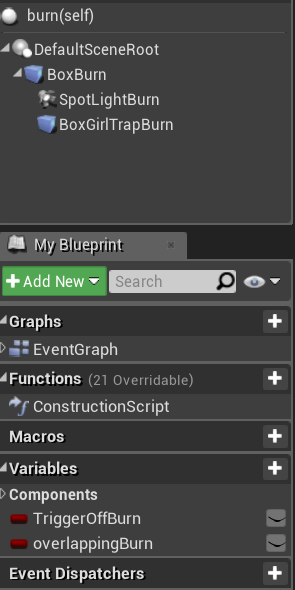
Ofis çalışanı (yapay zeka) BoxBurn isimli hacim kutusunun kapsadığı alana girdiğinde aktif haldeki tuzak çalışır. Tuzağın aktif olup olmadığı kontrolü sarı renkteki spot ışığının görünür olup olmadığı durumuna bakılarak yapılır. Tuzak çalıştığında kağıtların yandığını gösteren bir görsel efekt görürüz ve ofis çalışanı çığlık atar. Ofis çalışanının korku seviyesi 10 birim kadar artar.







**Burn Tuzak Değişkenleri:**



# 3. Materyaller ve Yöntem

## 3.1. Modelleme



Oyundaki karakterler için Epic Games mağazasından (asset store) iskelet ve uygun kıyafetleri kapsayan asset satın alındı. Örnek resimde görülen parça modeller kullanılarak karakterler blueprint ile oluşturuldu, oyunun stiline uygun olacak şekilde birleştirildi. Ofis modellerinin de karakterler ile stil bütünlüğünde olmasına önem verildi.

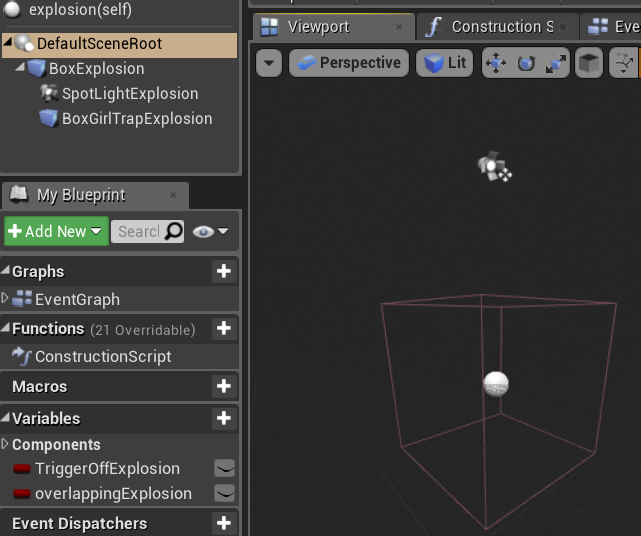
## 3.2. Işık ve Kamera



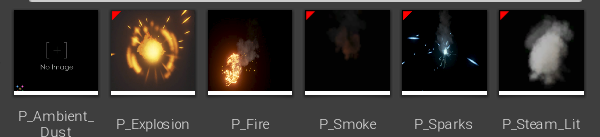
Unreal engine oyun motorunda bulunan SkyLight ve DirectionalLight nesneleri kullanılarak sahne ışığı ayarlanır. Sahneye ışık kaynakları ile birlikte Sphere Reflection Capture Actor isimli yansıtıcı objeler de eklenebilir. Kameranın oyuncunun arkasında hareket etmesi amaçlandığı için oyuncu için oluşturulmuş Girl\_BP isimli blueprint içerisine yerleştirilmiştir.



Ayrıca sahne içerisinde tuzakların aktif ve pasif durumda olduğunu belirten spot ışıkları bulunur. Bunlar da tuzaklar için oluşturulmuş blueprint sınıfları içerisinde hiyerarşide tetikleyici kutuların altında bulunur.



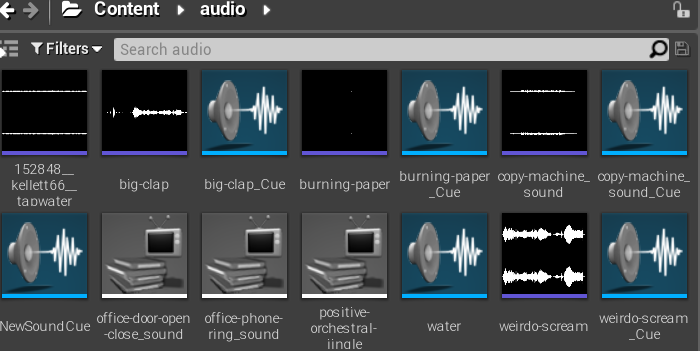
## 3.3. Görsel Efekt



Tuzaklar içerisinde unreal oyun motoru içinde bulunan görsel efektler kullanılmıştır. Bunlar tetikleyici olay gerçekleşip tuzak aktif hale geldiğinde oyun içerisinde görülmektedir. Görsel efekler kullanılırken arzu edilen etkiyi görmek için ayarlardan faydalanmak gerekmektedir. Boyutları, etkinlik süresi gibi özellikler konumlandığı yer ve gerçekleştiği zaman düşünülerek gerçekçi bir etki yaratması için dikkatlice ayarlanmalıdır.



## 3.4. Ses



Sesler için öncelikle bir arşiv oluşturuldu. Oyun içerisinde hangi noktalarda kullanılmasının uygun olacağına karar verildikten sonra arşivlenen ses dosyaları seçilerek uygun uzunluklara getirildi. Bunlar yine tetikleyici kutularla ilişkilendirilerek blueprint kodları içerisine yerleştirildi.

## 3.5. Kodlama

Unreal Oyun Motoru blueprint özelliği kullanılarak ana karakterlerin hareket alanlar, girdilere verecekleri tepkiler, oyun içerisindeki görevleri kodlandı. Tuzakların çalışma prensipleri ve karakterler ile iletişimleri kodlandı.

# 4. İhtiyaç Duyulan Araçlar

|  |  |  |  |
| --- | --- | --- | --- |
| **YAPIMCI** | **ARAÇ** | **Kullanım Alanı** | **Yapılan Çalışma** |
| Epic Games | Unreal Engine 4.21.2 | Oyun Motoru | Arkaplan aktiviteleri gerçekleştirildi. |
| Epic Games | Marketplace  Unreal Engine 4.21.2 | Grafik Dizayn ve Animasyon | 3 boyutlu model ve animasyonlar oluşturuldu. |
| Epic Games | Unreal Engine 4.21.2 | Kodlama aracı | Oyun kodlama işlemi blueprint ile gerçekleştirildi. |
| Microsoft | Microsoft Office | Döküman Hazırlanması | Fokus ve tasarım dokümanları yazıldı. |