Numele Proiectului
“Try To Survive”

Profesor Îndrumător
Ovidiu Andrei Schipor

Numele Studentului
Tudosă Iulian

Cuprins

1. Informaţii Generale

1.1 Obiectivele proiectului

Am ales să fac acest proiect, deoarece de mic copil am fost pasionat de jocuri şi mi-am dorit ca atunci când voi creşte să pot să creez jocuri. De aceea acest proiect este un joc.

Tema proiectului, lumea post-apocaliptică, o lume plină de zombii mi s-a părut o idee foarte bună ca temă de licenţă, deoarece cred că voi veni cu ceva original, cred că voi impresiona cu tema mea profesorii şi ceea ce îmi doresc foarte mult este să inspir cât mai mulţi studenţi spre această direcţie de creare de jocuri. Un domeniu fantastic care îţi ţine mintea ocupată zi de zi şi înveţi în fiecare zi lucruri pe care nu credeai că o să le înveţi făcând un joc.

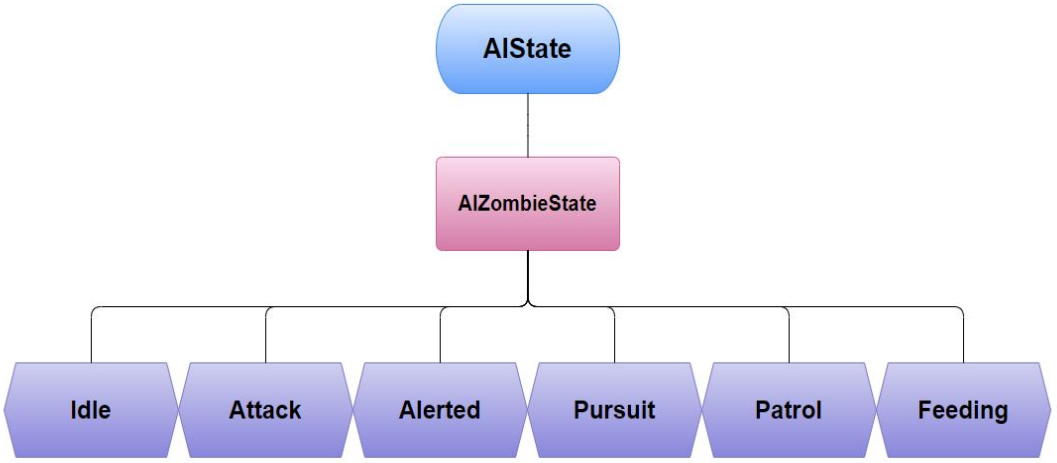
Desigur că nu este uşor să faci un joc, deoarece necesită timp foarte mult petrecut în faţa calculatorului, timp foarte mult de căutat informaţii de bună calitate, de citit cărţi de specialitate, şi partea cea mai frumoasă pentru mine de petrecut ore multe jucându-te jocuri pentru a extrage tot ce e mai bun de la un joc. Aici pot să menţionez că “GamePlay-ul” (modul de joa) este cel mai important din punctul meu de vedere, iar un utilizator trebuie să se simtă bine când se joacă un joc. Apoi intervine povestea sau misiunile pe care trebuie să le facă, apoi grafica şi aşa mai departe.

Multe variabile care sunt necesare pentru a scoate un joc de calitate şi jucabil. Este un domeniu greu dar frumos, iar eu merg pe premisa “ Cu cât este mai greu, cu atât este mai frumos” . Ştiu că mulţi dintre voi îmi daţi dreptate când zic asta, mai ales când aţi jucat un joc la un nivel foarte ridicat şi l-aţi terminat. Sentimentul pe care vi-l oferă este de nedescris, iar eu vă las să trăiţi acest sentiment pe propria piele şi jocul pe care l-am făcut poate fi un prim start.

1.2 Arhitectura proiectului

Arhitectura proiectului este foarte amplă, foarte bogată din punct de vedere a claselor folosite, a liniilor de cod scrise, a mecanicii din spate şi nu în ultimul rând al algoritmilor şi conexiunile lor pe final.

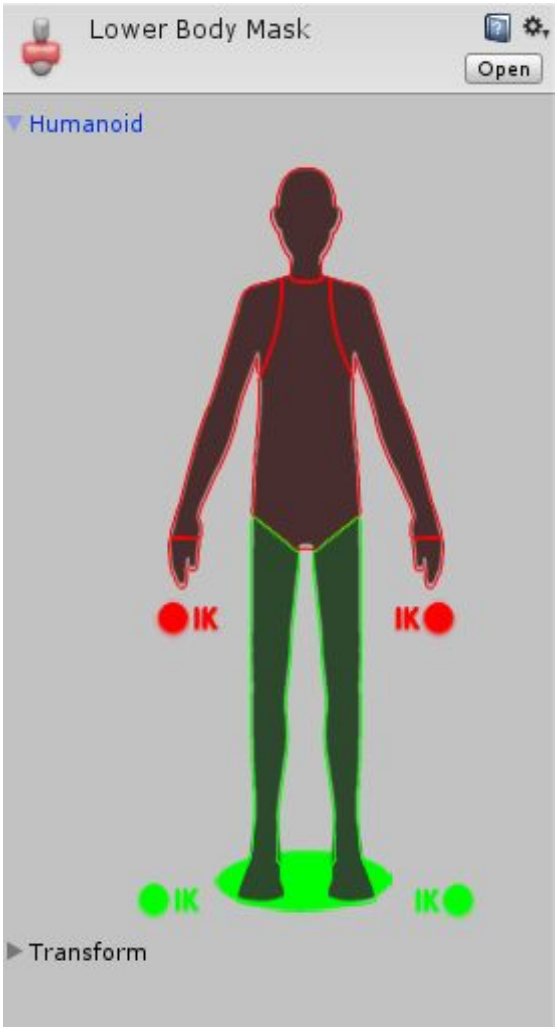
Pentru început vă prezint stările pe care trebuie să le respecte inteligenţa artificială , în special , stările zombiilor, deoarece ei sunt singurii inamici ai jucătorului.

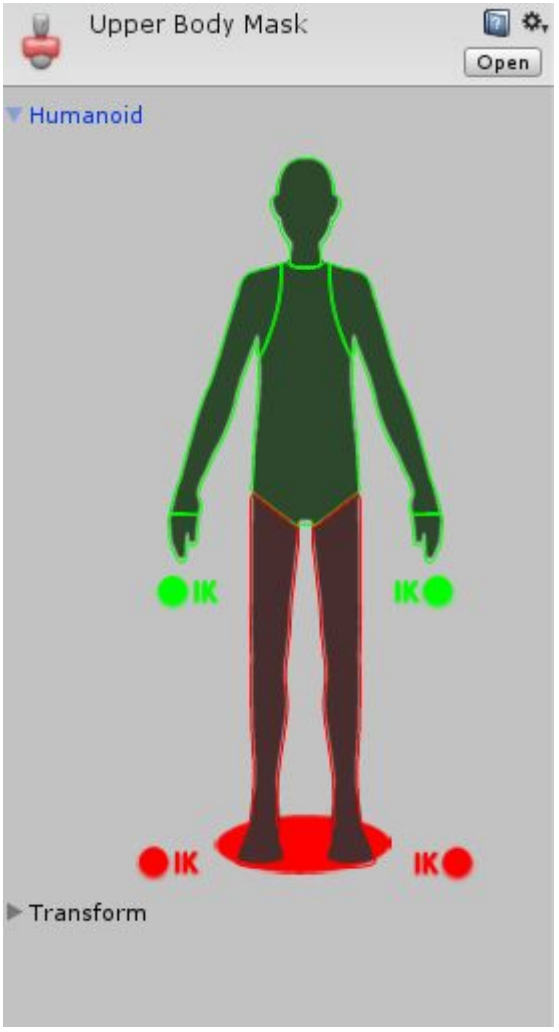


Imaginea 1.0

În imaginea 1.0 puteţi observa cum starea zombilor moşteneşte starea de inteligenta artificiala, iar cele 6 stări principale şi esenţiale implementate în acest proiect moştenesc la rândul lor stările zombilor alături de stările inteligenţei artificiale.

O observaţie foarte importantă; aş vrea să precizez, că un zombi nu va avea niciodată 2 stări deodată, adică nu poate patrula şi să se hrănească în acelaşi timp, excepţie face doar partea de atac unde cu ajutorul lui Unity 3D am reuşit să despart corpul în 2 şi să las zombiul să se mişte, şi în acelaşi timp să atace. Imaginea 2.0 şi imaginea 2.1 vă arăta cum este împărţit corpul.



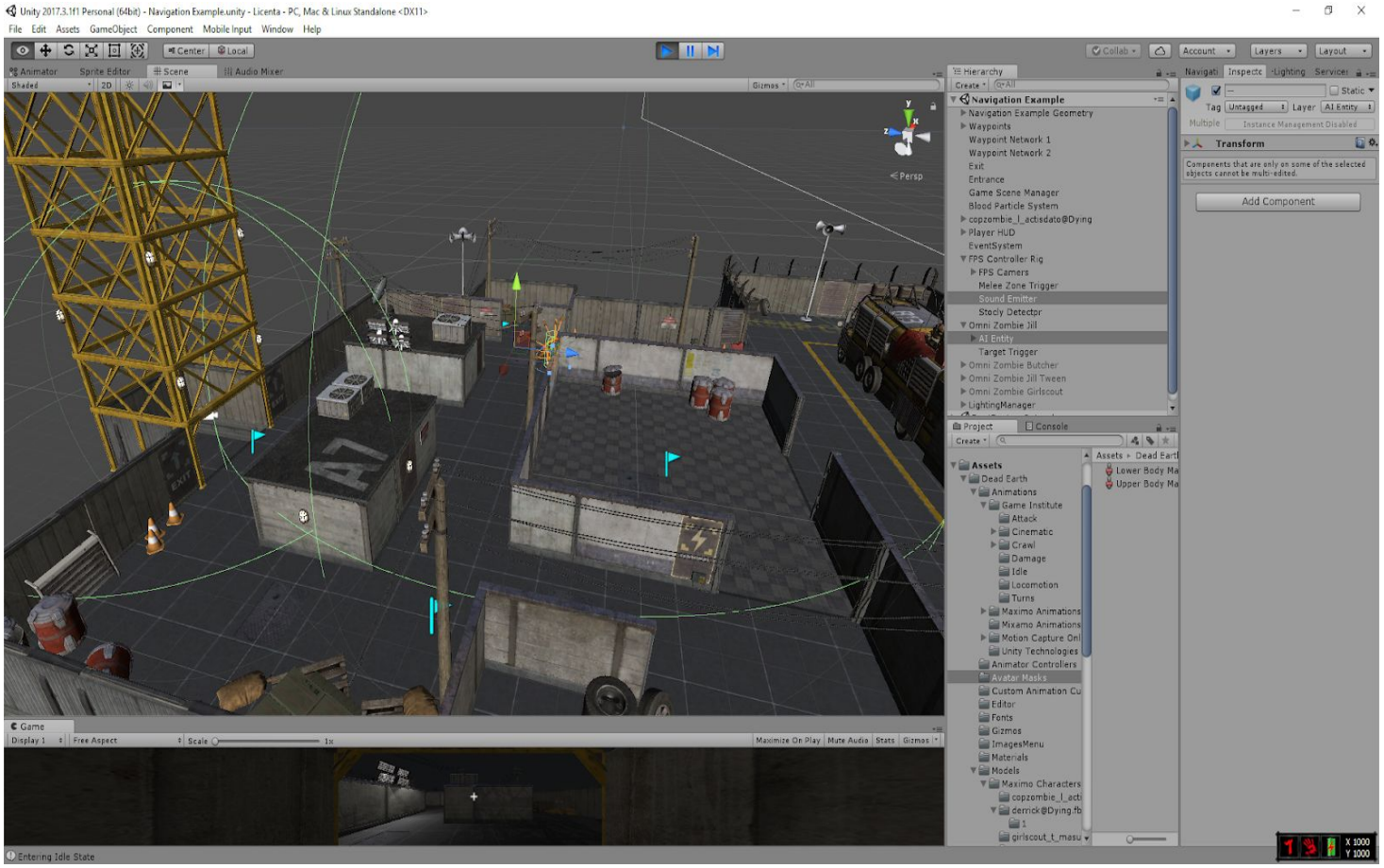


Imaginea 2.0 Imaginea 2.1

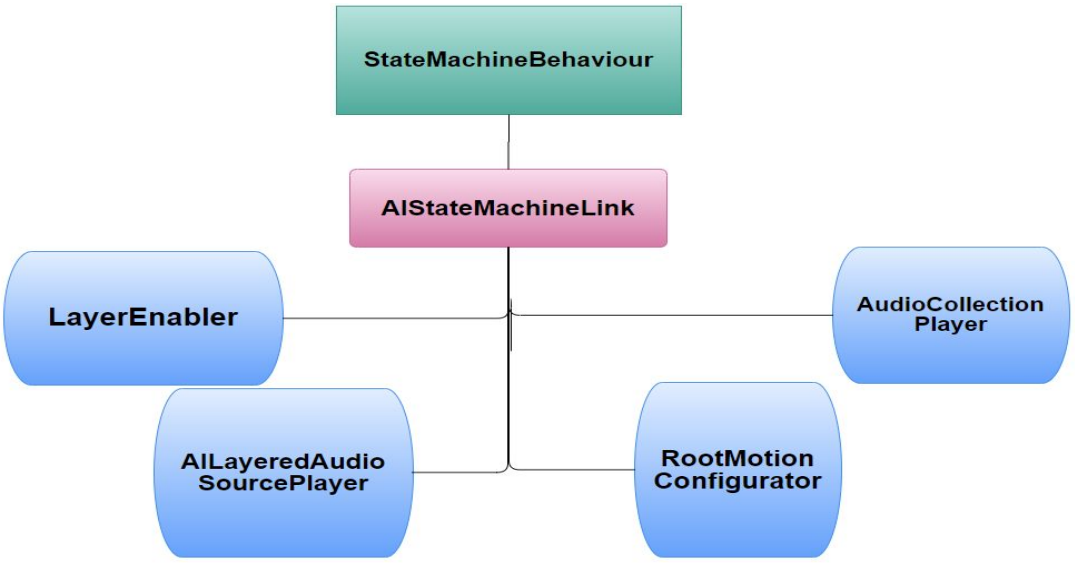
Aşadar imaginea 2.0 reprezintă partea corpului care se mai poate mişca (verde), ea fiind rezervată pentru mişcare când zombiul încă nu este rănit suficient ca să fie la pământ, iar imaginea 2.1 ca şi cea precedentă , ea fiind rezervată pentru mişcări are braţelor şi a corpului în special când atacă. Exact ca un om normal care poate merge şi atacă în acelaşi timp.

Zombii înafară de acele 6 stări, continuă cu o serie de bonusuri. Aceste bonusuri constau în, detectarea inamicului prin sunete, adică dacă utilizatorul face zgomot iar un zombi este în acea rază cum se poate observa în imaginea 3.0, aceştia detectează sunet şi se duc să investigheze (intrând în modul alert şi apoi în modul patrulare mergând spre acea zonă). Un alt bonus este sistemul audio al zombiului care se activează, atunci când este rănit, când merge sau când atacă.

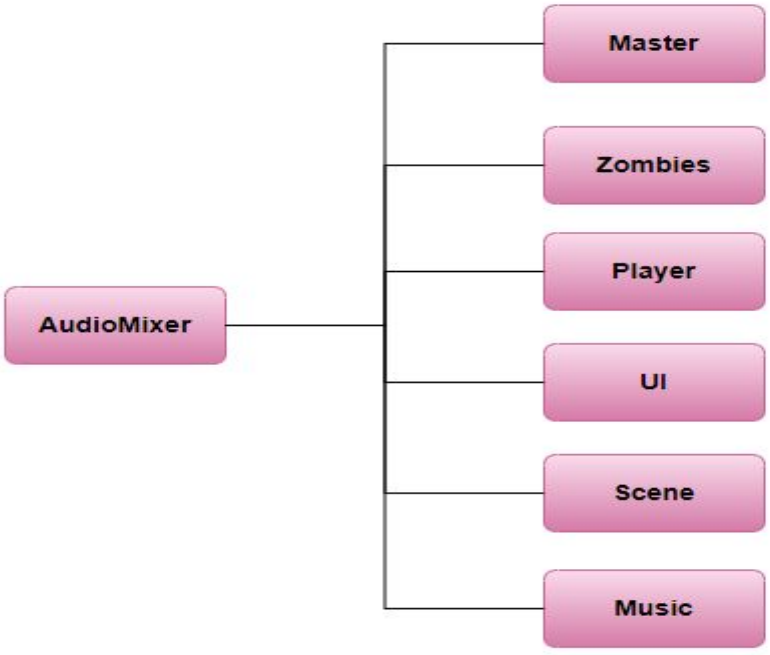
Ca toate acestea să meargă independent şi sa fie cât mai modular am fost nevoit să creez o clasă numită LayerEnabler care permite să acceseze comportamentul zombiului. În imaginea 3.1 puteţi observa cum se leagă toate aceste bonusuri una de alta.



Imaginea 3.0



Imaginea 3.1

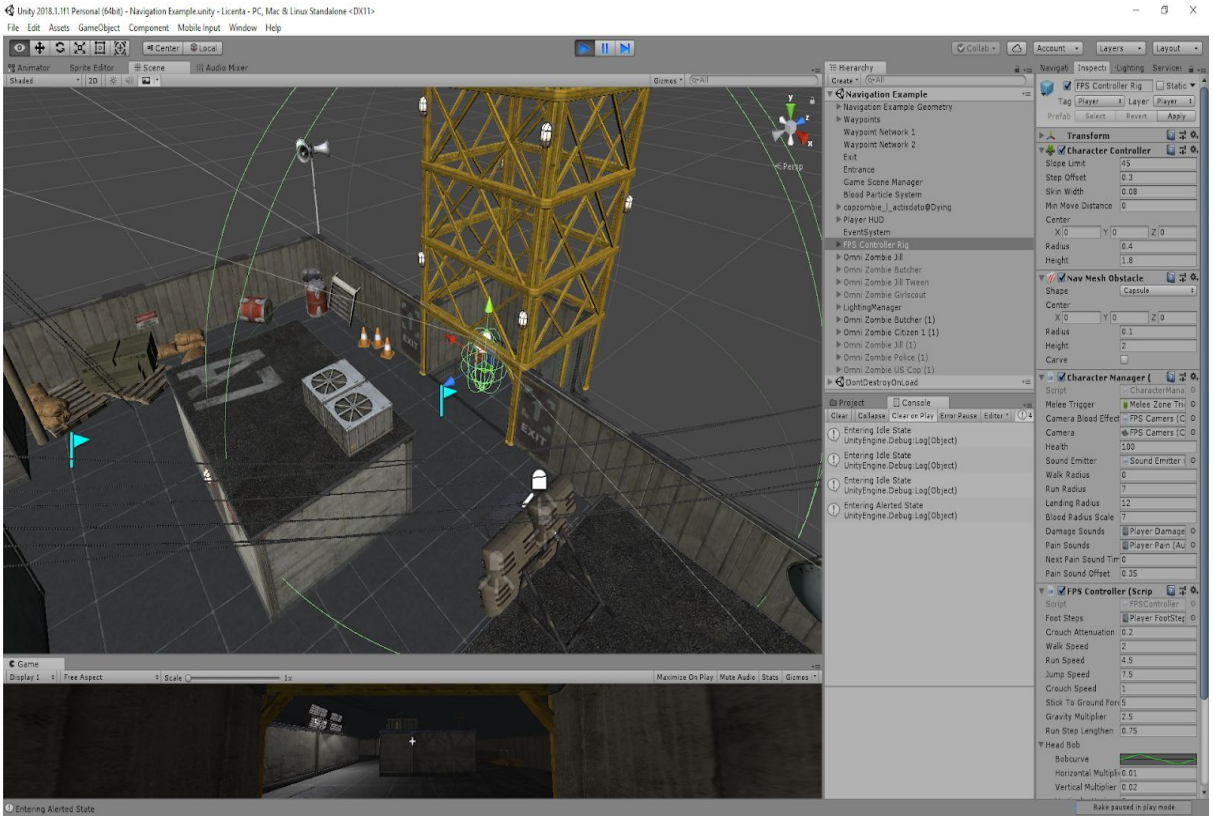


Imaginea 3.2

În imaginea 3.2 putem vedea o arhitectură a sistemului de sonorizare a jocului.

Acesta este împărţit în 6 mari categorii. **Master** este volumul principal al jocului. El poate modifica sunetul tuturor celorlalte categorii simultan. Apoi avem categoria **Zombies**, unde se vor auzi doar sunete ale zombilor, ca de exemplu: atunci când este rănit sau când se mişcă. Desigur nu ne oprim doar la aceste două, dar datorită faptului că sunt destul de multe şi că orice sunet are legătură cu un zombie, îl regăsim aici.

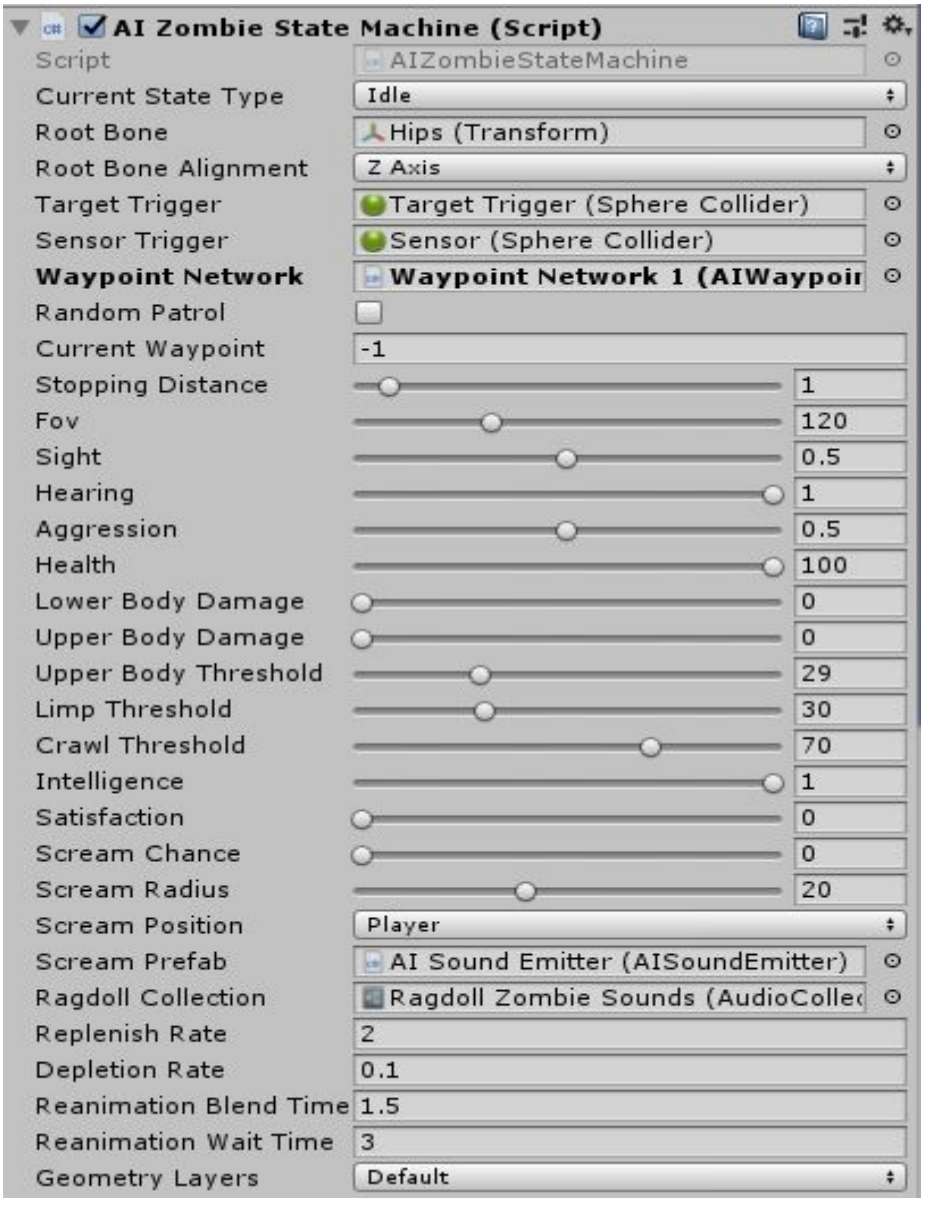
Următoarea categorie este **a jucătorului** unde aici regăsim sunete de paşi, diferite sunete în funcţie de metoda de mers. Pentru mers avem un tip de sunet, pentru alergare alt tip de sunet, pentru mers încet “mersul piticului” alt tip de sunet. Pentru că sunetul jucătorului impactează foarte mult jocul, zgomotul produs de mers atrage zombi spre jucător, iar acesta are la dispoziţie modul silenţios pentru a nu fi detectat de zombi. În următoarea imagine o să observaţi cum după o săritură a jucătorului acesta provoacă o rază de zgomot destul de mare.



Imaginea 3.3

După cum se observă din imaginea 3.3 acea mare rază verde este raza pe care o produce jucătorul. Dacă un zombi se află în acea rază, acesta intră în starea de alertă şi începe să caute jucătorul. Dacă îl vede, atunci îl atacă. De asemenea zombiul are o rază în dreptul ochilor, dacă jucătorul intră în acea rază atunci, zombiul intră imediat în starea de urmărire şi aleargă după utilizator.

Următoarea categorie este **UI** unde regăsim toate sunetele ce ţin de această categorie. Iar ultimele 2 categorii **Scene** şi **Music** conţin elemente sonore de fundal sau care rulează în spate.



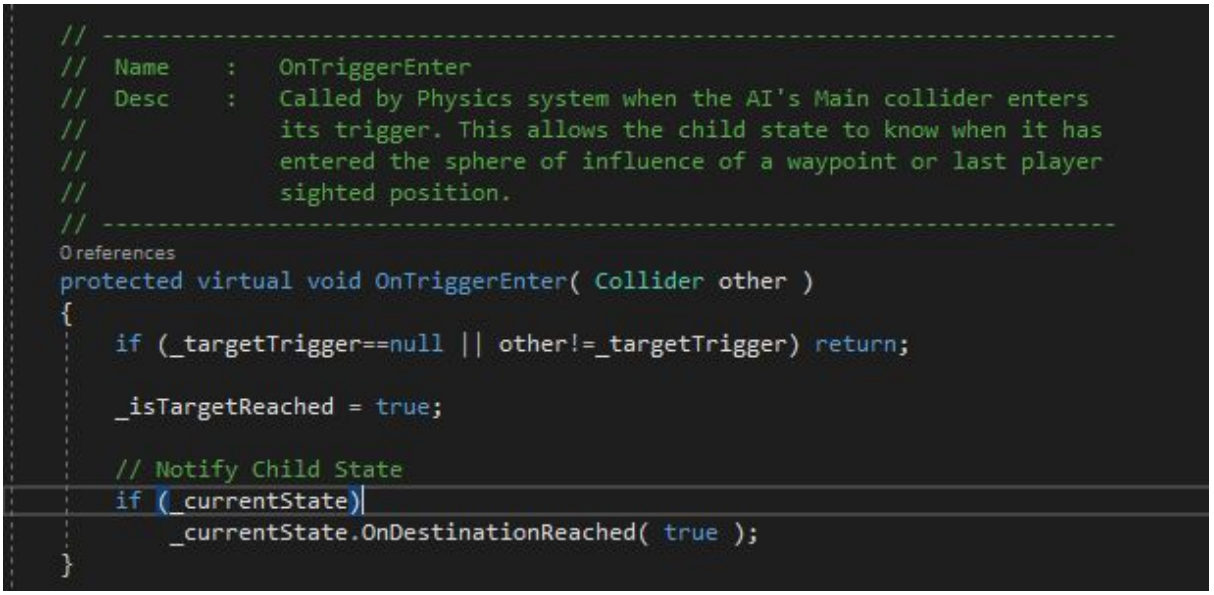
Imaginea 3.4

În imaginea 3.4 avem comportamentul unui zombi. După părerea mea cea mai importantă parte a proiectului. Spun asta deoarece conţine o multitudine de funcţii şi parametrii care dacă sunt schimbaţi, schimbă total comportamentul zombi-ului. Pentru că această secţiune a fost cea mai dificilă de realizat voi explica toţi parametrii folosiţi pentru a evidenţia cât de mult pot comportamentul gândit pentru zombi.

Primul element din şirul imens de elemente este tipul stării curente al zombi-ului. Pe scurt în ce stare se află zombiul la un moment dat. În cazul nostru este în starea de Idle (inactiv sau repaus). Mai sus puteţi regăsi toate stările zombi-ului. În timpul jocului aceste stări se vor schimba în funcţie de evenimentele pe care le întâmpină zombiul. Precizez că un zombie poate avea doar o singură stare la un moment dat.

Următoarele 2 elemente au legătură între ele, deoarece primul adică Root Bone este referinţa corpului zombi-ului, mai exact scheletul lui creat 3D. Apoi elementul următor Root Bone Alignment reprezintă axa 3D în care acesta este aliniat. În cazul nostru avem coordonatele X, Y , Z. X reprezintă ca şi în multe cazuri partea orizontală, Y reprezintă partea verticală, iar Z reprezintă partea de adâncimea sau lăţimea. Aşadar aceste două elemente memorează poziţia în spaţiu 3D a zombi-ului.

Următoarele 2 elemente sunt trăgaci traduse în română sună foarte nespecific pentru rolul pe care îl au ele, dar pentru oamenii în domeniu înţeleg termenul mult mai bine. Aşadar Target trigger (trăgaci ţintă) reprezintă în joc acea zonă reprezentativă pentru zombie atunci când jucătorul trece prin acea zonă. Dacă jucătorul trece prin acea zonă atunci activează acest trăgaci ceea ce în cod schimbă starea zombi-ului.



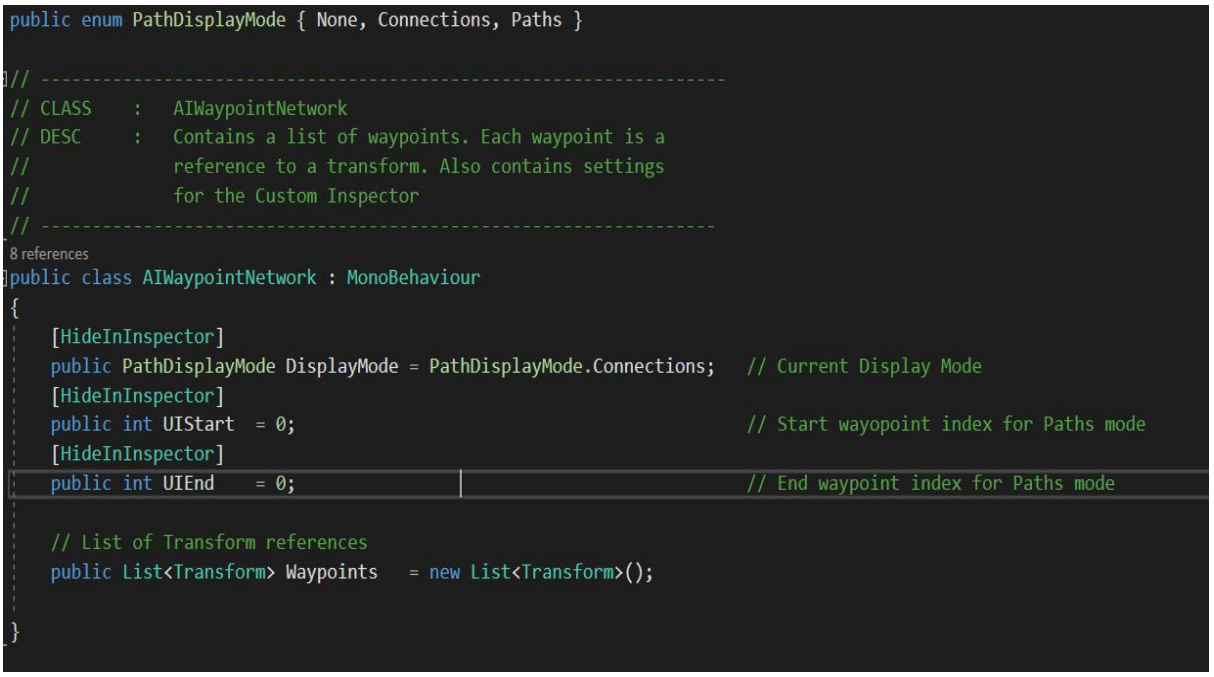
```
// ------------------------------------------------------------------------
//  Name    :   OnTriggerEnter
//  Desc    :   Called by Physics system when the AI's Main collider enters
//              its trigger. This allows the child state to know when it has
//              entered the sphere of influence of a waypoint or last player
//              sighted position.
// ------------------------------------------------------------------------
0 references
protected virtual void OnTriggerEnter( Collider other )
{
    if (_targetTrigger==null || other!=_targetTrigger) return;

    _isTargetReached = true;

    // Notify Child State
    if (_currentState)
        _currentState.OnDestinationReached( true );
}
```

Imaginea 3.5

După cum se poate observa în imaginea 3.5, Unity ne oferă o funcţie specială pentru acest tip de trăgaci, funcţia se numeşte OnTriggerEnter, iar deasupra funcţiei este explicată mai în detaliu. Apoi în funcţie se face o verificare dacă ţinta este nulă sau altă sursă care o interceptează e diferită de tipul pe care îl căutăm, iar dacă aceste condiţii sunt adevărate atunci părăsim funcţia, altfel atribuim variabilei care ţine evidenţa ţintei valoarea de adevăr. Apoi la final dacă starea curentă este o valoare de adevăr atunci, trimitem stării curente că destinaţia a ajuns.

Pe acelaşi principiu merge şi senzorul, următorul trăgaci. Acesta are un rol diferit faţă de ţintă, iar acest rol este pe partea audio. Fiecare zombi are acest tip de senzor pentru a detecta jucătorul. Jucătorul la fiecare pas, produce zgomot, iar când jucătorul aleargă acesta produce un zgomot şi mai mare. Atunci când acest trăgaci este activat atunci zombiul îşi schimbă starea curentă în starea de alertă şi începe să verifice zona pentru a detecta jucătorul.

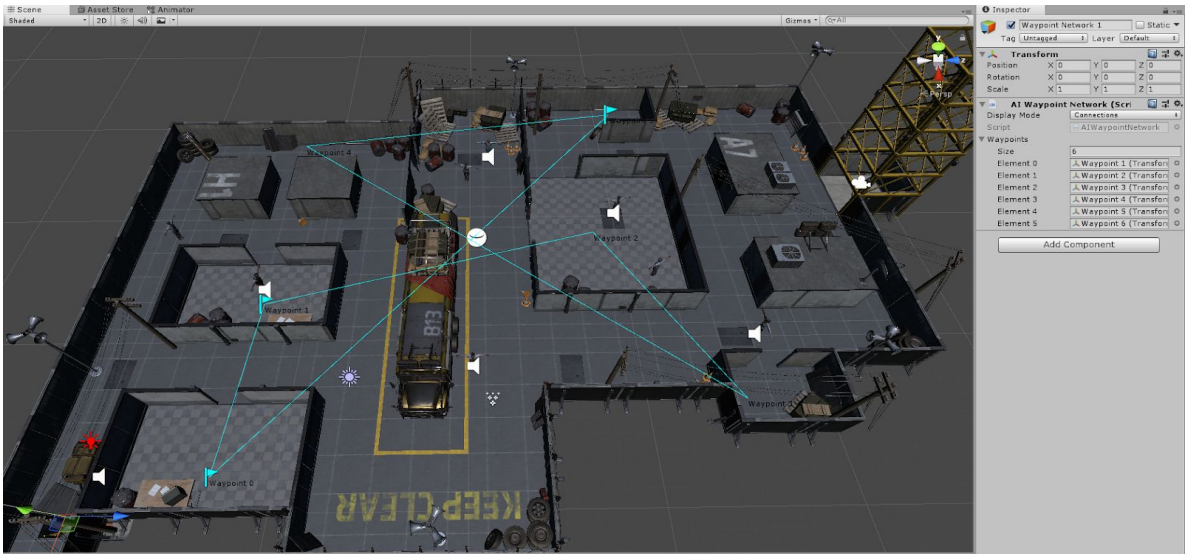
Următorul element este unul dintre cei mai importanţi. Waypoint Network sau reţeaua de puncte tradus în română, această reţea reprezintă punctele pe care un zombie trebuie să le atingă cât jocul funcţionează. Pe această reţea se bazează traseul zombilor în joc.



```
public enum PathDisplayMode { None, Connections, Paths }

// ------------------------------------------------------------------
// CLASS    :   AIWaypointNetwork
// DESC     :   Contains a list of waypoints. Each waypoint is a
//              reference to a transform. Also contains settings
//              for the Custom Inspector
// ------------------------------------------------------------------
8 references
public class AIWaypointNetwork : MonoBehaviour
{
    [HideInInspector]
    public PathDisplayMode DisplayMode = PathDisplayMode.Connections;   // Current Display Mode
    [HideInInspector]
    public int UIStart  = 0;                                            // Start wayopoint index for Paths mode
    [HideInInspector]
    public int UIEnd    = 0;                                            // End waypoint index for Paths mode

    // List of Transform references
    public List<Transform> Waypoints   = new List<Transform>();

}
```

Imaginea 3.6

În Imaginea 3.6 puteţi observa cât de simplă este clasa pentru această reţea, asta datorită motorului grafic unity care în modul editor face partea mai complicată. Lista de puncte este iniţializată în clasă şi populată cu elemente în editor. În imaginea 3.7 puteţi vedea acest lucru. Şi cum o reţea are nevoie de un punct de început şi unul de final , nici această reţea nu duce lipsă. În cazul de faţă avem 6 puncte conectate între ele. desigur că putem adăuga sau elimina din ele. Partea cea mai frumoasă este că această reţea este conectată la modul de navigare a zombi-ului. Zombiul va trebui ca să ajungă spre exemplu de la punctul 0 (0 datorită faptului că în programare se începe de la 0), la punctul 1 , acesta nu va trece prin perete, ci va ocoli şi va merge pe varianta cea mai scurtă. Algoritmul de mers al zombi-ului este prezentat mai jos. Pentru această versiune a jocului am pregătit doar 2 reţele, datorită faptului că harta jocului nu este foarte complexă şi spaţioasă. Pentru următoarea hartă voi adăuga mai multe reţele şi o varietate mai complexă pentru mişcarea zombilor.



Imaginea 3.7

Random patrol sau patrularea aleatorie , este următorul element care are legătură cu sistemul de reţele prezentat mai sus. Aceasta este doar o variabilă de tip adevărat sau fals care influenţează modul de parcurgere a reţelei. După nume este evident , dacă este activat atunci zombiul va parcurge această reţea aleatorie, iar dacă acesta este fals atunci va respecta ordinea pusă în lista vectorului ce ţine aceste valori. Un lucru foarte simplu care adaugă jocului un plus de valoare, astfel încât jucătorul nu poate memora traseul unui zombi dacă acesta va fi aleatoriu şi va trebui să fie dinamic astfel încât să finalizeze runda într-un mod cât mai corect.

Current Waypoint sau punctul curent al reţelei, reprezintă punctul curent la care trebuie să ajungă zombiul. De reţinut că dacă un zombi termină reţeaua de parcurs acesta nu se va opri şi va continua reţeaua, dacă aceasta este aleatorie atunci va merge aleatoriu din punctele oferite, iar dacă nu este aleatoriu, se va întoarce de la punctul de start şi va relua tot traseul pe care l-a parcurs.

Următorul element este Stopping Distance sau distanţa de stop pe care zombiul trebuie să o respecte faţă de jucător. Datorită faptului că jocul conţine o varietate de zombi, aceştia cu diferite forme şi înălţimi, zona de atac dintre jucător şi zombi nu va fi la fel. Aşadar am introdus această variabilă care îmi permite să setez în funcţie de zombi distanţa dintre el şi jucător pentru a oferi un aspect vizual cât mai plăcut. Dacă nu exista această valoare atunci zombiul putea vizual să apară cu jumătate de corp după cameră şi cu cealaltă jumătate în faţa camerei. Iar pentru zombi cu înălţime mai mică şi fizic mai mic, putea fi la o depărtare prea mare, iar atunci când atacau mâna lor nici măcar nu atingea jucătorul dar acesta provoca daune.

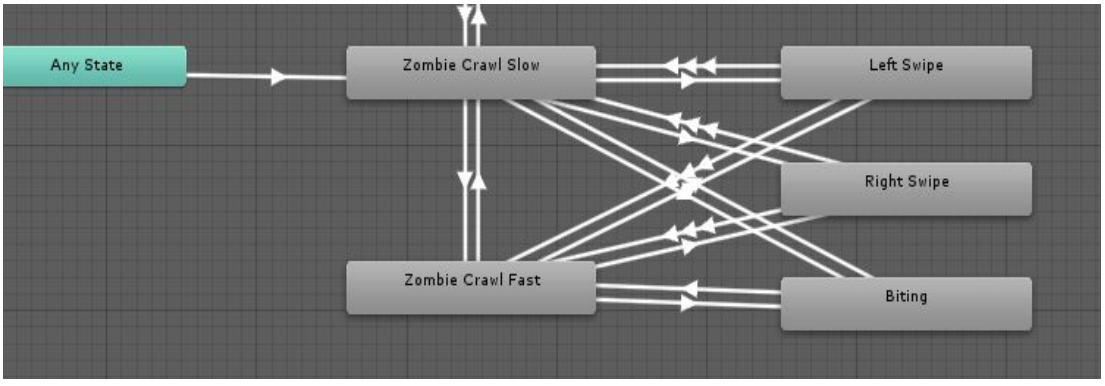
Field of view sau fov sau în română zona de vedere , reprezintă unghiul pe care îl are un zombi pentru a detecta mişcarea jucătorului. Cu cât acest unghi este mai mare cu atât zombiul poate detecta mai uşor jucătorul, în caz contrar dacă unghiul este mai mic. Această valoare oferă jocului un strat bonus de realism.

Sight sau tradus “vedere” reprezintă un multiplicator pentru distanţa pe care un zombi poate detecta jucătorul. Fiecare zombi vine prestabilit cu o distanţă , iar acest multiplicator oferă o varietate zombilor, de a nu a avea comportament identic.

Hearing şi aggression (auzire şi agresiune) sunt următoarele două elemente ce au rol de multiplicator ca şi Sight de mai sus. Hearing setează modul cât de sensibil este zombiul la sunet, cu cât acesta este mai ridicat cu atât acesta este mai sensibil şi poate identifica poziţia jucătorului mult mai rapid, iar în caz contrar îl poate detecta mult mai lent. Aggression oferă zombi-ului un surplus de agresivitate, acesta reacţionează mult mai rapid când trece în starea de urmărire sau direct în cea de atac.

Următorul element este cel mai esenţial pentru modul de joc, acesta este viaţa zombi-ului. Fără acest parametru jocul nu ar mai avea sens din modul cum l-am creat. Ca şi în oricare joc fiecare daună provocată unui zombi îi se scade din viaţa, dar nu şi de data asta. Din filme ştim că un zombi poate să moară doar dacă acesta îşi pierde capul sau este avariat creierul. Aşadar un jucător pentru a distruge un zombi trebuie să îl atace în cap.

Următoarele elemente au legătură cu sistemul de viaţă dar au o proprietate specială. Aceste 2 elemente sunt lower body damage şi upper body damage (daune zonei inferioare a corpului şi daune zonei superioare a corpului). Dacă un zombi este atacat în una din aceste 2 zone, atunci variabilele vor stoca daunele provocate. Pentru zona inferioară , dacă zona este avariată foarte grav atunci zombiul va trece într-o nouă zonă de animare. Se poate observa în imaginea 3.8 noua zonă de târâş pe care zombiul o va efectua.



Imaginea 3.8

Zona inferioară a zombi-ului se regăsesc picioarele, aşadar dacă acestea sunt distruse , zombiul se va târî pentru a ajunge la jucător.

Zona superioară a zombi-ului se regăsesc braţele şi corpul. Ca şi în cazul zonei inferioare, dacă braţele sunt distruse atunci zombiul nu îşi foloseşte braţele ci doar picioarele pentru a merge şi capul pentru a muşca. Pentru a diversifica puţin această translaţie, următorul element Upper body threshold (pragul corpului superior). Dacă această valoare este depăşită atunci se produce această translaţie.

Următoarele 2 elemente sunt tot praguri. Acestea sunt Limp threshold şi crawl threshold (pragul de neputincios şi pragul de târât). Ele au rolul ca şi celui prezentat mai sus de a stabili o limită de atins pentru a activa translaţia din cea normală în cea reprezentativă pentru pragul atins.

Un alt element foarte interesant este inteligenţa. Fiecare zombi are o anumită inteligenţă ce îi ajută în căutarea jucătorului. Spre exemplu dacă jucătorul face zgomot dar acesta face zgomot după un zid ce îl fereşte de vederea zombi-ului, atunci dacă un zombi are inteligenţă ridicată , poate să îşi dea seama unde se află jucătorul şi merge în căutarea lui. Această căutare o face pe baza poziţiei jucătorului pe hartă şi trăgaciul prezentat mai sus, acel de sunet.

Satisfacţia este un element care oferă jocului un punct de originalitate. Acesta are rolul să ofere zombi-ului modul de hrănire. Cu cât zombiul este mai satisfăcut, cu atât modul de agresivitate devine mai mic, deoarece nevoia de hrană nu mai este necesară. Desigur că atunci când un zombi se plimbă această satisfacţie scade considerabil, eu am asociat-o cu caloriile. Adică dacă zombiul se mişcă atunci acesta pierde caloriile , ceea ce îl face să îi fie foame. Atunci simţurile lui sunt crescute pentru aş-i satisface poftele. În joc sunt puşi special anumiţi zombie morţi care sunt sursă de hrană pentru zombie care sunt periculoşi pentru jucător. În imaginea următoare puteţi observa un exemplu.

Imaginea 3.9

Următoarele 4 elemente au legătură cu strigatul. Primul din cele 4 este Scream Chance (şansa de a striga). Acesta este un procentaj care se aplică atunci când un zombie intră în starea de alertă. Acest mod de strigare oferă celorlalţi zombi un semn de întrebare. Pentru cei mai inteligenţi vor veni în ajutorul zombi-ului care provoacă strigătul, iar pentru cei mai puţin inteligenţi intră în modul de alertă pentru câteva secunde şi dacă nu detectează nimic revine la modul lui normal. Al doilea element este Scream Radius (raza zgomotului). Clar că nu putem alerta toţi zombi când unu strigă aşa că această rază limitează zona şi doar acei zombi care se află în rază se vor activa. Al treilea element este Scream Position (poziţia strigătului). După cum spuneam mai sus, strigătul vine de la un zombi şi în funcţie de inteligenţă acesta merge la zombi sau direct la jucător dacă un alt zombie îl are deja în vizor. Iar această variabilă menţine poziţia ţintei. Iar ultimul element din această categorie este Scream Prefab (obiectul care menţine scriptul pentru strigăt).

Următorul element este Ragdoll Collection , o colecţie de sunete pentru zombie atunci când este lovit. Această colecţie oferă sunete aleatorii cât timp zombiul trăieşte.

Replenish rate (rata de umplere), reprezintă cât de rapid se hrăneşte zombiul. Mai bine zis cât de rapid se satisface cu hrană, iar depletion rate(rata de epuizare) cât de repede oboseşte zombiul. Aceste 2 variabile sunt explicate mai în detaliu mai sus.

Reanimation Blend Time sau timpul petrecut pentru animaţia de doborâre sau de ridicare. Pe scurt cât de rapidă să fie făcută tranziţia de la o stare de animaţie la alta.

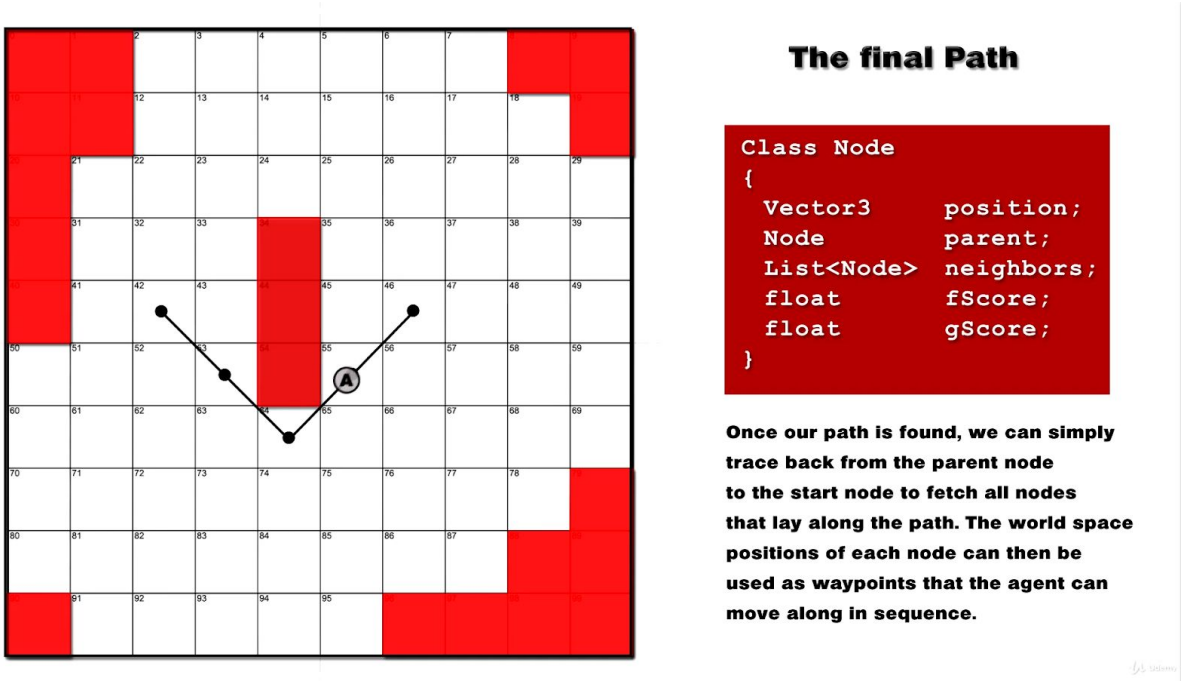
Reanimation Wait Time sau tradus timpul de aşteptare pentru reanimare reprezintă câte secunde durează până când zombiul revine în animarea normală după ce acesta a fost doborât.

Iar ultimul element din această listă este Geometry layer (stratul de geometrie). Acesta este ceva reprezentativ pentru motorul grafic Unity care să poată face legătura cu straturile reprezentative sau selectate. Aceste straturi sau layere cum li se spune în engleză au în principal aceeaşi responsabilitate cum o au şi în photoshop.

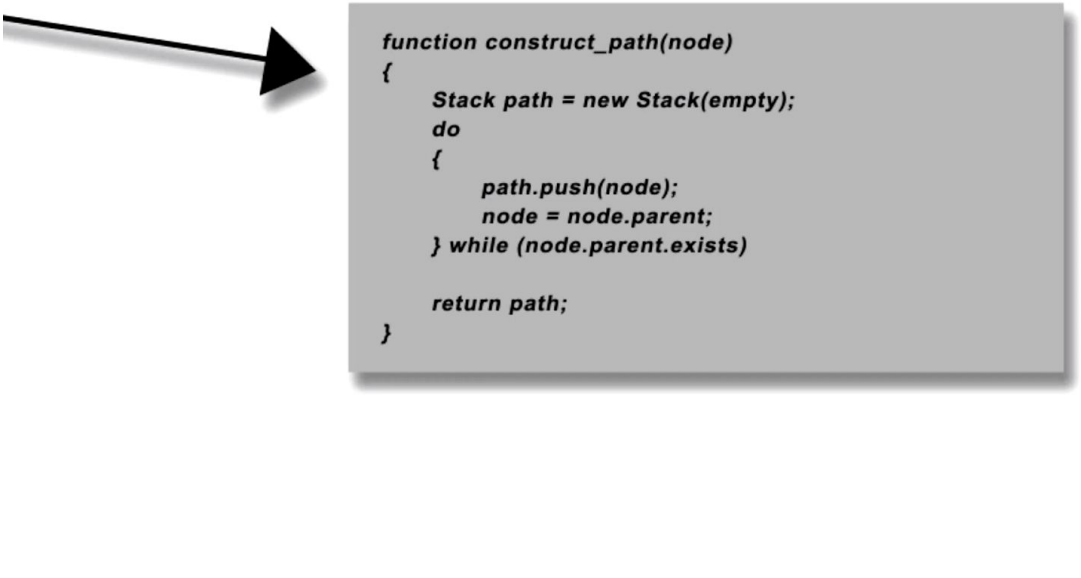
În proiect regăsim peste 30 de scripturi, peste 50 de clase, şi peste 5000 de linii de cod. Acestea combinându-se duc la următoarele produse finale: Sistem de inteligenţă artificială pentru toate categoriile de zombi, sistem de navigare pe hartă atât pentru zombi cât şi pentru utilizator, sistem de mişcare umană pentru zombi, sistem de reactivare a elementelor de mobilitate, sistem de atac atât pentru zombi cât şi pentru utilizator, sistem de energie (consum de energie cât timp se mişcă şi adăugare de energie când zombiul găseşte hrană), sistem audio de detectare a utilizatorului de către zombi.

1.3 Algoritmi

Proiectul are la bază un algoritm foarte important de mişcare pe hartă a zombilor. De acest algoritm am fost inspirat de la cursul de Proiectarea algoritmilor de către profesorul Radu Vatavu. Sursa principală al algoritmului folosit este A* Path Finding. În următoarele imagini putem urmării explicarea algoritmului.

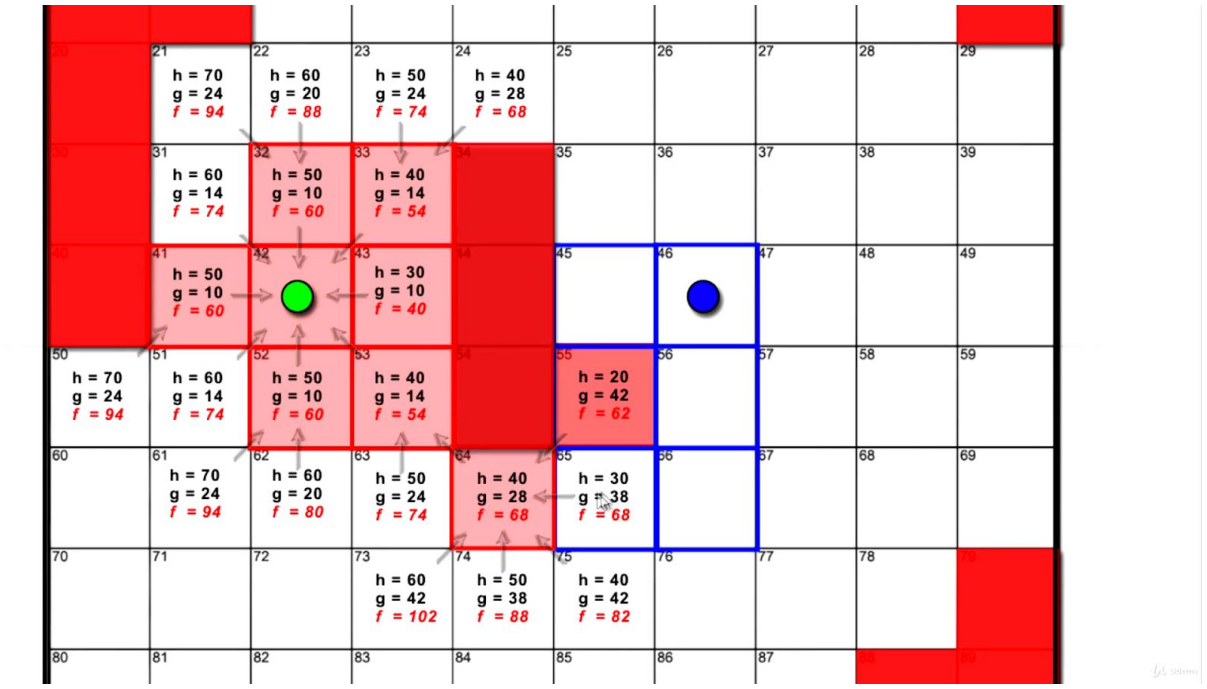


Imaginea 4.1



Imaginea 4.2

În imaginea 4.2 putem observa funcţia unde returnăm drumul pe care trebuie îl parcurgă un agent de mers.(“NavMeshAgent”).



Imaginea 4.3

În imaginea 4.3 se explică cum funcţionează algoritmul. Algoritmul se concentrează pe distanţa minimă sau distanţa cea mai scurtă dintre cele 2 puncte, prima dată se face o matrice în care se poziţionează cele 2 puncte. Apoi se calculează distanţa de la un punct la altul prin intermediul acelor noduri. Fiecărui nod îi se atribuie 10 puncte, iar cu cât este mai departe de punctul de final fiecărui nod îi creşte cu 10 puncte. În imaginea 4.3 putem observa că aceste valori sunt stocate în variabila “h”. Apoi în funcţie de direcţie se adaugă o altă valoare. Pentru nodurile care se află în nordul,sudul,estul şi vestul celui precedent acestea au valoare de 10 puncte. Pentru cele de pe diagonala precedentului nod, au 14 puncte în cazul nostru (Teorema lui Pitagora). În imagine am referinţă la această valoare prin intermediul variabilei “g” . În final toate acestea adunate pe fiecare nod primesc un punctaj , iar punctajul cu cea mai mică valoare înseamnă cel mai scurt drum. În imagine se regăseşte sub forma variabilei “f”.

1.4 Modurile de joc

Jocul este un shooter surviver, pe scurt împuşcarea tuturor zombilor pentru a supravieţuii. Pentru versiunea curentă , doar acest mod de joc este valabil, pentru următoarele versiuni am în plan să adaug şi o lista de misiuni, chestii logice pe care utilizatorul trebuie să le descifreze şi să evadeze din acel loc, evitând să omoare toţi zombi; desigur asta este opţional. Modul de joc pe care pun mare preţ şi cred că va fi cel mai jucat, ar fi un Battle Royal (pentru că tot este la modă).

Acest mod prezentat pe scurt ar fi: utilizatorul contra a 100 de zombi, muniţie limitată , zonă restrânsă, timp redus. Scopul este de a supravieţuii în faţa celor 100 de zombi.

În urma unor actualizări a jocului, am adăugat o mică misiune în joc; acela de a evada din nivelul curent. Pentru asta jucătorul trebuie să facă anumite misiuni. Pentru a afla ce misiuni are de făcut un utilizator, invit jucătorii să încerce jocul şi să ofere un feedback şi totodată să pună pauză lecturi şi să joace jocul, pentru că imaginile ce urmează sunt soluţiile misiunilor, iar acestea fiind foarte sugestibile recomand evitarea lor până la terminarea nivelului. În aceeaşi ordine de idei sunt deschis pentru orice iniţiativă de a face jocul mai distractiv.

Imaginea 5.0

După cum se observă în imagine , pentru a evada din acest loc, trebuie să folosim liftul, iar acesta ne arată ca mesaj că nu avem putere sau nu avem curent pentru a alimenta liftul. Aşadar primul lucru pe care îl vom face este să găsim un obiect ce alimentează liftul.

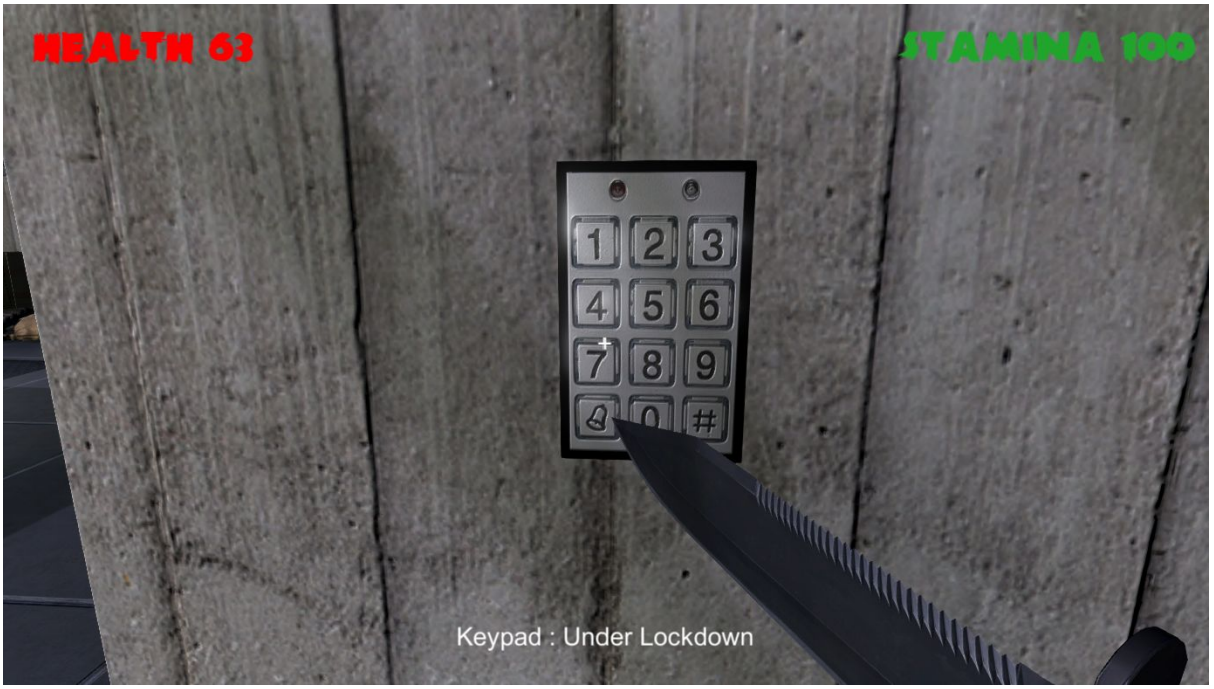
Imaginea 5.1

După lungi lupte cu zombie, ajungem cu viaţă scăzută lângă un generator de curent. Desigur că acesta nu funcţionează. Pentru a-l face să funcţioneze folosim tasta E de la tastatură să apăsăm butonul de start a generatorului. Subliniez faptul că sunt conştient că un jucător nu are de unde să ştie ce tastă trebuie să apese pentru a funcţiona generatorul şi că trebuia pus un mesaj în care să explice ce tasta trebuie apăsată, dar am considerat că jocul este destul de simplu ca nivel de complexitate, era prea banal din punctul meu de vedere să afişez şi tastele pe care jucătorul trebuie să le folosească pentru a finaliza runda. Dar totuşi am folosit pentru interactivitatea cu obiectele taste comune care se folosesc în majoritatea cazurilor, iar un jucător care s-a mai jucat până acum sunt sigur că ar nimeri ce tastă trebuie folosită. Pentru cei care se joacă prima data, le ofer puţin mister, iar modul de căutare a tastei şi punerea minţi la gândit soluţia de rezolvare, cred eu că este binevenită. Dacă voi primi feedback în legătura cu acest subiect sigur că voi schimba acest lucru, iar pentru versiunile mai avansate a jocului cu o complexitate mult mai ridicată voi face din prima acest lucru, deoarece jocul fiind complex trebuie explicat de la început noţiunile de bază, ca şi în majoritatea jocurilor de calitate de altfel.

Imaginea 5.2

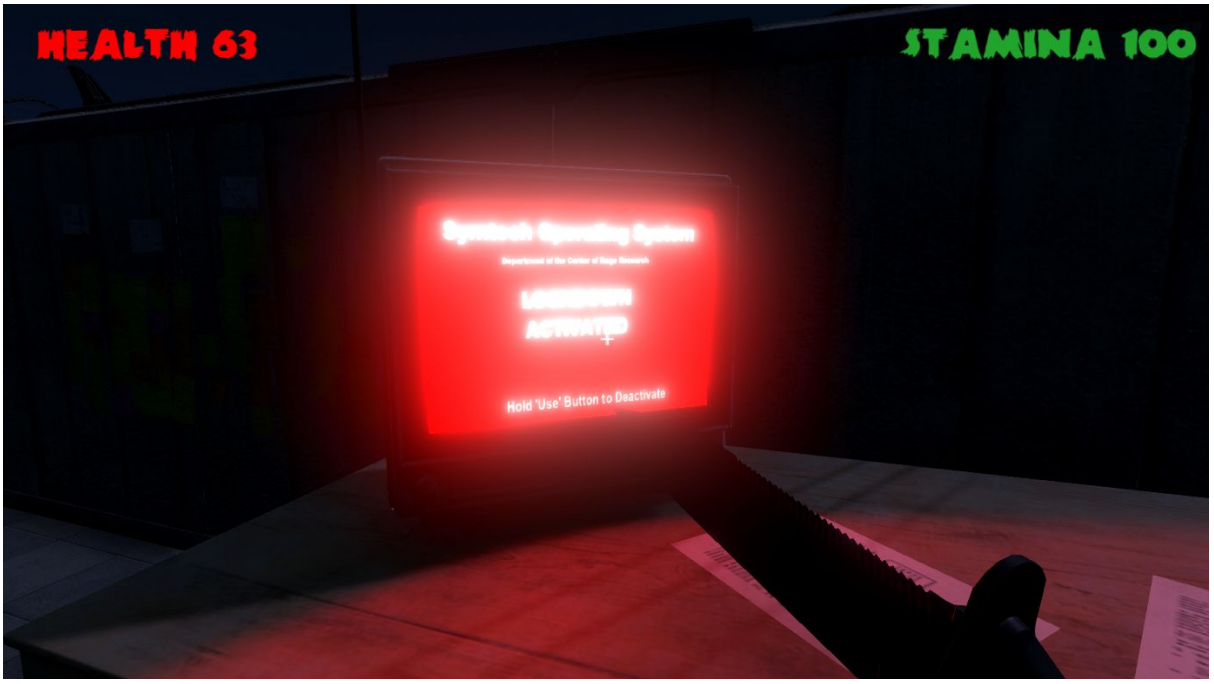
După ce am activat generatorul, jocul ne oferă vizual acest lucru prin două metode: prima prin textul afişat în josul ecranului, în care ne spune că funcţionează, iar a doua metodă prin luminile verzi alături de zgomotele generatorului.

Activarea generatorului provoacă de asemenea alarma, ceea ce alertează zombi pentru a investiga locul cât mai bine, iar pentru cei mai inteligenţi chiar pot ajunge la locul unde se află generatorul.



Imaginea 5.3

Următorul pas este de a debloca panoul liftului. Pentru această misiune trebuie să găsim un calculator. Iar după lungi căutări găsim un monitor care ne oferă o informaţie foarte sugestibilă. În imaginea 5.4 puteţi vedea mesajul:



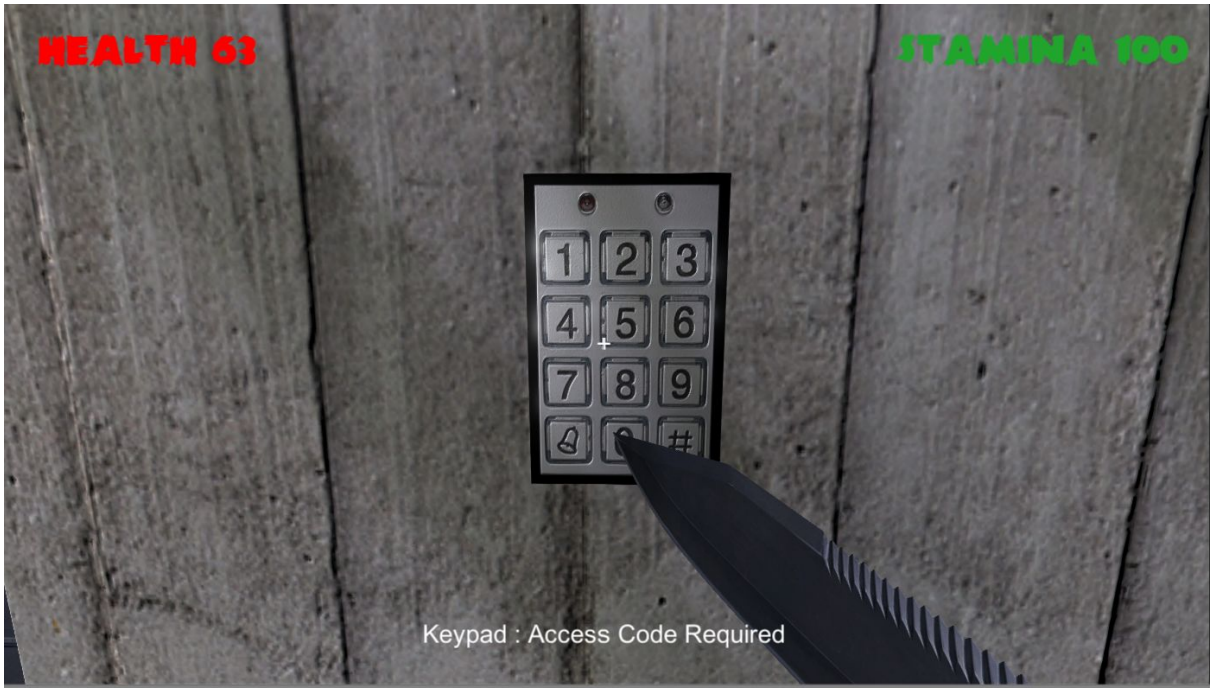
Imaginea 5.4

Mesajul monitorului este: LockDown activated, iar aici pentru a-l dezactiva trebuie să folosim tasta E din nou. Pentru a face acest lucru mai interesant am pus ca atunci când jucătorul întrerupe comanda de dezactivare, atunci tot procesul se va relua, adică jucătorul va trebui să ţină apăsat din nou de la început tasta E până bara se încarcă până la maxim, iar procesul va fi complet.

În imaginea 5.5 putem observa după ce am completat procesul de dezactivare, culoarea monitorului devine albastră , iar textul se schimbă oferindu-ne informaţia că totul a fost realizat cu succes şi că acum putem să ne întoarcem la lift.

Desigur că jocul încă nu s-a terminat, iar în imaginea 5.6 ne oferă informaţia că acest lucru este adevărat şi că misiunea de a evada din acest loc devine din ce în ce mai greu. Din imagine putem vedea că acum avem nevoie de un card de acces.

Imaginea 5.5



Imaginea 5.6

Pentru a găsi cardul de acces, trebuie să căutam acei zombi morţi de care zombi vii se hrănesc din ei. Unul din aceştia au acel card care conţine codul de acces. Aşadar trebuie să ne plimbăm pe toată harta şi să găsim acel card fără să murim.

Imaginea 5.7

După lungi căutări, găsim acest zombi mort din imaginea 5.7 care ne oferă pe ecran informaţia că el deţine cardul de acces. După ce folosim din nou tasta E, jocul oferă un sunet sugestibil când utilizatorul colectează cardul de acces.

După achiziţionarea cardului, trebuie să ne întoarcem la lift pentru a vedea dacă trebuie să mai facem ceva pentru a scăpa din acest infern.

În imaginea 5.8 apare într-un final doar numele panoului fără alte cerinţe, asta înseamnă că misiunile au fost completate cu succes şi acum se aşteaptă tasta E pentru a activa liftul.

În imaginea 5.9 vedem mesajul care ne bucură în orice joc la care am muncit şi am depus efort pentru a-l finaliza.

Imaginea 5.8

Imaginea 5.9

În următoarele versiuni jocul va avea mult mai multe moduri de joc, asta depinde foarte mult de feedback-ul persoanelor care vor juca jocul.

1.5 Meniul principal

Imaginea 6.0

Meniul principal, reprezintă punctul cel mai atractiv al aplicaţiei. După cum se observă în imaginea 5.0, meniul jocului este unul foarte simplu şi uşor de utilizat. Am aplicat User-Friendly (utilizare prietenoasă) , deoarece este foarte important ca utilizatorul să poată accesa jocul cât mai uşor, şi fiecare element grafic să fie foarte sugestibil. Exemplu concret pot să ofer textul din mijloc, adică un input text care utilizatorul trebuie să introducă parola pentru a putea juca efectiv jocul. Elementul grafic oferă o informaţie utilizatorului până ca acesta să îl acceseze, iar acea informaţie este “Enter Security Code here...” . Tradus în română ar însemna “Introduceţi codul de securitate aici...”. Acest mic detaliu de informaţie oferă aplicaţiei un plus de valoare profesională. Recomand foarte mult să se utilizeze aceste mici informaţii adiţionale pentru utilizator indiferent de domeniu în care programaţi.

Jocul întâmpină utilizatorul cu un general zombie, unul dintre zombi folosiţi în joc. Acesta are o simplă animaţie de Idle (stat pe loc). Animaţia este într-o reluare continuă cât timp utilizatorul doreşte să părăsească aplicaţia sau să intre în joc.

În spatele zombi-ului este afişat un spaţiu ce îi este aplicat o rotaţie de 360 grade, ceea ce produce un efect destul de interesant şi activ. Ceea ce am dorit să subliniez este că nu este o imagine statică cum majoritatea jocurilor o au.

Fontul ales este un font creat , apoi importat în unity şi utilizat în aproape toate elementele de afişare din joc.

1.6 Resursele folosite

Acest proiect a fost realizat cu ajutorul motorului grafic Unity 3D. Toate obiectele 3D au fost realizate în 3D studio Max şi Maya, elementele grafice de UI(interfaţa utilizator) au fost realizate în Photoshop. Elementele de sunet au fost realizate în Audacity, iar video-ul de promovare a fost realizat în Adobe Premier Pro.

Codul a fost scris în limbajul de programare c# fiind şi singurul care mai este suportat de Unity 3D de la ultima actualizare.

1.7 Noile versiuni

Pentru viitor deja lucrez la un mod nou. Acesta se numeşte Battle Royal, iar acest mod este cel mai popular mod din 2018. Acest mod reprezintă supravieţuirea dintr-un număr limitat de persoane, iar modul de supravieţuire depinde de fiecare jucător. Jocul se termină atunci când doar 1 rămâne în viaţa.

Datorită numele jocului, la o versiune superioară, aplicaţia va avea şi acest mod cât de curând posibil, dar sper că nu o să mă opresc aici şi să adaug mai mulţi zombie, nivele mai multe pentru partea de misiuni şi desigur arme noi, arme ce se pot colecta de pe jos, echipamente medicale pentru a regenera viaţa şi aşa mai departe. Să îmi las imaginaţia să îşi facă treaba şi desigur aştept şi feedback-ul jucătorilor, deoarece consider că este o opinie de luat în seamă de fiecare dată.

1.8 Blocaje

Principalele blocaje pentru a realiza acest proiect au fost în mare parte cunoştinţele pe care trebuia să le am legate de zombi. Aici mă refer la motricitatea lor. Trebuia să studiez despre cum se mişcă un zombi pentru a-l putea anima aşa cum trebuie. Sunetele pe care le scot; modul de revenire atunci când este lovit sau după ce se hrăneşte. Dar cel mai complicat lucru pe care l-am întâmpinat, a fost inteligenţa artificială a zombiilor. Peste 4 luni de lucru pentru câteva stări esenţiale a unui zombi. Printre acestea se numără , starea de inactivitate, starea de atac, starea de alertă şi asa mai departe.

Modul de abordare a proiectului a fost dificil, nefiind obişnuit cu începerea documentaţiei , zile petrecute de scris arhitectura, de legat sistemele intre ele fără să fii scris o linie de cod. Dar aceste lucruri s-au dovedit foarte folositoare pe parcursul proiectului , deoarece mi-au salvat mult timp, am ştiut întotdeauna în ce stadiu mă aflu cu proiectul şi ce trebuie să fac în continuare.

Desigur că nelipsitele “bug-uri” mi-au pus probleme, cred că cel mai lung timp de rezolvare a unui bug a fost de peste 2 săptămâni la sistemul audio a zombilor. Îmi aduc şi acum aminte eroarea sau modul care a produs acel bug. O condiţie care verifica anumite elemente audio , iar problema a fost un “1” în loc de “ i “ această mică diferenţă de font aproape insesizabilă a fost greu de depistat, iar această cauză nu verifica toate elementele muzicale şi nu îmi producea sunetul pe care îl doream. Această mică problemă m-a pus în dificultate peste 2 săptămâni şi mi-am petrecut nopţile la bug fixing (rezolvarea bug-urilor) până am dat de ea.

Să fac un mic rezumat, am fost aproape de toate tipurile de realizator de un joc. De la programator, la creator de sunete, creator de animaţii, regizor (pentru a crea video-ul), level designer până chiar şi puţină parte de grafician.

2. Bibliografie

https://docs.unity3d.com/Manual/ScriptingSection.html

https://docs.unity3d.com/Manual/Audio.html

https://docs.unity3d.com/Manual/AnimationSection.html

https://docs.unity3d.com/Manual/UISystem.html

https://docs.unity3d.com/Manual/PhysicsSection.html

https://docs.unity3d.com/Manual/Graphics.html

https://www.mixamo.com/#/