‘’Robotica’’

Game Design Document

GDD geschreven door: Sven Evertse

Februari – Mei 2018

Deltion College Zwolle

Inhoudsopgave

1. Algemene Informatie
   1. Concept
   2. Genre
   3. Doelgroep
   4. Game Flow
   5. Setting
2. Gameplay en Game Mechanics
   1. First Person Shooter
   2. Advanced Movement
   3. Vertical Gameplay
   4. Wave Based System
   5. Point system
   6. Objectives
   7. Uitdaging Factoren
3. Verhaal
   1. Algemeen verhaal
   2. Wereld en Setting
4. Level
5. Interface
   1. HUD
   2. Controls
   3. Audio
6. Artificial Intelligence
   1. Pathfinding
   2. Enemy Combat
7. Technisch Aspecten
   1. Target Software
   2. Development Hardware + Software
   3. UML
8. Game Art
   1. Art Style
   2. Assetlist
9. Plan van aanpak
10. Bronvermelding

Algemene Informatie

**1.a Concept**

Je bevind je in een sci-fi omgeving in een toren. Het doel is dat je zolang mogelijk overleeft en zoveel mogelijk vijanden vermoord waardoor. Doordat je langer overleefd en meerdere vijanden vermoord krijg je meer punten. Het doel van het spel is dan om de hoogste score te halen. Hierdoor blijft het spel herspeelbaar omdat mensen natuurlijk hun Highscore willen verbreken.

De map blijft hetzelfde en is statisch zodat de speler elke keer zijn tactieken dan kan verbeteren. Met statisch bedoel ik dat de map zelf geen random events heeft op het spawnen van random power-ups na dan. Ook word het level verticaal. Ik bedoel daarmee dat de speler voortgang kan maken naar boven om daar nog meer punten te halen. Alleen daar komen wel sterkere vijanden. Zo kan de speler beslissen om zich juist uit te dagen om hoger in het gebouw te spelen en met meer risico meer punten kan winnen of dat de speler het safe speelt en slomer punten binnen haalt. De vijanden worden ook sterker met hoeveel waves je speelt.

Het concept is heel simpel maar heel vet. De game zelf is een first person shooter met verschillende coole elementen zoals de Alternatieve Beweging, een Wave Based System en verschillende Arcade elementen zoals meerdere levens, een Highscore en random spawning power-ups.

**1.b Genre**

Het genre van het spel is een First Person Shooter met verschillende Arcade en horror elementen. First Person Shooters zijn vrij populair onder de doelgroep die ik gekozen heb. De Arcade en horror aspecten maken het spel nog iets interessanter waardoor het speel herspeelbaar blijft.

**1.c Doelgroep**

Het spel is vooral bedoeld voor mensen met de leeftijd van 18 jaar en ouder. Dat komt omdat er veel bloed en geweld in het spel voor gaat komen. Ook komt er een grimmige sfeer dat eng kan voor komen voor minderjarige spelers. Het spel is ook meer bedoeld voor de hardcore gamer. Dat komt omdat je tijd nodig hebt om dit spel te leren en je niet zo een twee drie dit spel kunt opstarten en kunt relaxen.

**1.d Game Flow**

De speler beweegt zich voor door het spel door middel van muis en toetsenbord. Het spel forceert de speler niet om in een bepaalde richting te bewegen en is vrij om te gaan waar hij of zij wilt.

Algemene Informatie

**1.e Setting**

Het spel speelt zich af in het jaar 2080 in China. Je zit in een toren met Oud Chinees interieur. Je moet vooral bij de omgeving denken aan behang met Oud Chinese kunst, Palen/standbeelden met draken etc. Ook moet je denken aan een futuristische twist zoals : High tech computers, Robot installaties etc. Ook is het de bedoeling om het spel een Grimmig tintje te geven. Daarmee bedoel ik veel bloed op de grond en een hele grimmige atmosfeer. Die grimmige atmosfeer kan ik bereiken door veel gebruik te maken van lichte dunne mist en een post processing volume. Ook kun je een grimmige sfeer creëren door enge ambient sounds te gebruiken. De grimmigheid moet niet in de weg staan tot het Arcade element om de speelbaarheid zo optimaal mogelijk te houden.



Voorbeeld voor kop van de palen of standbeeld (1) Voorbeeld van robot installaties (2)



Voorbeeld interieur van het spel (3) Voorbeeld Horror lighting (4)

Gameplay en Game Mechanics

**2.a First Person Shooter**

Een van de belangrijkste gameplay factoren is dat dit spel een First Person Shooter moet worden. Dat houd in dat de speler het spel/scenario vanuit de ogen van het speelbare karakter ziet. Daarbij moet je denken aan titels zoals Call of Duty en Battlefield. De speler ziet dus alles vanuit de Point of View (POV) van het speelbare karakter.

Omdat het een shooter is zal er natuurlijk geschoten moeten worden dus er komen ook wapens in. Er komt geen Aim Down Sight systeem in. Dat houd in dat je altijd schiet vanuit de heup vergelijkbaar met de titels van Counter Strike. In tegenstelling tot Counter Strike kun je wel rennen en hurken. De movement leg ik later in dit concept uit.

De wapens die erin zouden moeten komen zijn fictief en futuristisch zodat we in de development alle kanten op kunnen qua de design van de wapens. De wapens schieten vooral kinetische energie in plaats van kogels.



POV Call of Duty 4 Modern Warfare (5)



POV Counter Strike Global Offense (6)

Gameplay en Game Mechanics

**2.b Advanced Movement**

De movement van het spel is best interessant. Je hebt de standaard First Person Shooter movement met een paar toevoegingen. Je voert deze bewegingen uit door de W,A,S,D, Left Shift en de Left Control keyboard knoppen in te drukken. Met de standaard First Person movement bedoel ik vooral standaard bewegingen die in dit genre worden gebruikt zoals : lopen, rennen, springen en vrij rondkijken vanuit de POV van het karakter.

Een toevoeging is dat je een limiet aan Stamina (uithoudingsvermogen) hebt. Een andere toevoeging is dat je heel snel kan rond bewegen en zelfs kan vliegen als je het goed aanpakt. Door op een keyboard button te klikken activeer je die speciale actie. Als je die actie uitvoert gaat je Stamina wel sneller leeg waardoor je het slim en tactisch moet gebruiken.

Door deze speciale actie daag je spelers uit om tactisch te spelen en hun bepaald skill niveau te halen. Hierdoor blijven spelers het vaker spelen om beter te worden en verder te komen.

**2.c Vertical Gameplay**

Dit spel is vooral gericht op dat spelers tactische voordelen gebruiken om zoveel mogelijk punten te halen en zolang mogelijk proberen te overleven. Dat kun je doen met de Advanced Movement maar dat kun je ook doen om gebruik te maken van de Verticale gameplay.

Vertical gameplay houd vooral in dat de game vooral gebruikt maakt van tactische voordelen gebaseerd op hoogtes. Daarmee bedoel ik dat als je slim gebruik maakt van de verticaliteit van het level, makkelijker je vijanden kan uitschakelen en sneller punten kan verdienen. Ook kun je de Verticaliteit met de Advanced Movement combineren om je vijanden makkelijk af te schudden.

Zoals ik al eerder vermelde bevind dit spel zich in een toren waarbij je vrij bent om overal naar toe te gaan en op hogere etages meer punten kan krijgen.

**2.d Wave Based System**

Alle vijanden spawnen in zogenaamde Waves. Dat houd in dat er één groep vijand per keer spawned tot je ze allemaal uitgeschakeld hebt. Als je alle vijanden van die Wave uitgeschakeld hebt spawned er een nieuwe wave vijanden. Het aantal waves wat je gehaald hebt word bijgehouden. Hoe meer waves vijanden je verslagen hebt hoe meer vijanden er per wave komen en hoe sterker ze worden. Ook krijg je meer punten per wave.

Gameplay en Game Mechanics

**2.f Point System**

Elke keer als je een vijand uitschakelt krijg je punten. Die punten van vijanden verschillen op welke etage je zit. Als je op een hogere etage zit krijg je meer punten per keer dat je een vijand uitschakelt dan dat je op een lagere etage zou zitten. Zo kan de speler zelf bepalen of hij met weinig risico speelt en het minder moeilijk krijgt maar ook weinig punten verdient of dat de speler meer risico kiest en zo meer punten krijgt. Het aantal punten per uitschakeling vergroot per wave van vijanden. Ook krijg je als je de wave haalt punten. Dit punten blijft constant door het hele spel heen.

**2.g Objectives**

Het doel van het spel is om de hoogste score te halen. Dit doe je door zo lang mogelijk te overleven en zoveel mogelijk vijanden uit te schakelen. Je kunt gebruik maken van de hogere etages van de toren zodat je meer punten per uitschakeling krijgt en meer punten aan het eind van elke wave krijgt. Dit word een heel simpel lokaal systeem zodat je op elke computer waar het spel geïnstalleerd staat je eigen highscores kan neerzetten.

**2.h Uitdagende factoren**

De factoren die het spel uitdagend maken zijn vooral de Advanced Movement en de Vertical gameplay. Deze mechanics laten de speler zelf tactische keuzes maken. Die tactische keuzes hebben veel invloed op de moeilijkheidsgraad en je eigen scores. Spelers leren hierdoor hun eigen tactieken en gaan daarom vaker spelen om beter te worden met die tactieken.

Verhaal

**3.a Algemeen Verhaal**

Het scenario speelt zich af in het jaar 2080. Een professor heeft een Robot ontworpen om alle ziektes in de wereld uit de weg te ruimen. Hij is in China gevestigd omdat ze daar een zeldzame koper soort hebben gevonden die 100 keer meer energie kan opleveren en doorsturen dan de normale kopersoorten. Het prototype werd een groot succes waardoor deze robots massa geproduceerd werden. Tijdens de massa productie heeft een crimineel een virus in het systeem gestopt waardoor de robots de mensen zien als een ziekte. Het is nu aan de speler om deze robots zolang mogelijk op te houden zodat de professor het systeem kan uitschakelen.

**3.b Wereld en Setting**

Zie 1.e

Level

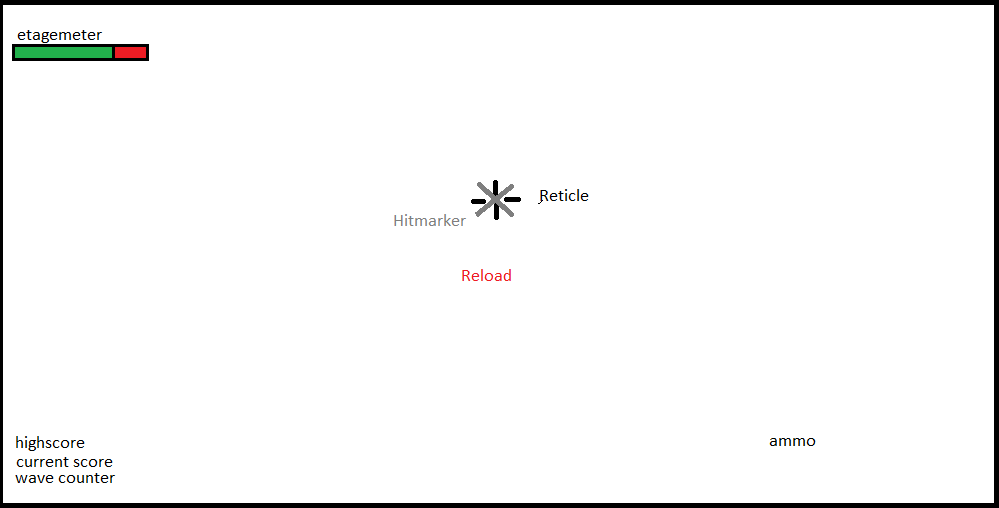
Zoals eerder vermeld bestaat het level uit meerdere etages. Iedere etage heeft zijn eigen moeilijkheidsgraad van laag naar hoog. De laagste en hoogste etage hebben een vaste ondergrond terwijl de etages er tussen meer een soort van overloop zijn waarbij je van boven naar beneden kan kijken. Om van een lagere etage naar een hogere etage te gaan kun je gebruik maken van de trappen of van je speciale ability om naar boven te boosten.

Interface

**5.a HUD**

Het design voor de in game HUD is vergelijkbaar met de HUD van Call of Duty Zombies.

* Wave Counter : Links onderin heb je een teller die telt hoeveel waves van vijanden hebt overleefd
* Current Score : Links onderin boven de Wave Counter heb je een teller die meet hoeveel punten je op dat moment behaald hebt.
* Highscore : Links onderin boven de Current Score heb je een teller die meet wat je Highscore is.
* Ammo : Rechts onderin heb je een teller die meet hoeveel ammo je op dat moment in je magazijn hebt.
* Reticle : in het midden van je scherm heb je een reticle waarmee je kunt richten.
* Etagemeter : Links bovenin heb je een element die zegt op welke etage je bent en welke moeilijkheidsgraad deze etage heeft.
* Reload : Een element die laat zijn of je wapen bijna leeg of helemaal leeg is om de speler de indruk te geven dat hij zijn wapen moet herladen. Word alleen zichtbaar als het wapen bijna leeg is.
* Hitmarker : Een element dat aantoont dat je een vijand geraakt hebt. Word alleen zichtbaar als je een vijand raakt.
* Heath : Links bovenin onder de etagemeter heb je een balkje die meet hoeveel Health je nog hebt.

****

Voorbeeld voor de layout van de UI

Interface

**5.b Controls**

De controls worden allemaal bestuurd met muis en keyboard.

* W/Pijl naar voren : Beweegt naar voren
* A/Pijl naar links : Beweegt naar links
* S/Pijl naar achteren : Beweegt naar achteren
* D/Pijl naar rechts : Beweegt naar rechts
* Left Shift : Sprint
* Left Ctrl : Boost
* Rechter muisknop : Schieten
* R : Reload
* Escape : Menu Openen
* Scrollwheel : Wisselen tussen wapens

**5.c Audio**

De audio die ik wil gebruiken zijn vooral om extra sfeer te creëren en om de speler feedback te geven op wat hij doet. Bijvoorbeeld, elke keer als je een wave van vijanden hebt uitgeschakeld wil ik het de speler duidelijk maken dat hij de wave gehaald heeft en elke keer als een nieuwe wave start wil ik dat ook duidelijk maken door middel van geluid. Andere belangrijke geluiden zijn natuurlijk van het schieten, wisselen en herladen van een wapen. Ook wil ik de robots zelf geluiden geven zodat de speler kan horen als er een robot in de buurt is.

Artificial Intelligence

**6.a Pathfinding**

Voor de Pathfinding wil ik de geïntegreerde NavMesh van Unity3D gebruiken. Ik heb hier bewust voor gekozen omdat deze NavMesh zeer betrouwbaar is en makkelijk in de omgang. Ook gebruik ik één statisch level waarmee ik makkelijk de NavMesh kan baken en dat veel performance kost. Zo kunnen er meer vijanden te gelijk in het level rondlopen waardoor er meer uitdaging komt voor de speler. Er komt wel een limiet aan vijanden dat te gelijk in een level kan spawnen. Dat limiet wil ik houden tussen de 15 en 24 vijanden. Zo blijft de gameplay soepel en word het niet te druk in de map zelf. De vijanden zijn wel bijna net zo snel als jij bent waardoor je goed gebruikt moet maken van je boost en van de hoogte verschillen.

Artificial Intelligence

**6.b Enemy Combat**

De vijanden krijgen een heel simpel maar effectief Combat Systeem. De enige manier hoe de vijanden je schade kunnen aanrichten is door je te slaan. Dat komt omdat deze robots niet ontworpen zijn om te vechten maar om mensen te helpen. Ze hebben hydraulische armen die wel hard kunnen slaan dus ze kunnen je zeker wel schade doen. Hierdoor krijg je interessante Close Quatter Combat. Daarmee bedoel ik dat zowel bijna alle combat op korte afstanden zijn.

Technische Aspecten

**7.a Target Software**

De Target Software voor dit spel is Windows. Het ontwikkelproces is een stuk sneller en efficiënter als je ontwikkelt voor Windows. Dat komt omdat je gelijk je builds op je eigen computer kan testen zonder dat je het over hoeft te transporteren naar een console.

**7.b Development Software + Hardware**

**Software**

Voor dit spel gebruiken we Unity3D als game engine. Unity3D is een gratis en betrouwbaar ontwikkelplatform waarmee we snel en efficiënt dit spel kunnen realiseren. Als programmeer omgeving gebruiken we Microsoft Visual Studio Community 2017. Visual Studio is een gratis programeer omgeving waarmee je snel en efficiënt kunt programmeren.

Voor de 3D modellen gaan we Autodesk 3DS Max gebruiken. 3DS Max is een 3D Moduleer omgeving die goed aansluit op Unity3D. Daarmee bedoel ik dat je makkelijk je 3D modellen kunt exporteren vanuit dit programma en dan makkelijk zonder problemen kunt importeren in Unity3D. Voor 2D art gaan we Adobe Photoshop gebruiken. Dat is een programma waarmee artist makkelijk 2D Sprites en art kunnen maken en ook menig ervaring mee hebben in deze sector. Voor het Texturen van de 3D models gaan we Substance Painter 2018 gebruiken. Dit is een programma waarmee je op je models kunt tekenen als het ware en makkelijk gebruik kunt maken van Physically Based Rendering.

Als repository gebruiken we Github. Met Git kun je makkelijk file sharen zonder dat je mensen in de weg kan zitten.

Om de planning en de progressie bij te houden gebruiken we Trello. Hierbij kunnen we goed controleren of mensen op schema lopen en kijken waarmee mensen op dat moment mee bezig zijn.

Technische Aspecten

**Hardware**

Een sterke processor is belangrijk voor een snel ontwikkelproces. Daarom hebben we minimaal een computer/laptop nodig met minimaal een Intel I-5 Quad Core processor. Ook hebben we een externe grafische kaart nodig om een snel ontwikkelproces mogelijk te maken. Daarom moet een computer/laptop minimaal beschikken over een Nvidia GTX 960 grafische kaart of een Nvidia GTX van een nieuwere generatie. 8GB aan werkgeheugen is ook een minimale eis waarover een computer/laptop moet beschikken.

Game Art

**8.a Art Style**

De hoofd art style van deze game is Sci-Fi. Er word veel nadruk gelegd op futurisme en robotische apparatuur. Deze art style is makkelijk te behalen als we veel gebruik gaan maken van Metallic en Emmisive maps. Ook word er een cultureel tintje in het spel gestopt. Het gebouw zelf heeft veel weg van de oud Chinese cultuur.

Ook word de art realistisch. Niet foto realistisch want dat gaat te veel afleiden van de Arcade elementen. Met realistisch bedoel ik dat de modellen en textures niet te cartoonisch worden en dat verhouden kloppen.

**8.b Asset List**

Zie bestand : KT1 Dev Deel 1 Asset List Proeve

Plan van aanpak

Om dit plan te realiseren hebben we twee artists nodig en één programmeur. De artists worden dan ook onderverdeeld in twee disciplines. Een artist die de Characters maakt en een artist die vooral de omgeving maakt en het interieur van het level.

De artist die de characters maakt gaat zich dan vooral richten op het moduleren, texturen, riggen en animeren van de characters. De artist die vooral het interieur en de omgeving maakt houd zich vooral bezig met het moduleren van die omgeving en het interieur en ook het texturen daarvan.

De programmeur houd zich vooral bezig met het programmeren van de benodigde functionaliteiten. Ook gaat hij zich bezig houden met het in elkaar zetten van het level en de builds.

Alle Assets die los gemaakt worden, worden eerst in het juiste mapje op de Git repository gepushed. Daarna worden ze geïmplementeerd in Unity door de persoon die de Asset heeft gemaakt.

Door deze workflow is het spel makkelijk haalbaar en ontstaan er weinig porblemen onderweg.

Bronvermelding

**Bronvermelding :**

1. : <https://pxhere.com/nl/photo/999360>
2. : <http://www.impressiveinteriordesign.com/the-intriguing-beauty-of-chinese-interior-design-39-pictures/>
3. : <http://media.daimler.com/marsMediaSite/en/instance/picture/Mercedes-Benz-plant-Bremen-Human-robot-collaboration.xhtml?oid=9969270>
4. : <https://www.artstation.com/artwork/nLrZX>
5. : http://www.guru3d.com/articles-pages/geforce-8800-gts-512mb-bfg-amp-pov-review,11.html
6. : <https://steamcommunity.com/sharedfiles/filedetails/?id=211661622>