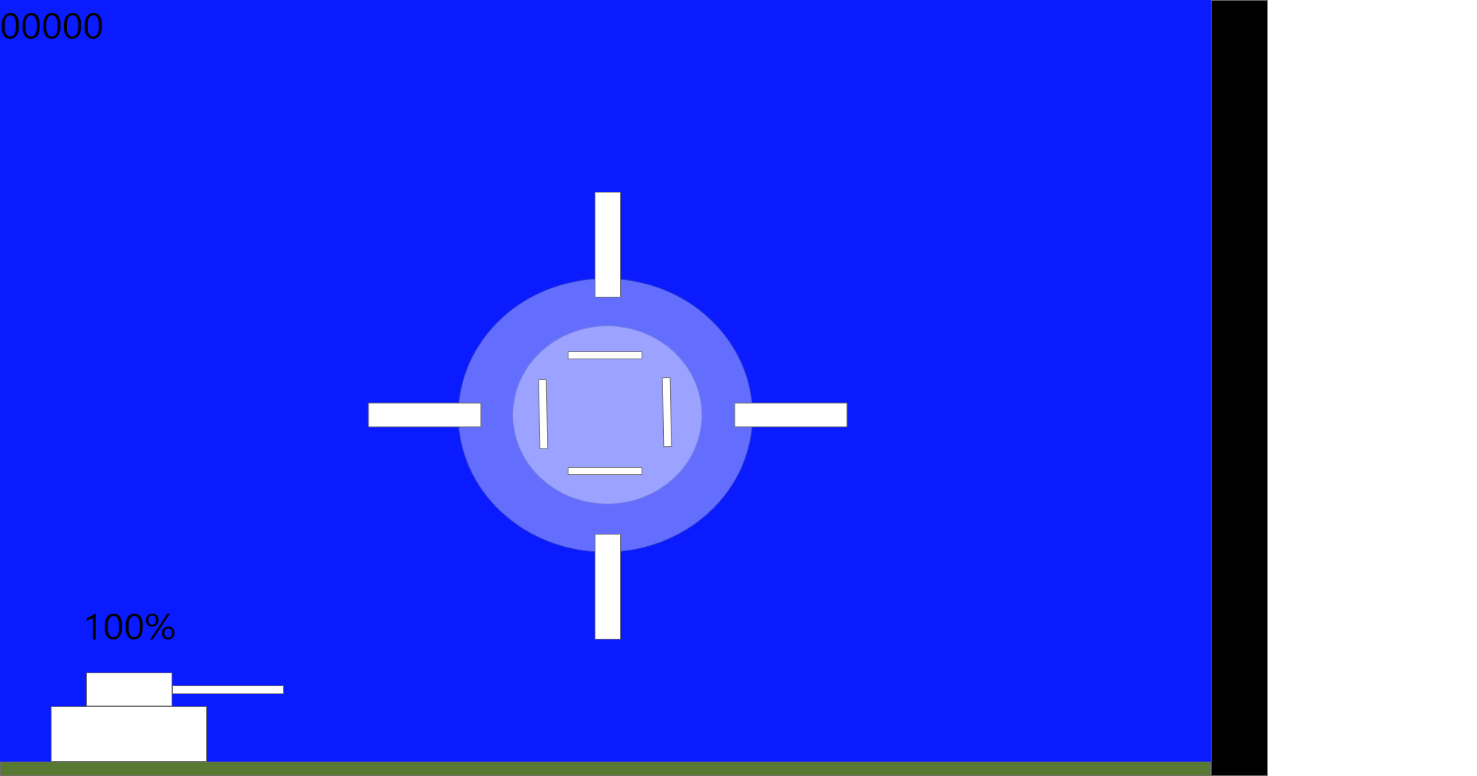
Projectplan

Defend Your Castle

Opgesteld:

23 november 2019

Versie: 1.1



Opdrachtgever:

Frans de Boer

Assistent:

Richard Kingma

Bastiaan Kramer

I4O3A

Inhoudsopgave

[Inleiding 1](#_Toc26643401)

[Aanleiding 1](#_Toc26643402)

[Doelen 2](#_Toc26643403)

[Resultaat 4](#_Toc26643404)

[Succesfactoren per onderdeel 5](#_Toc26643405)

[Afbakening 6](#_Toc26643406)

[Planning 7](#_Toc26643407)

[Risico’s 8](#_Toc26643408)

[Randvoorwaarden 8](#_Toc26643409)

[Testen 9](#_Toc26643410)

[Implementatie 9](#_Toc26643411)

# Inleiding

Dit project is opgestart voor de opdrachtgever (Frans de Boer) omdat hij graag een spin-off van de game Paratrooper wilde zien. Het eindproduct moest een Tower Defense game worden. Na aanleiding van deze vraag is door mij een vragenlijst opgesteld en beantwoord door de opdrachtgever. Deze vragen vormen later de basis voor de afbakening van het project. De opdrachtgever heeft duidelijk zijn wensen aangegeven en hierop is ingehaakt door mij (Bastiaan Kramer).

De opdrachtgever heeft mij ter referentie/verduidelijking een video laten zien van de game zelf, wat het concept was en hoe de game gespeeld werd. (Link naar de video: <https://www.youtube.com/watch?v=KXjWHYst1Nk>)

# Aanleiding

De aanleiding voor het opstarten van dit project is de proef proeve van bekwaamheid. Simpeler gezegd is het een soort voorbereiding op het eindexamen, de uitkomst van dit project bepaald of ik mijn eindexamen mag doen.

Dit was een van de elf projecten die er in totaal waren, dit project is uitgekozen omdat het toch wel interessant was om een game te maken als applicatieontwikkelaar.

# Doelen

Hier wordt beschreven wat per sprint de doelen zijn. Al deze subdoelen zijn een onderdeel voor het behalen van het hoofddoel. Doel 1 en 2 zijn documentatie, zij staan niet als sprints vermeld maar horen er wel in omdat zij de belangrijke basis voor het hele project vormen.

**Hoofddoel:**

Er is een functioneel product dat voldoet aan de eisen en wensen van de opdrachtgever en er is kwalitatief voldoende documentatie aanwezig ter ondersteuning van de opdrachtgever.

**Doel 1:**

Er wordt een kwalitatief goed projectplan opgesteld en -geleverd dat de eisen en wensen van de opdrachtgever helder en duidelijk in kaart brengt zodat er geen verwarring kan ontstaan.

**Doel 2:**

Er worden kwalitatief goede ontwerpen (Technisch & Functioneel) opgesteld en -geleverd die duidelijk de functionele en technische aspecten naar voren brengen en helder beschrijven hoe het product inelkaar zit.

**Doel 3:**

Er zullen soldaatmodellen en een tankmodel gemaakt en geanimeerd (tot op zekere hoogte) zijn. Ook is het vizier ontworpen en zal er een kasteelmodel aanwezig zijn, de hele gameomgeving is ontworpen. Ook zal er een acceptatietest gemaakt zijn die gaat bekijken of de opdrachtgever het product goedkeurt, en of het volgens de ontwikkelaar aan alle eisen voldoet.

**Doel 4:**

De code voor het project zal geschreven zijn. De code die wordt geschreven is:

* het tonen en updaten van de gezondheid van vijanden
* het tonen en updaten van de gezondheid van het kasteel
* het registreren van schade aan vijanden en het kasteel, het teruggeven van gezondheid na 10 seconden geen schade te hebben opgelopen
* het programmeren van het aftrekken van gezondheid van vijanden na een raak schot (afhankelijk van niveau, ook dit zal worden meegenomen in dit stukje)
* het opstellen van terreingrenzen zodat vijanden niet onnodig stilstaan en ze het kasteel aan alle kanten aan kunnen vallen
* het laten bewegen van het vizier (boven-onder-links-rechts)

**Doel 5:**

Er is voor gezorgd dat dode vijanden niet worden opgeruimd door het systeem, ook zal hier de logica voor het berekenen van de hoeveelheid schade per vijand worden geprogrammeerd, hier zal ook het tonen van de nieuwe gezondheid worden gemaakt (gezondheid = oude gezondheid – percentage schade).

Het scorebord zal hier ook zijn intrede maken. Er zal worden gezorgd dat de punten live optellen bij het doden van een vijand, ook zal het systeem rekening houden met hoeveel schoten elke soort vijand nodig heeft en hoeveel punten elke vijand waard is. Tevens zal hier het testplan geschreven worden ter voorbereiding op de testsprint (sprint 5 op de planning, doel 6 in dit document).

**Doel 6:**

Het testen van de componenten is voltooid aan de hand van het geschreven testplan in doel 5, de opdrachtgever speelt hier een grote rol in. De aanpassingen zijn gemaakt waar nodig. Ook zal het testrapport geschreven zijn na aanleiding van het testplan bij doel 5). Ook zal er een evalutatierapport worden geschreven aan de hand van de acceptatietest die zal worden uitgevoerd door/met de opdrachtgever. Er zal ook systeem documentatie zijn geschreven die duidelijk uitlegt hoe toekomstige programmeurs het spel zelf kunnen aanpassen naar wens.

**Doel 7:**

Er zal een kwalitatief en inhoudelijk goede presentatie gemaakt zijn die helder uitlegd hoe het product tot stand is gekomen, het verloop van het ontwikkelproces etc. Er is geen ruimte voor onduidelijkheden ontstaan en er is een duidelijk beeld van het totaalplaatje n.a.v. de presentatie, ook is er een gesprek met de opdrachtgever geweest dat positief is afgesloten.

# Resultaat

Uiteindelijk moet er dus een functioneel product komen. Maar wat houdt het product concreet in? Als het opgebroken wordt in kleinere stukken krijgen we het volgende:

* Er is een kwalitatief goed projectplan aanwezig dat duidelijk de eisen en wensen van de opdrachtgever vastlegt.
* Er zijn goede ontwerpen aanwezig (technisch & functioneel) die duidelijk uitleggen wat de technische en functionele aspecten van het product zijn en hoe deze inelkaar steken.
* Er zijn soldatenmodellen en een tankmodel gemaakt en geanimeerd (indien haalbaar, eigen initiatief) en het vizier en kasteel zullen gemaakt en gestyled zijn. Ook zal er een acceptatietest gemaakt zijn voor het testen in een latere fase.
* De code voor het spel zal grotendeels geschreven zijn, de soldaten en het kasteel hebben een gezondheidsbalk of -percentage en het vizier zal geprogrammeerd zijn (zie doel 4 onder “Doelen” voor de volledige inhoud).
* Dode vijanden zullen op het scherm blijven liggen en er is een logische berekenmethode aanwezig om de gezondheid na een treffer te berekenen per vijand. Het scorebord werkt naar behoren en toont het aantal punten en updatet deze live wanneer van toepassing.
* Het testen is afgerond en de gevonden bugs zijn gefixt. Ook zal het product een acceptatietest hebben doorlopen met de opdrachtgever. Er is duidelijke systeem documentatie aanwezig die omschrijft hoe in de toekomst het level kan worden aangepast en veranderd naar wens van de opdrachtgever.
* Er is een goede, duidelijke presentatie van het product gehouden, de twijfels die bestonden zijn hier zoveel mogelijk weggenomen en vragen werden effectief en doeltreffend beantwoord. Er heeft een eindgesprek met de opdrachtgever plaatsgevonden dat positief is afgesloten.

## Succesfactoren per onderdeel

Per onderdeel zal hier worden vastgesteld wanneer het als succesvol mag worden beschouwd. Het zijn in feite criteria die worden gesteld aan de resultaten. Wanneer is een resultaat echt behaald? Als we de resultaten stuk voor stuk bekijken dan kunnen we het volgende concluderen:

1. De documentatie (projectplan, functioneel- en technisch ontwerp, de systeemdocumentatie, testplan, testrapport etc.) is succesvol/voldoende wanneer zij onduidelijkheden zoveel mogelijk weg kunnen nemen, de onduidelijkheden die ontstaan zijn minimaal en kunnen door de ontwikkelaar opgehelderd worden.
2. De modellen van de soldaten, tanks, het kasteel en het vizier zijn succesvol wanneer zij de schermschetsen uit het technisch ontwerp zoveel mogelijk volgen, het creatieve deel wordt bij de ontwikkelaar gelaten met een paar uitzonderingen gesteld door de opdrachtgever (zie kopje “Must haves” onder Afbakening voor meer informatie).
3. De acceptatietest is succesvol wanneer duidelijk naar voren komt welke criteria er worden gesteld, wanneer het als voldoende mag worden beschouwd en er voldoende ruimte is voor opmerkingen. Belangrijk is ook dat de opdrachtgever bij het testen (de acceptatietests en de “normale” tests) wordt betrokken waar mogelijk.
4. De code mag worden gezien als voldoende/succesvol wanneer deze alle functies uit de must haves categorie vervult en de functies uit de should haves categorie zo veel mogelijk vervult. Het is belangrijk dat de code ook efficient en duidelijk wordt geformatteerd voor mensen die in de toekomst de game willen uitbreiden.
5. De logica achter de vijanden (gezondheid e.d.) is succesvol wanneer duidelijk kan worden ingesteld hoeveel gezondheid een bepaalde vijand heeft en dit ook duidelijk wordt verwerkt per voltreffer. Een schot van het geweer heeft per vijand een andere impact (soldaten niveaus 1 t/m 3 en tanks hebben allemaal een andere weerbaarheid). Het scorebord is succesvol wanneer het correct en georganiseerd het totale aantal punten toont en dit tijdens het spel bijwerkt.
6. Het testen is succesvol wanneer de opdrachtgever zoveel mogelijk betrokken kan worden bij het testen, de opdrachtgever ziet het nut van het testen van het product en kan aan de hand van het testen vragen stellen bij de ontwikkelaar indien gewenst. Ook is het belangrijk (persoonlijk) dat de opdrachtgever ziet dat de ontwikkelaar fouten in de code actief en effectief heeft verholpen (voor en na) zodat hij zeker kan zijn dat het product deugt. Ook is voldoende en duidelijke systeemdocumentatie zeer gewenst i.v.m uitbreiding in de toekomst, dit moet duidelijk vastgelegd worden voor toekomstige ontwikkelaars
7. De presentatie is als voldoende beoordeeld wanneer het voldoet aan alle gestelde eisen van de docent (tevens opdrachtgever) en het product een werkende demo heeft die de volledige functionaliteit van het product toont.

# Afbakening

In de afbakening worden alle eisen van de opdrachtgever op een rijtje gezet die tijdens de bevraging genoemd zijn. In dit gedeelte wordt aangegeven wat de opdrachtgever er absoluut in wil hebben (must-haves), wat er in moet komen indien er voldoende tijd en mogelijkheden voor zijn (should-haves), wat er in zou kunnen komen als er genoeg tijd en middelen (in dit geval kennis) voor is (could-haves) en wat er absoluut niet in moet/hoeft (won’t haves).

**Must haves:**

* Soldaten en tanks spawnen tegelijkertijd op het scherm
* Soldaten hebben 3 verschillende niveaus van sterkte
* Soldaten worden per level duidelijk onderscheiden (door middel van kleurverschillen)
* De speler is de enige die vijanden doodt, er komen geen “hulpmiddelen” aan te pas
* Het design is in overeenkomst met de opdrachtgever tot stand gekomen
* Soldaten en tanks schieten **niet** op het kasteel van de speler
* Punten worden berekend op basis van sterkte: het level van de soldaat staat gelijk aan het aantal punten (1 punt voor level 1, 2 voor level 2 etc.)
* De bewegingssnelheid van de vijanden wordt in overleg met de opdrachtgever bepaald
* Dode vijanden worden niet verwijderd van het scherm
* Het spel detecteert wanneer er 10 seconden geen vijand tegen de muur staat en geeft de muur wat gezondheid terug. (1% per 10 seconden, gezondheid mag niet over de 100% heen gaan)
* De game begint met level 1 soldaten en spawnt na verloop van tijd ook level 2 en 3 soldaten in.
* De criteria voor het voltooien van een level is gebaseerd op een puntenaantal.
* De muis dient als vizier en gaat op en neer op basis van ademhaling

**Should haves**

* Er is een structuur voor het voltooien van een level (bijv. aantal punten)
* Er is een highscore lijst aanwezig (in principe is dit een must, mochten er redenen zijn dat dit niet lukt dan wordt dit overlegd)
* De beweging van de muis is afhankelijk van de gezondheid van het kasteel

**Could haves**

* Er zijn 3 kleuren/niveaus aan tanks

**Won’t haves**

Niet aanwezig in dit project.

# Planning

De planning is een belangrijk onderdeel van het project. Omdat de uiteindelijke planning te groot is om hier in te plakken deel ik het op in sprints. Sprints zijn delen van het project waarin bepaalde onderdelen worden opgeleverd.

Vanwege de grootte van de planning (en de eis van de opdrachtgever) zal de planning als bijlage meegestuurd worden met het projectplan.

# Risico’s

Projectrisico’s staan het succesvol afronden van het project in de weg. In dit stuk wordt beschreven wat een reden kan zijn dat de opleverdatum van het project in gevaar komt

* Geen concrete kennis als gamedeveloper

Omdat mijn vakgebied ligt bij het ontwerpen van applicaties en websites is dit nogal een omscholing als het gaat om vakgebieden. Er kunnen momenten zijn dat ik niet weet wat ik precies moet doen.

* Tijd te kort

Tien weken is krap om alles te kunnen realiseren wat de opdrachtgever wenst, indien nodig moeten knopen worden doorgehakt en duidelijke afspraken worden gemaakt wanneer iets niet gehaald kan/dreigt te worden.

|  |  |  |
| --- | --- | --- |
| **Risico** | **Impact** | **Hoe te voorkomen** |
| Geen concrete kennis als gamedeveloper | Ik weet niet op alle momenten wat ik moet/kan doen en loop vast | Vragen stellen aan opdrachtgever/vriend van mij over bepaalde dingen waar ik niet uitkom. |
| Tijd te kort | Niet alles dat gewenst is kan mogelijk geïmplementeerd worden | Tijdig en duidelijk overleggen met de opdrachtgever wanneer sommige aspecten in gevaar dreigen te komen zodat er indien nodig keuzes gemaakt kunnen worden |

# Randvoorwaarden

De randvoorwaarden zijn voorwaarden opgesteld door de ontwikkelaar zodat het project tot een goed einde gebracht kan worden. Het zijn eigenlijk de oplossingen voor eventuele risico’s, aangevuld met andere eisen die nodig zijn om het product goed te laten functioneren

Allereerst is er een concrete kennisbank nodig, in de vorm van docenten die kennis hebben van de dingen waar ik tegenaan kan lopen (zoals bijvoorbeeld het probleem van de cursor)

Ten tweede heb ik een backup-plan (of liever gezegd een uitwijkmogelijkheid) nodig voor bepaalde zaken, stel dat de muis echt niet uit zichzelf te bewegen is: dan moet er een oplossing komen die beide partijen tevreden stelt.

Ten derde is een live-omgeving een vereiste. Omdat het doel van het spel in eerste instantie is dat het speelbaar is en voldoet aan de gestelde eisen en wensen is live testen een belangrijk onderdeel zodat met zekerheid kan worden gezegd dat alles ook op andere omgevingen werkt.

# Testen

Het testen van het product is een belangrijk aspect. Het zorgt dat de gebruiker weet *hoe* het product werkt, en de programmeur weet *of* het werkt. Het product in zijn geheel zal in onderdelen getest worden. De vereisten zijn door de opdrachtgever vastgelegd (terug te vinden onder het hoofdstuk “Afbakening”). Omdat het project bestaat uit meerdere onderdelen zal vooral dynamisch worden getest.

De testmethode zal voornamelijk black box testen zijn. Het product (de game) moet uitvoeren wat er in de afbakening staat. De feitelijke code is enkel relevant als hiertoe wordt verzocht. Het introduceren van het eindproduct zal in een keer worden gedaan. Ook zal het product worden onderworpen aan een acceptatietest. Meer hierover in het hoofdstuk “Implementatie”.

# Implementatie

Voor het goed implementeren van de software (in dit geval de game) is een goede planning en organisatie vereist. Wat wordt verwacht van wie, wanneer wordt de persoon geacht dit te doen, en waarom (eventueel) is dit noodzakelijk? Op deze manier weet iedereen wat hij of zij te wachten staat. Ook moet het duidelijk worden wie er extra instructies nodig hebben (denk in dit geval aan een korte optionele tutorial waarin de controls worden uitgelegd aan de speler). Ook zal er een document (of meerdere documenten) in de loop van tijd opgesteld worden zodat de opdrachtgever zelf later naar wens levels kan creeëren.

Na aanleiding van de (constructieve) kritiek van de tester(s), zullen aanpassingen worden gedaan zodat het spel fijner speelt. Een spel wat simpel maar uitdagend is, is een zekere manier om spelers tevreden te stellen. Na het afronden en opleveren van het product zal het in zijn volledigheid worden doorlopen, dit om te kijken of alles werkt naar behoren en of er onverwachte problemen optreden.