Functioneel Ontwerp

Dungeon Crawler Game ‘Tomb of the Slime King’

Lodewijk Klaassen, Jasper Kooy & Patrick Ubbink

Studentnummers: 676682, 674152 & 669278  
Mitch Holewijn  
7 april 2022

Inhoud

[1. Inleiding 2](#_Toc100216594)

[2. Het Spel 2](#_Toc100216595)

[2.1. Het Verhaal 2](#_Toc100216596)

[2.2. Technische Aspecten 2](#_Toc100216597)

[2.3. Schetsen 3](#_Toc100216598)

[2.4. MoSCoW prioritering 4](#_Toc100216599)

[3. Conclusie 6](#_Toc100216600)

[4. Persoonlijke conclusie 6](#_Toc100216601)

[5. Bronvermelding 7](#_Toc100216602)

# Inleiding

In dit document gaan we het hebben over de fucntionele aspecten van ons spel ‘Tomb of the Slime King’. Eerst hebben we een kort overzicht van het spel, waarna we ingaan op de wat meer technische aspecten van het spel, zoals gebriukers-invoer etc. Ook hebben we een aantal schetsen van het spel en een MoSCoW-lijst met dingen die wij in het spel willen implementeren.

# Het Spel

Ons spel gaat een soort ‘Dungeon Crawler’ worden. Geinspireerd op spellen zoals ‘The Binding of Isaac’ en ‘Vampire Survivors’, het doel van het spel is om zo lang mogelijk in leven te blijven terwijl steeds sterkere tegenstanders je het leven moeilijk proberen te maken en uiteindelijk de ‘Slime King’ te verslaan.

## Het Verhaal

Ons spel zal Tomb of the Slime King heten. De speler heeft controle over een poppetje genaamd Jake the Noble. Jake zit vast in een kerker en is op zoek naar een uitweg. Wat Jake echter niet weet is dat hij in een oneindig grote kerker zit, waar je alleen uit kunt komen door de Slime King te verslaan. Elke keer dat Jake een deur opent naar een nieuwe kamer komen groepen tegenstanders die hem niets liever dan dood zien tegemoet. Jake moet proberen zich door deze tegenstanders heen te vechten en zo sterker te worden. Zal het hem lukken de Slime King te verslaan en te ontsnappen?

## Technische Aspecten

Het spel zal van bovenaf worden gespeeld, waarbij de camera de hele kamer in beeld heeft en het poppetje in deze kamer rondwandelt.

De speler zal het poppetje kunnen besturen door middel van de WASD-toetsen. Met de linker muisknop zal hij een aanval kunnen uitvoeren en met de rechter muisknop een speciale aanval.

In de kamers bevinden zich verschillende objecten, die willekeurig worden neergezet. Dit zijn kratten, waar power-ups in kunnen zittten, tegenstanders en muren.

De power-ups die in kratten zitten kunnen de kracht van het poppetje vergroten, de bewegingssnelheid verhogen, of de tegenstanders zwakker maken.

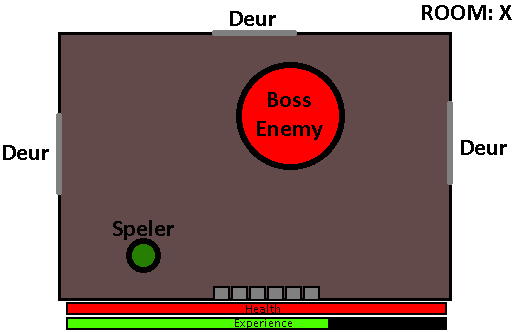
De tegenstanders zullen langzaam naar de speler toe bewegen en de speler aanvallen wanneer ze dichtbij genoeg zijn.

Het Spel zal beginnen met Jake in een lege ‘tutorial’-kamer. In deze kamer worden kort de controls uitgelegd. Als de speler door de enige deur in deze kamer gaat, komt de speler in een nieuwe kamer, waar het spel begint.

Als de speler de ‘Slime King’ kamer weet te bereiken en hier de Slime King weet te verslaan, wint de speler het spel.

De speler zal een health-bar habben die constant onderin het spel zichtbaar is. Als deze balk helemaal leeg is, is de speler vanzelfsprekend af. Eventueel kunnen we boven de health-balk een abilities balk maken, waarin verschillende abilities staan die de speler kan genruiken. Rechtsboven staat het kamernummer waar de speler zich bevindt. Ook zal er een balkje onder de health-balk zijn die de experience-punten weergeeft en daarbij de voortgang naar het volgende level.

## Schetsen



## MoSCoW prioritering

|  |  |  |
| --- | --- | --- |
| ***Eis*** | ***Omschrijving*** | ***Gelukt?/Opmerkingen*** |
| **Must Haves** | | |
| Kamers | Er zijn meerdere kamers die willekeurig geselecteerd worden. Er zijn vijf zelfgemaakte kamers die willekeurig gekozen worden als volgende kamer. | Kamers worden willekeurig gegenereerd, behalve de boss-kamer, die staat vast. |
| Speler | De speler kan een poppetje bewegen en met dit poppetje aanvallen. | Gelukt |
| Tegenstanders | Er zijn slimes die tellen als tegenstanders. Als de speler hier tegenaan loopt, krijgt de speler schade toegediend. | Gelukt, tegenstanders bewegen echter niet |
| Eindbaas | Er is een kamer met een eindbaas die sterker is dan normale tegenstanders. Als deze tegenstander verslagen is, gaat het spel door naar het eindscherm. | Gelukt |
| Eindscherm | Een eindscherm waarbij de speler opnieuw kan spelen of het spel kan verlaten. | Gelukt |
| **Should Haves** | | |
| Experience-points | Wanneer de speler een tegenstander verslaat moet de tegenstander een bepaalde hoeveelheid ‘experience’-points achterlaten. De speler kan deze oppakken en met een bepaalde hoeveelheid opgepakte punten gaat de speler level-up en wordt de speler sterker.   * De Ridder krijgt sterkere aanvallen en meer levens. * De Tovenaar krijgt toegang tot meer en sterkere spreuken. * De Boogschutter kan sneller bewegen en sneller schieten. | Niet gelukt |
| Meer diverse kamers | Er worden willekeurig kamers gemaakt via een random-generator. | Gelukt, de boss-kamer wordt niet willekeurig gemaakt. |
| Vallen | In de kamers zijn vallen die de speler schade toedienen, vastzetten of op een andere manier hinderen. | Niet gelukt |
| Power-ups | In de kamers kan de speler power-ups oppakken die de speler sterker maken voor korte tijd. | Niet gelukt, er liggen wel kratten die gesloopt kunnen worden door de kamers heen. |
| Meerdere Soorten Tegenstanders | Er zijn meerdere enemy types:   * Een schietende slime, die vanaf afstand op de speler kan schieten. * Een schuddende slime, die moeilijker te raken is door zijn hoge snelheid; * Een mitose slijm, die in twee normale slimes splitst wanneer deze gedood wordt; * Boze slime, die kleine slimes kan opeten om sterker te worden; * Vlieg slime, die over obstakels en kratten heen kan vliegen. | Niet gelukt |
| Mini-bosses | Er zijn een aantal kamers waar een mini-boss-slime is, dit is niet de eindbaas maar een significant sterkere tegenstander. | Niet gelukt |
| Loot-kamers | Er is een kans dat de speler in een loot-kamer komt, waar alleen bonussen liggen voor de speler en geen tegenstanders zijn. | Niet gelukt |
| Classes | De speler kan in de eerste ruimte een class kiezen, bijvoorbeeld Ridder, Tovenaar, Boogschutter.  Elke class heeft zijn voor- en nadelen:   * De Ridder is de basis class en heeft geen extra voor- en nadelen; * De Tovenaar kan krachtige spreuken gebruiken maar heeft minder levens;   De Boogschutter, die vanaf veraf pijlen kan schieten op tegenstanders. | Niet gelukt |
| **Could Haves** | | |
| Speciale acties | De speler kan speciale acties uitvoeren zoals een krachtigere aanval of een korte sprint.  De speciale acties verschillen per class. | Niet Gelukt |
| **Won’t Haves** | | |
| Save files | Het spel zal niet opgeslagen kunnen worden. | N.V.T. |
| Highscores | Er zal geen score toegediend worden en er zal geen highscore-lijst zijn. | N.V.T. |
| Multiplayer | Het spel zal enkel solo gespeeld kunnen worden. | N.V.T. |

# Conclusie

Bij het schrijven van de code zijn we erachter gekomen dat een aantal dingen niet zouden werken op de manier hoe wij in gedachten hadden:

* De YeagerEngine kan maar één collision tegelijk detecteren, wat erg problematisch werd in meerdere gevallen. Zo kunnen slimes onsterfelijk worden als ze tegen een muur aan staan, kunnen kraatten niet gesloopt worden als er een muur naast staat en zijn slimes die ook maar met één pixel in elkaar staan niet meer te doden.
* Het laten bewegen van de slimes zorgde ook voor problemen. Zodra we de slimes een eigen movement methode gaven, waarbij ze op eigen houtje een beetje zouden rondwandelen, gingen de frames per second van het spel direct met 20 naar beneden.
* Een ander probleem kwam naar boven met het toevoegen van Entities. De YaegerEngine kan alleen entities toevoegen in een scene, waardoor de scene een hele hoop overbodige klasses moet importeren om een entity toe te voegen.
* Het bewegen van de speler bleek af te hangen van de ververs-snelheid van het scherm van de gebruiker. Patricks scherm ververst met 144Hz, die van Lodewijk en Jasper met 60Hz. Hierdoor beweegt de speler en bewegen de enemies veel sneller bij Patrick.

# Persoonlijke conclusie

***Lodewijk*:**

Als we meer tijd hadden gehad, hadden we een veel beter spel kunnen neerzetten doordat we meer tijd zouden hebben om dingen als collision, inheritance en polymorfie uit te denken en toe te passen. Ik vond het fijn om niet de Newtonian-methodes toe te voegen voor zwaartekracht, omdat dit mij een enorme taak lijkt. Verder was het project leuk om aan te werken en heb ik veel inzichten gekregen over java-code.

***Patrick:***

Ik heb veel te lang gezeten aan het proberen op te lossen van de meerdere collisions, wat meerdere hoofdharen heeft gekost. Als we de engine beter hadden gekend en misschien wat meer tijd hadden gehad, hadden we dit kunnen oplossen en een beter product kunnen opleveren.

De communicatie in de groep ging goed en het zorgde er voor dat we allemaal met veel plezier aan het spel hebben kunnen werken.

***Jasper:***

Het was erg leuk om aan het spel te werken met z’n drieën. Wanneer ik ergens even niet uit kwam, waren er meteen twee mensen aan wie ik kon vragen of ze even zouden willen meekijken naar mijn domme foutjes.

De engine hielp hierbij niet echt, er zijn veel verwarrende termen die wat mij betreft wat beter benoemd of beter uitgelegd hadden kunnen worden. Denk aan de Collider en Collided, de timer-functie en de setMotion functie.

# Bronvermelding

*Primitive Data Types.* Oracle.com. Laatst geraadpleegd op 6 april 2022 , van <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html>

*Han-Yaeger Tutorial.* Han-yaeger.github.io. Laatst geraadpleegd op 5 april 2022, van

<https://han-yaeger.github.io/yaeger-tutorial/introduction.html>

*Han-Yager Book.* Han-yaeger.github.ino. Laatst geraadpleegd op 3 april 2022, van

<https://han-yaeger.github.io/yaeger/book/introduction.html>

*Han-Yaeger Github Repository.* Github.com. laatst geraadpleegd op 24 maart 2022, van

<https://github.com/han-yaeger/yaeger-tutorial>

*Javadocs.* www.jetbrains.com. Laatst geraadpleegd op 6 april 2022, van

<https://www.jetbrains.com/help/idea/working-with-code-documentation.html>