27 juni 2021

Thomas Luchies - 4673913

Sydney Minnaar - 4753045

Robin Goeree - 4697081

Storm Visser - 4651219

Onno Busscher - 4565533

Startdocument  
Toucan: Egg Quest 2D

# Versiebeheer

|  |  |  |  |
| --- | --- | --- | --- |
| Versie | Auteurs | Aanpassingen | Datum |
| 0.1 | Thomas | Omschrijving gemaakt | 13-05-2021 |
| 0.2 | Sydney | ERD en klassendiagram gemaakt | 14-05-2021 |
| 0.3 | Onno | Mockups gemaakt | 16-05-2021 |
| 0.4 | Robin | Planning gemaakt | 16-05-2021 |
| 0.5 | Storm | MoSCoW gemaakt | 17-05-2021 |
| 0.6 | De hele groep | MoSCoW ingevuld en laatste aanpassingen gemaakt | 18-05-2021 |
| 0.7 | De hele groep | Feedback verwerken en bronvermelding gemaakt | 26-05-2021 |
| 1.0 | De hele groep | Laatste feedback verwerken | 01-06-2021 |
| 2.0 | De hele groep | Updates aan de hand van realisatie | 27-06-2021 |

Inhoudsopgave

[Versiebeheer 1](#_Toc75726506)

[1. Functionele omschrijving 3](#_Toc75726507)

[1.1 Game mechanics 3](#_Toc75726508)

[1.2 Omgeving 3](#_Toc75726509)

[1.3 Vijanden 3](#_Toc75726510)

[1.4 Power-ups 4](#_Toc75726511)

[1.5 Levels 4](#_Toc75726512)

[2. Technische omschrijving 5](#_Toc75726513)

[3. MoSCoW prioriteiten 6](#_Toc75726514)

[3.1 Must have 6](#_Toc75726515)

[3.2 Should have 7](#_Toc75726516)

[3.3 Could have 7](#_Toc75726517)

[3.4 Wont have 8](#_Toc75726518)

[4. Mock-ups 9](#_Toc75726519)

[4.1 Het Splash-screen 9](#_Toc75726520)

[4.2 Het Hoofdmenu 10](#_Toc75726521)

[4.3 Het opties 11](#_Toc75726522)

[4.4 Het creatie scherm 12](#_Toc75726523)

[4.5 Het Speel scherm 13](#_Toc75726524)

[5. Klassendiagram 14](#_Toc75726525)

[6. Planning 15](#_Toc75726526)

[7. Verwijzingen 16](#_Toc75726527)

# 1. Functionele omschrijving

In dit hoofdstuk wordt er een omschrijving gegeven van het te ontwikkelen systeem.

## 1.1 Game mechanics

Het idee is dat er een 2d platform spel gemaakt wordt waarbij de gebruikers zelf ook hun eigen levels kunnen ontwerpen en zelf kunnen spelen. De gebruikers krijgen in het hoofdmenu twee opties. Ze kunnen een eigen level maken in de level designer of een eigen gemaakte level spelen. Als de gebruiker nog geen level heeft gemaakt zal er een standaard level worden ingeladen.

## 1.2 Omgeving

De omgeving bestaat uit meerdere objecten hieronder vallen beweegbare en niet-beweegbare. Als eerste heb je de baselevel, dit is een non beweegbare object waar de speler op kan staan en het level kan geen object hebben wat lager is als deze laag. Ook heb je een platform, dit is een beweegbaar object. Er kunnen meerdere platformen in het level geplaatst worden door de ontwerper van het level. Op deze platforms kan de speler springen. Je kan van onder op het platform springen maar niet van boven weer vallen. De wolk heeft dezelfde aspecten als een platform alleen is deze is deze hoger dan het platform.

Alle niet-beweegbare objecten in de game zijn de boom, een tak, een waterval, een vijver, een steen en een niet-beweegbare platform. Een boom bevat een tree cavity, of een boomholte waar de speler zijn of haar checkpoint kan instellen. De waterval en de vijver zijn objecten die de toekan niet kan aanraken. Als de toekan deze objecten aanraakt gaat er een leven van de speler af gaat de speler een stuk achteruit. Ook zit er een steen in het spel, dit is een object waar de speler niet doorheen kan lopen en overheen moet springen om door te kunnen.

## 1.3 Vijanden

De speler speelt als een toekan. Deze toekan moet het level doorgaan zodat hij bij zijn nest kan komen. Als hij bij zijn nest is, is het level voltooid. Onderweg naar zijn nest komt hij veel van zijn natuurlijke vijanden tegen, waaronder: een slang, een roofvogel en een jaguar. De slang en de jaguar bewegen op de horizontale as, de roofvogel beweegt op de verticale as. Als de speler een vijand aanraakt verliest hij of zij een leven.

## 1.4 Power-ups

De toeken kan verschillende power ups krijgen. De power-ups komen in de vorm van een besje. De besjes hebben verschillende kleuren en aan elke kleur zit een andere power-up verbonden. Het paarse besje is een double-jump. Met deze power-up kan de speler een extra sprong maken nadat hij er al één heeft gedaan zodat hij hogere platforms kan bereiken. De blauwe power-up geeft de speler de mogelijkheid om over water te lopen. De speler kan alleen met deze power-up over het water en verliest hier geen levens van. Met het rode besje krijgt de toekan de mogelijkheid om vijanden te verslaan met zijn snavel. De speler kan dan simpelweg tegen de vijanden aanbotsen. Dit zorgt ervoor dat de vijand doodgaat en de toekan van die vijand geen levens meer kan verliezen. De laatste power-up is de zweef power-up, deze kan verkregen worden door het witte besje op te pakken. Met de zweef power-up kan de toekan voor een tijdje zweven nadat hij gesprongen heeft zodat hij verder kan komen. Alle power-ups zijn tijd gebaseerd.

## 1.5 Levels

Zoals eerder vermeldt kan de speler zijn of haar eigen levels maken. Hierbij heeft de speler de optie om alle objecten, vijanden en objectieven in een gebied compleet vrij neerzetten. Nadat de speler een compleet level heeft gemaakt kan hij op dit level gaan spelen. Om te zorgen dat de spelers zo veel mogelijk variatie krijgen zullen er weinig restricties worden gelegd op het aanmaken van objecten. Als de speler nog geen level heeft gemaakt krijgt hij of zij de kans om een proeflevel te spelen. Dit level is voorgemaakt door ons en bevat een groot gedeelte van de verschillende objecten, maar niet alle objecten. Hierdoor heeft de speler een goede indruk van de mogelijkheden, maar kan hij of zij ook nog zelf experimenteren.

# 2. Technische omschrijving

Het spel zal worden ontwikkeld in C# .NET Core (contributors W. , 2021). Er is getwijfeld om gebruik te maken van .NET, maar in de context van de casus voor deze opdracht maakt het niet veel uit welke wordt gebruikt.

Voor het grafische gedeelte van het spel wordt er gebruik gemaakt van Universal Windows Platform (UWP) (contributors W. , Universal WIndows Platform, 2021). Alhoewel WPF een wat volwassener raamwerk is denkt de projectgroep dat UWP in de toekomst vaker zal worden gebruikt. Daarnaast vindt de projectgroep dat het basis design van UWP in Windows 10 er moderner uit ziet.

Er wordt geen gebruik gemaakt van een game-engine. Omdat UWP op zichzelf voldoende moet zijn om een 2D Platformer te ontwikkelen. Na onderzoek is er besloten om geen gebruik gemaakt van het ontwerp patroon model-view-view-model (MVVM) (contributors W. , Model view viewmodel, 2021). Alhoewel er een aantal aspecten in het spel kunnen voorkomen waarbij MVVM gebruikt kan worden, zoals een databinder die direct de juiste afstand meegeeft voor ‘set top’ of ‘set left’ (positioneren van objecten in UWP) op basis van de x en y as van een object, kan dit ook gemakkelijk worden opgelost zonder MVVM.

Als er daarnaast nog genoeg tijd over is zal er door middel van procedural generation (contributors W. , Procedural generation, 2021) sleutels worden gegenereerd die gedeeld kunnen worden met andere spelers. De level maker kan dan de sleutel delen met potentiële spelers via andere platformen, zoals sociale media.

Aangezien de projectgroep samen één applicatie ontwikkeld wordt er gebruik gemaakt van het versiebeheersysteem GitHub (GitHub, 2021). GitHub voldoet aan alle wensen voor dit project, daarnaast hebben alle groepsleden hier veel ervaring mee.

# 3. MoSCoW prioriteiten

De eisen voor dit project zijn beschreven in dit hoofdstuk. Er is gekozen om voor de MoSCoW methode (Wikipedia-bijdragers, MosCoW-methode, 2021) te kiezen om de prioriteiten in kaart te brengen, hierbij worden de verschillende taken in een aantal categorieën verdeeld.

## 3.1 Must have

Dit zijn de onderdelen van het spel die essentieel zijn voor de werking, en die dus ook de hoogste prioriteit hebben. Hieronder een lijst van de onderdelen in deze categorie;

* De basegame:
  + Bestaat uit statische objecten
  + Bestaat uit beweegbare objecten
  + Bestaat uit vijanden van de toekan
    - Wanneer de toekan in aanraking komt verliest hij een leven
  + Bestaat uit een tijd-gebaseerde gamemode (binnen een bepaalde tijd moet het level worden behaald)
  + Een speler (bestuurd de toekan)
    - Kan zowel naar links als naar rechts
    - Kan springen
    - Heeft collisie met alle objecten, behalve:
      * De waterval
      * Wanneer er wordt gesprongen onder een wolk of tak object (Dan stoot hij niet zijn hoofd maar gaat hij er als het ware doorheen)
    - Moet bepaalde doelen halen met betrekking tot de time modus
* In de level-maker:
  + Kunnen alle hierboven benoemde objecten worden aangemaakt
  + Kan de startpositie van de toekan worden bepaald
  + Kan een maximale tijd worden ingesteld om het level te halen

## 3.2 Should have

Dit zijn de onderdelen die wel in het spel moeten zitten, maar deze onderdelen zijn niet essentieel voor de werking van het spel. Deze onderdelen hebben de op een na hoogste prioriteit. Hieronder een lijst van de onderdelen in deze categorie;

* De basegame:
  + Bestaat uit twee extra modes:
    - “Stealth” mode
    - Verzamel mode
  + Bestaat uit effecten/power-ups
  + De speler
    - Kan de effecten/power-ups oppakken
    - Kan met bepaalde effecten/power-ups vijanden verslaan
  + Heeft geluid-effecten
* In de game-maker:
  + Kan de speler een bepaalde achtergrond toevoegen aan een level
  + Kan er een keuze worden gemaakt uit de drie verschillende modus
  + Kunnen texturen groter worden aangegeven dan het object zelf
  + Kan de speler de dimensies van de map bepalen
* Kan de speler de geluid-effecten uitschakelen
* Kunnen spelers levels delen over het internet
* Kan de speler de resolutie bepalen

## 3.3 Could have

Dit zijn onderdelen die aan bod komen als er tijd genoeg is om ze te maken en ze hebben dus de laagste prioriteit. Hieronder een lijst van de onderdelen in deze categorie;

* Extra modus:
  + Zombie mode  
    Hier worden er oneindig vijanden in de map gezet en de toekan heeft altijd een bepaald effect waarmee vijanden verslagen kunnen worden. De vijanden achtervolgen de toekan
* Extra toevoegingen aan objecten:
  + Een objecten breekt na dat erop gelopen is
  + Vijanden mogen alleen op objecten bewegen (dan ziet het er echter uit)
* Animaties toevoegen aan objecten
* Spelers zelf hun texturen laten toevoegen
* Procedurele generatie voor de mappen (alternatief of toevoeging aan de database) op basis van ‘seeds’ (contributors W. , Procedural generation, 2021)

## 3.4 Wont have

Dit zijn onderdelen die niet aan bod komen, maar eventueel interessant kunnen zijn voor een vervolgproject, of een andere uitbereiding. Hieronder een lijst van de onderdelen in deze categorie;

* Multiplayer
* Een aanpasbare level-modus buiten de level-maker
* 3D of hoger
* Zelfgemaakte objecten

Cross-platform (Alleen Windows platformen) (contributors W. , cross platform software, 2021)

* Ondersteuning van andere randapparatuur (alleen keyboard)
* Grafische kwaliteits-configuratie
* < 5.5

# 4. Mock-ups

In dit hoofdstuk worden er een aantal Mock-ups van de applicatie tentoongesteld. De schermen die tentoonstelt worden zijn:

* Het Splash-screen
* Het hoofdmenu
* De opties
* Het maken van een level
* Het spelen van een level

Er is gekozen voor een toekan als hoofdthema voor de applicatie. Hierbij is gekozen voor jungle/berg thema, want dat komt overeen met de leefomgeving van de toekan.

## 4.1 Het Splash-screen

Dit is het laadscherm dat wordt weergegeven wanneer de applicatie wordt opgestart door de gebruiker. Het heeft een laadbalk, zodat de gebruiker kan zien hoever de applicatie is met opstarten. Het nut van dit scherm is om de gebruiker een conformatie geven dat de applicatie wordt uitgevoerd.



Afbeelding 1 Splash-screen

## 4.2 Het Hoofdmenu

Het Hoofdmenu zal de hoofdscherm van de applicatie zijn, wat inhoudt dat vanaf deze pagina door de applicatie heen genavigeert kan worden. Het bevat de knoppen die leiden naar het spelen van het spel, het creeëren van een spel, de opties en de mogelijkheid om de applicatie te sluiten.



Afbeelding 2 Hoofdmenu

## 4.3 Het opties

Het opties scherm geeft de gebruiker wat opties om de applicatie naar zijn/haar voorkeuren af te stemmen. Er is gekozen voor drie verschillende opties waartoe de gebruiker toegang heeft, namelijk de scherm modus, de resolutie en het volume. Hiervoor is gekozen, omdat deze drie opties de meeste belangrijkste opties leken zodat elke gebruiker de applicatie comfortabel kan draaien.



Afbeelding 3 Opties

## 4.4 Het creatie scherm

In het creatie scherm zal de gebruiker de mogelijkheid hebben om een eigen level te creëren. De gebruiker zal de toegang hebben tot meerdere verschillende soorten blokken, vijanden, power-ups. Ook zal de gebruiker toegang hebben tot andere extra middelen zoals bomen, wolken, stenen en zelfs een waterval. Het doel van dit scherm is dat de gebruiker op een creatief en vrije manier een level kan creëren.



Afbeelding 4 Creëer een level

## 4.5 Het Speel scherm

In het speel scherm kan de gebruiker het uitgekozen level spelen. De gebruiker gebruikt de toekan als karakter om het doel van het level te behalen.

Afbeelding 5 Het spelen van een spel

# 5. Klassendiagram

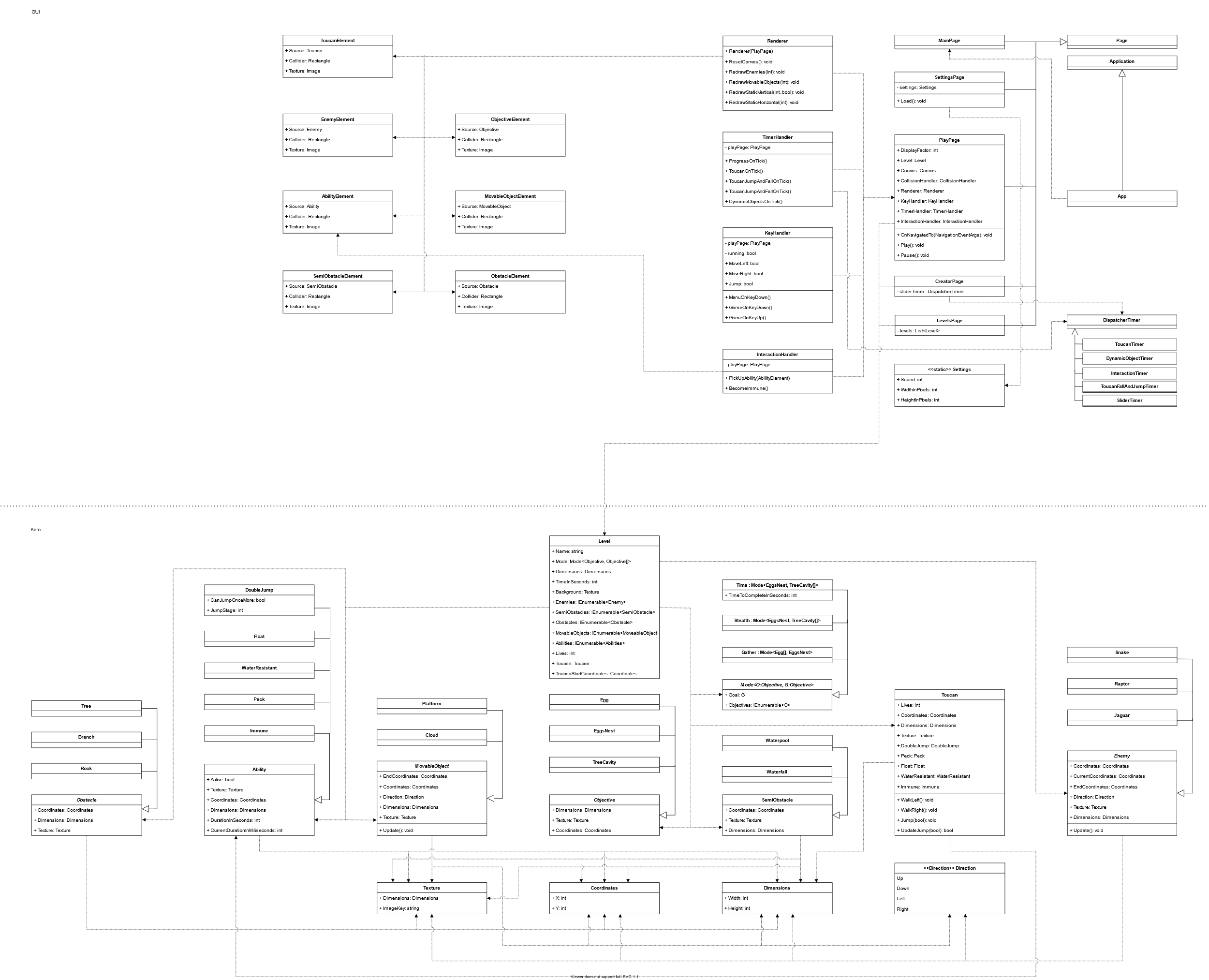
In afbeelding 6 wordt het klassendiagram getoond.

De GUI gerelateerde objecten staan aan de bovenste zijde. De applicatie bestaat uit meerdere pagina’s, namelijk:

* ‘SettingsPage’
* ‘LevelsPage’
* ‘CreatorPage’
* ‘PlayPage’
* ‘MainPage’

Al deze pagina’s hebben hun eigen functies. De ‘PlayPage’ is de pagina die waarin het level wordt gespeeld. De ‘CreatorPage’ is verantwoordelijk voor het maken van een level. Beide pagina’s zijn afhankelijk van de ‘Renderer’ klasse. Daarnaast is de PlayPage afhankelijk van ‘DispatcherTimer’ deze timers roepen iedere 16 ms een speciale klasse aan die de logica afhandelt voor de desbetreffende timer. ‘ToucanTimer’ zorgt voor de beweging van toekan. De ‘DynamicObjectTimer’ zorgt voor de beweging van alle dynamische objecten en de ‘InteractionTimer’ zorgt voor de interactie tussen de toekan en objecten. Daarnaast kan het zo zijn dat er meerdere timers worden toegevoegd voor bijvoorbeeld het bijhouden van de voortgang in het spel.

Onder het gedeelte voor de GUI wordt het gedeelte van het klassendiagram getoond van de Kern van de applicatie. Logica voor de beweging en interactie tussen de toekan en andere objecten wordt verwerkt in ‘walk left’, ‘walk right’ en ‘jump’ methodes, hetzelfde geldt voor de beweging van ‘moveable object’, dat wordt verwerkt in de ‘update’ methode. Daarnaast is te zien dat er ook veel lege klassen afhankelijk zijn van de Object klasse. Dit zorgt ervoor dat in de ‘CreatorPage’ specifieke objecten kunnen worden geselecteerd. Dit kan ook worden gerealiseerd door middel van een string met daarin de object type/naam van het object met daarnaast een constante lijst van alle object types (als string), maar de projectgroep heeft gekozen om klassen te gebruiken.



Afbeelding 6 Klassendiagram (inzoomen voor duidelijkheid)

# 6. Planning

Afbeelding 9 Planning

# Verwijzingen

4ian/GDevelop. (2021). *GitHub*. Opgehaald van GDevelop: https://github.com/4ian/GDevelop

bijdragers, W. (2019, November 4). *Microsoft SQL Server*. Opgehaald van Wikipedia: https://nl.wikipedia.org/wiki/Microsoft\_SQL\_Server

Bonzon. (2021, Maart 31). *How to Build a Complete 2D Platformer in Unity*. Opgehaald van GameDev Academy: https://gamedevacademy.org/how-to-build-a-complete-2d-platformer-in-unity/

contributors, W. (2021, Mei 25). *.NET Core*. Opgehaald van Wikipedia: https://en.wikipedia.org/wiki/.NET\_Core

contributors, W. (2021, Mei 27). *cross platform software*. Opgehaald van Wikipedia: https://en.wikipedia.org/wiki/Cross-platform\_software

contributors, W. (2021, Juni 1). *Model view viewmodel*. Opgehaald van Wikipedia: https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93viewmodel

contributors, W. (2021, Mei 5). *Procedural generation*. Opgehaald van Wikipedia: https://en.wikipedia.org/wiki/Procedural\_generation#:%7E:text=In%20computing%2C%20procedural%20generation%20is,create%20textures%20and%203D%20models

contributors, W. (2021, Mei 7). *Universal WIndows Platform*. Opgehaald van Wikipedia: https://en.wikipedia.org/wiki/Universal\_Windows\_Platform

GDevelop. (2021). *Create games without programming*. Opgehaald van gdevelop-app: https://gdevelop-app.com/

GitHub. (2021). *Github*. Opgehaald van Build software better, together: https://github.com/about

Godot. (2021). *C# basics*. Opgehaald van Godot Engine Documentation: https://docs.godotengine.org/en/stable/getting\_started/scripting/c\_sharp/c\_sharp\_basics.html

Godot. (2021). *Free and open source 2D and 3D game engine*. Opgehaald van Godot Engine: https://godotengine.org/

Microsoft. (2021). *.Net documentation* . Opgehaald van Microsoft Docs: https://docs.microsoft.com/en-us/dotnet/fundamentals/

Microsoft. (2021). *UWP applications*. Opgehaald van Microsoft Docs: https://docs.microsoft.com/en-us/windows/uwp/

MonoGame. (2021). *MonoGame*. Opgehaald van https://www.monogame.net/

MonoGame. (2021). *MonoGame*. Opgehaald van GitHub: https://github.com/MonoGame/MonoGame

Nintendo. (2019). *Op Mario kun je bouwen!* Opgehaald van Nintendo: https://www.nintendo.nl/Games/Nintendo-Switch/Super-Mario-Maker-2-1514009.html

Wikipedia. (2021, April 23). *Super Mario Maker 2*. Opgehaald van Wikipedia: https://nl.wikipedia.org/wiki/Super\_Mario\_Maker\_2

Wikipedia. (2021, Juni 1). *Wikipedia*. Opgehaald van Model view viewmodel: https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93viewmodel

Wikipedia-bijdragers. (2020, Maart 11). *How to build a complete 2D platformer in unity* . Opgehaald van Wikipedia: https://nl.wikipedia.org/wiki/Platformspel

Wikipedia-bijdragers. (2020, November 19). *Wikipedia*. Opgehaald van Wikpedia: https://nl.wikipedia.org/wiki/Super\_Mario\_Maker

Wikipedia-bijdragers. (2021, Mei 16). *GitHub*. Opgehaald van Wikipedia: https://nl.wikipedia.org/wiki/GitHub

Wikipedia-bijdragers. (2021, April 30). *MosCoW-methode*. Opgehaald van Wikipedia: https://nl.wikipedia.org/wiki/MoSCoW-methode