Technisch ontwerp

Game Opdracht

Gemaakt door:

Ahsen Kiliç

Cédric van Lenthe

Mohammed Abo Khaled

20 oktober 2022

Versie: 1.1

Versiebeheer

|  |  |  |  |
| --- | --- | --- | --- |
| **Versienummer** | **Datum** | **Auteur** | **Veranderingen** |
| 1.0 | 17-10-2022 | *Ahsen Kiliç* | *Document gemaakt en aangevuld met data* |
| 1.1 | 20-10-2022 | *Ahsen Kiliç* | *Document Ingevuld en requirements aangepast.* |

Contents

[Inleiding 4](#_Toc117169958)

[Over dit document 4](#_Toc117169959)

[Wat is Seadric? 5](#_Toc117169960)

[Seadric wereld 5](#_Toc117169961)

[Besturing 5](#_Toc117169962)

[Klassendiagram 6](#_Toc117169963)

[Toelichting klassendiagram 7](#_Toc117169964)

[Spelapp | Klasse en gerelateerde subklassen. 7](#_Toc117169965)

[Componenten | Herbruikbare klassen 7](#_Toc117169966)

[Spelkarakters | Klassen 8](#_Toc117169967)

[Bibliography & Bronnen 10](#_Toc117169968)

# Inleiding

## Over dit document

In dit document wordt een eerste uitwerking van de wensen en eisen van de opdrachtgever vastgelegd, zodat hiermee aan de opdrachtgever kan worden uitgelegd hoe het project eruit komt te zien en basaal hoe de werking van het programma zal zijn.

# Wat is Seadric?

Seadric is een spel ontwikkeld met behulp van de Java yaeger game-engine.

Het idee van dit spel is gebaseerd op het echte spel Feeding Frenzy (feeding-frenzy, n.d.), In dit spel draait alles om het eten van vissen, hoe meer vissen je personage eet, hoe groter en sneller het wordt, zodat je kunt concurreren en grotere vissen kunt eten, en zeeschatten kunt verzamelen.

De uitdaging in dit spel is om zoveel mogelijk vissen te eten, om een nieuwe hoge score te behalen, de grotere uitdaging is de roofvis die je niet kunt eten, maar er wel van weg kunt zwemmen.

# Seadric wereld

De game vindt een diepe plek in de Indische Oceaan, waar de hele omgeving vissen, schatten en mijnen is.

De speler heeft het 2D-perspectief (2D\_computer\_graphics, 2022), om van de kleurrijke objecten te genieten.

In het spel zie je verschillende soorten objecten en obstakels, zoals:

* Vissen die het personage kan consumeren.
* Zeeschatten die het personage kon verzamelen.
* Zeemijn die je karakter doodt, wanneer je hem raakt

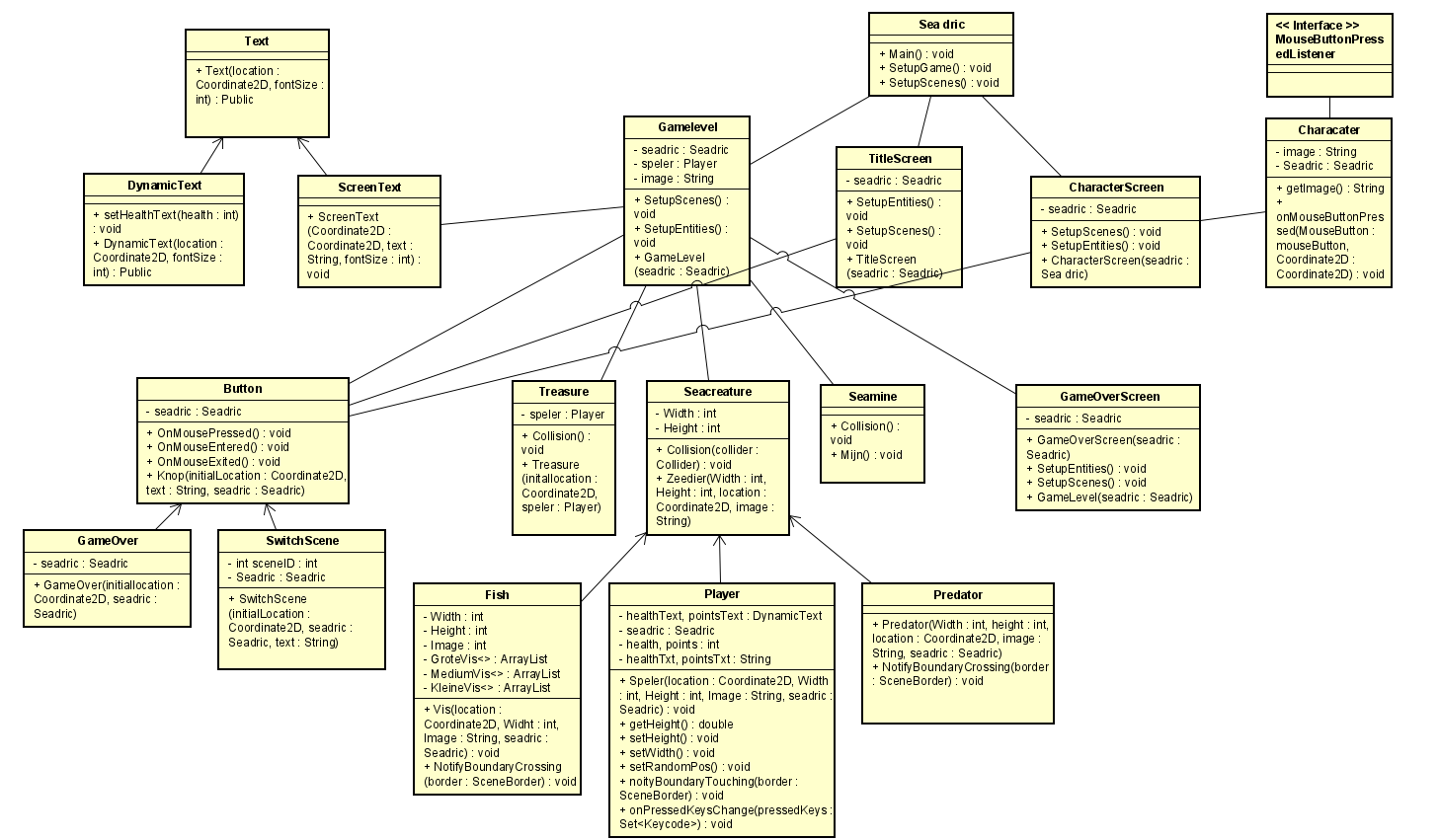
Predator is een moordende vis die je personage niet kon eten, maar er wel van weg kon zwemmen.

# Besturing

Om je karakter te laten bewegen in Seadric moet je de knoppen W, A, S, D gebruiken, dit hebben we gekozen omdat (WASD) een van de meest bekende en makkelijke besturing is.

# Klassendiagram

De opgestelde klassendiagram voor Seadric.



# Toelichting klassendiagram

## Spelapp | Klasse en gerelateerde subklassen.

**Spelapp**  
De spelApp is de main constructor van het hele spel. Hier wordt alles gestart en aangeroepen.

Main()  
Dit is de aanroep/start van het programma. Hier wordt het titelscherm, karakterscherm en spelscherm aangeroepen. Deze zullen toegankelijk zijn op basis van de interactie met het programma.

**Schermen**

*Bijzonderheden schermen:*

De schermen zullen gebruik maken van componenten als Knoppen en Tekst.

**TitleScreen**  
Titelscherm is het eerste scherm dat de speler zal zien wanneer het spel wordt gestart

**CharacterScren**   
Hier heeft de speler de mogelijkheid om tussen 3 karakters te kiezen

**Game Level**

Dit is het scherm waar het spel start. Hier worden alle vis, bom en schatkist elementen opgeroepen om de omgeving op te maken.

*Bijzonderheden:*

Om de uiterlijk van het vis op te stellen voordat het spel is opgemaakt moet er gebruik gemaakt worden van een static String type. Via het CharacterScreen word de waarde van de nieuwe afbeelding toegekend en aangepast in het gameLevel scherm.

**GameOverScreen**

Dit is het scherm waar de speler kan kiezen of het verder wilt spelen of het spel wilt beëindigen.

## Componenten | Herbruikbare klassen

**Superklasse : Button**

Dit is de basis opmaak van een knop, maar niet elke knop voert dezelfde functie uit. Daarom hebben we van button een superklasse gemaakt waar we polymorfie voor toepassen. De volgende subklasse zijn:

* GameOver
* SwitchScene

**Subklasse**

Gameover

Deze knop beindigt het spel wanneer hierop wordt geklikt.

**SwitchScene**

Afhangend welke parameters en waarde verstuurd zijn naar deze knop wordt er tussen de schermen in gegaan om bij het gewenste scherm te komen.

**SuperKlasse : Tekst**

Dit is de basis van elke tekstcreatie, maar niet elke tekst heeft dezelfde uitvoer. Waar de een enkel informatie toont, verandert de ander op basis van opgegeven parameters.

**Subklasse**

DynamicText

Deze tekst houdt de punt of levenswaarde bij van de speler terwijl dit spel wordt bespeeld. Op basis van de ontvangen parameters zal de waarde aangepast worden.

**ScreenText**

Deze tekst is om aan te geven waar de speler zich bevindt bij het uitvoeren van de programma. Bijvoorbeeld: Hoofdscherm, Game over of ter introductie van het spel “Seadric”.

**OnMousePressed()**

De actie geeft de gebruiker de mogelijkheid te switchen tussen schermen.

**OnMouseEntered()**

Wanneer de cursor op de knop valt zal de button van kleur veranderen om aan te tonen dat deze knop actief is.

**OnMouseExited()**

Wanneer de cursor de knop verlaat zal de button in de originele staat terugkregen

## Spelkarakters | Klassen

**Superklasse**  
**Seacreature**  
Dit is de superklasse van alle vissen in het spel, hier wordt de basis van de vis aangemaakt. De volgende subklasse zijn hiermee verbonden.

**Waarde**

Breedte en hoogte. Deze waarde maken de standaard viskarakter aan voor het spel.

**Subklasse**

**Fish**  
Dit zijn alle vissen waar de speler mee kan interacteren. Deze vissen zullen verdeeld worden over grootte, breedte en welke interactie zal plaatsvinden afhangend van de situatie en omvang. Elke vis die wordt opgemaakt wordt op basis van de grootte en breedte verdeeld worden over 3 arraylijsten.

* Groot
* Middel
* Klein

Afhangend van de grootte van de vis zal hierbij de juiste zwemsnelheid toegepast worden. Maar ook bij het consumeren van deze vis zal een bepaalde puntentelling opgesteld worden.

**Player**  
Dit is de vis die bestuurd zal worden door de speler, met een andere uiterlijk dat te onderscheiden is van de andere vissen. Deze viskarakter kan niet zoals de andere vissen verder zwemmen dan het scherm groot is, deze zal tegen de randen opbotsen en moet andere kleine vissen opeten om sterker te worden zodat hij de grootste vissen kan consumeren.

Maar er zijn uitzonderingen: De speler kan geen predator opeten en het spel zal beëindigd worden als de vis tegen een mijn aanzwemt.

**Predator**  
Dit is de vijand van het spel. Deze kan niet opgegeten worden en is bedoelt om het spel moeilijker te maken voor de speler.

In de superklasse zal de volgende abstracte methode toegepast worden.

**Methodes**

**Collision()**  
Bij collision() wordt er gekeken naar de grootte van de twee objecten die elkaar tegenmoet komen. Dat wat kleiner is zal opgegeten worden door dat wat groter is.

**Treasure**De schatkist draagt een x aantal punten/waarde die de puntentelling van het spel bevordert.

Collision()  
Bij collision zal dit object verdwijnen.  
  
**Mijn**  
Dit is een zwevend explosief. Als hiertegen wordt gezwommen wordt het spel beëindigt.

Collision()  
Bij collision ontploft de bom en wordt het spel beëindigt.

# Bibliography & Bronnen

*2D\_computer\_graphics*. (2022, oct 06). Retrieved from en.wikipedia.org: https://en.wikipedia.org/wiki/2D\_computer\_graphics

*feeding-frenzy*. (n.d.). Retrieved from www.ea.com: https://www.ea.com/nl-nl/games/feeding-frenzy/feeding-frenzy

https://github.com/han-yaeger/yaeger