Technisch Ontwerp

Dr. Doctor vs. Corona



Namen: Stefan Teunissen & Thomas van Minnen

Studentnummers: 652054 & 652055

Klas: ITA-1M

Docent: Paksha Thullner

Versie: 1.0

Datum: 17-10-2020

Course: OOPD

**Inhoudsopgave**

Inhoud

[1 Over dit document 3](#_Toc54614756)

[1.1 Over het project 3](#_Toc54614757)

[2 Klassen beschrijving 4](#_Toc54614758)

[2.1 Klassen bij de eisen 5](#_Toc54614759)

[3 Klassediagram 6](#_Toc54614760)

# Over dit document

Het doel van het technisch ontwerp is om inzicht te krijgen in de verschillende technieken die gebruikt gaan worden bij ontwikkelen van de gevraagde software. Daarnaast zal er een duidelijk beeld worden geschetst hoe de verschillende technieken met elkaar verbonden zijn.

Dit document is voor deze opdracht gemaakt zodat eenieder die betrokken gaat worden bij dit project makkelijk zijn weg zal vinden in de al bestaande architectuur en softwarefuncties.

## Over het project

# Klassen beschrijving

|  |  |
| --- | --- |
| Klassen | Beschrijving |
| Buttons | De klasse *Buttons* maakt een knop object aan, en heeft ook een methode om te kijken of de muis boven de knop is. |
| ExitButton | De klasse *ExitButton* maakt een knop aan door gebruik te maken van de klasse *Buttons*. Ook controleert de klasse of hij wordt ingedrukt, als dit zo is zal hij het spel afsluiten. |
| LevelsButton | De klasse *LevelsButton* maakt een knop aan door gebruik te maken van de klasse *Buttons*. Ook controleert de klasse of hij wordt ingedrukt, als dit zo is zal hij naar het level menu gaan. |
| LevelSelectButton | De klasse *LevelSelectButton* maakt een knop aan door gebruik te maken van de klasse *Buttons*. Ook controleert de klasse of hij wordt ingedrukt, als dit zo is zal hij het geselecteerde level starten. |
| MainMenuButton | De klasse *MainMenuButton* maakt een knop aan door gebruik te maken van de klasse *Buttons*. Ook controleert de klasse of hij wordt ingedrukt, als dit zo is zal hij terug gaan naar het start menu. |
| AnimatedLoopObject | De klasse AnimatedLoopObject zorgt er voor dat objecten geanimeerd kunnen zijn, dit houdt bij ons bijvoorbeeld in dat een munt rondjes draait terwijl hij naar links of rechts beweegt. |
| Background | De klasse *Background* maakt een background sprite object aan door middel van de klasse *SpriteObject* te gebruiken. Deze background kan vervolgens in een level/ menu worden gebruikt. |
| Block | De klasse *Block* maakt een block object aan door middel van de klasse *SpriteObject* te gebruiken. Dit block object kan vervolgens in een level worden gebruikt. |
| Coin | De klasse *Coin* maakt een coin object aan door middel van de klasse *SpriteObject* te gebruiken. Dit coin object kan vervolgens in een level worden gebruikt. |
| Finish | De klasse *Finish* maakt een finish object aan door middel van de klasse *SpriteObject* te gebruiken. Dit finish object kan vervolgens in een level worden gebruikt. |
| Heart | De klasse *Heart* maakt een heart object aan door middel van de klasse *SpriteObject* te gebruiken. Dit heart object kan vervolgens in een level worden gebruikt. |
| IFlyingObject | De interface *IFlyingObject* zorgt er voor dat andere klassen die *IFlyingObject* extenden de functies er in moeten gebruiken. |
| ImageObject | De klasse *ImageObject* zorgt er voor dat er een plaatje in het spel kan, bijvoorbeeld Trump op het level menu. |
| TextPanel | De klasse *TextPanel* kan gebruikt worden om tekst op het scherm te zetten. |
| BackgroundHandler | De klasse *BackgroundHandler* zorgt er voor dat de achtergronden bewegen. |
| Level | De klasse *Level* regelt alles wat te maken heeft met de levels zoals: het beginnen van het alarm voor de klasse *ObjectSpawner* en het regelen van de score en highscore en het opslaan hiervan. |
| Main | De klasse *Main* zorgt voor het opstarten van het spel en het initialiseren er van. |
| MenuState | De klasse *MenuState* is een enum, en dit bevat meerdere variabelen die maar één keer gevuld kunnen worden(zoals een final). |
| ObjectSpawner | De klasse *ObjectSpawner* zorgt er voor dat er in een level blokken en munten komen. De klasse roept de klassen *block, coin, heart* aan en zorgt er voor dat deze in de wereld verschijnen. |
| Player | De klasse *Player* regelt alles wat te maken heeft met het Hoofdpersonage zoals: de besturing, de levens er af en het knipperen na het krijgen van pijn. |
| SoundHandler | De klasse *SoundsHandler* zorgt voor de geluiden die er in het spel horen zoals: het oppakken van een munt, het pijn krijgen en bij het halen van de finish. |
| TextObject | De klasse *TextObject* maakt van een stuk tekst een object, waardoor deze bewogen kan worden. Dit is terug te zien op de aftiteling |

## Klassen bij de eisen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Eisen** | **Benodigde klassen** | **Must have** | **Should have** | **Could have** | **Won’t have** |
| Main menu | Main, level | X |  |  |  |
| Level menu |  | X |  |  |  |
| Verschillende levels |  | X |  |  |  |
| Blokken komen op willekeurige locaties |  | X |  |  |  |
| Blokken zorgen voor een leven er af |  | X |  |  |  |
| Finish aan het einde van het level | Finish, | X |  |  |  |
| Verschillende achtergronden | BackgroundHandler |  | X |  |  |
| Level progressie |  |  | X |  |  |
| Munten zorgen voor +1 bij score | Coin |  | X |  |  |
| Highscore wordt opgeslagen |  |  |  | X |  |
| Aftiteling | TextObject |  |  | X |  |
| Hoofdpersonage is na het raken van een blok tijdelijk onsterfelijk |  |  |  | X |  |
| Munten waardoor je een leven terug krijgt |  |  |  | X |  |
| Eigen geluiden |  |  |  | X |  |
| Tijd per level op de aftiteling |  |  |  | X |  |
| Munten hebben een ronddraaiende animatie |  |  |  | X |  |
| Tegenstanders |  |  |  |  | X |
| Hoofdpersonage kan vaccins schieten |  |  |  |  | X |

# Klassediagram

Hieronder vind u het klassediagram zoals wij verwachten dat hij er uit gaat zien. Het kan zijn dat er aan het einde van het project verschillen zijn tussen het klassediagram en de code zelf, deze verschillen zullen aan worden gegeven in het

