|  |
| --- |
| Technisch Ontwerp Jumper\_  [ondertitel] |

Kevin Noppers

27 maart 2024

INHOUDSOPGAVE

[1 Klassen diagram 4](#_Toc162552810)

[2 Toelichting klassen diagram 4](#_Toc162552811)

[2.1 Buttons 4](#_Toc162552812)

[2.1.1 StartButton 4](#_Toc162552813)

[2.1.2 QuitButton 4](#_Toc162552814)

[2.1.3 TryAgainButton 5](#_Toc162552815)

[2.1.4 Finish 5](#_Toc162552816)

[2.1.5 Speler 5](#_Toc162552817)

[2.1.6 Rocket 5](#_Toc162552818)

[2.1.7 Platform 5](#_Toc162552819)

[2.1.8 BluePlatform 5](#_Toc162552820)

[2.1.9 GreenPlatform 6](#_Toc162552821)

[2.1.10 BrownPlatform 6](#_Toc162552822)

[2.1.11 BombPlatform 6](#_Toc162552823)

[2.1.12 PlatformSpawner 6](#_Toc162552824)

[2.1.13 GameOverScene 6](#_Toc162552825)

[2.1.14 GameScene 6](#_Toc162552826)

[2.1.15 TitleScene 7](#_Toc162552827)

[2.1.16 ScoreText 7](#_Toc162552828)

[2.1.17 Jumper 7](#_Toc162552829)

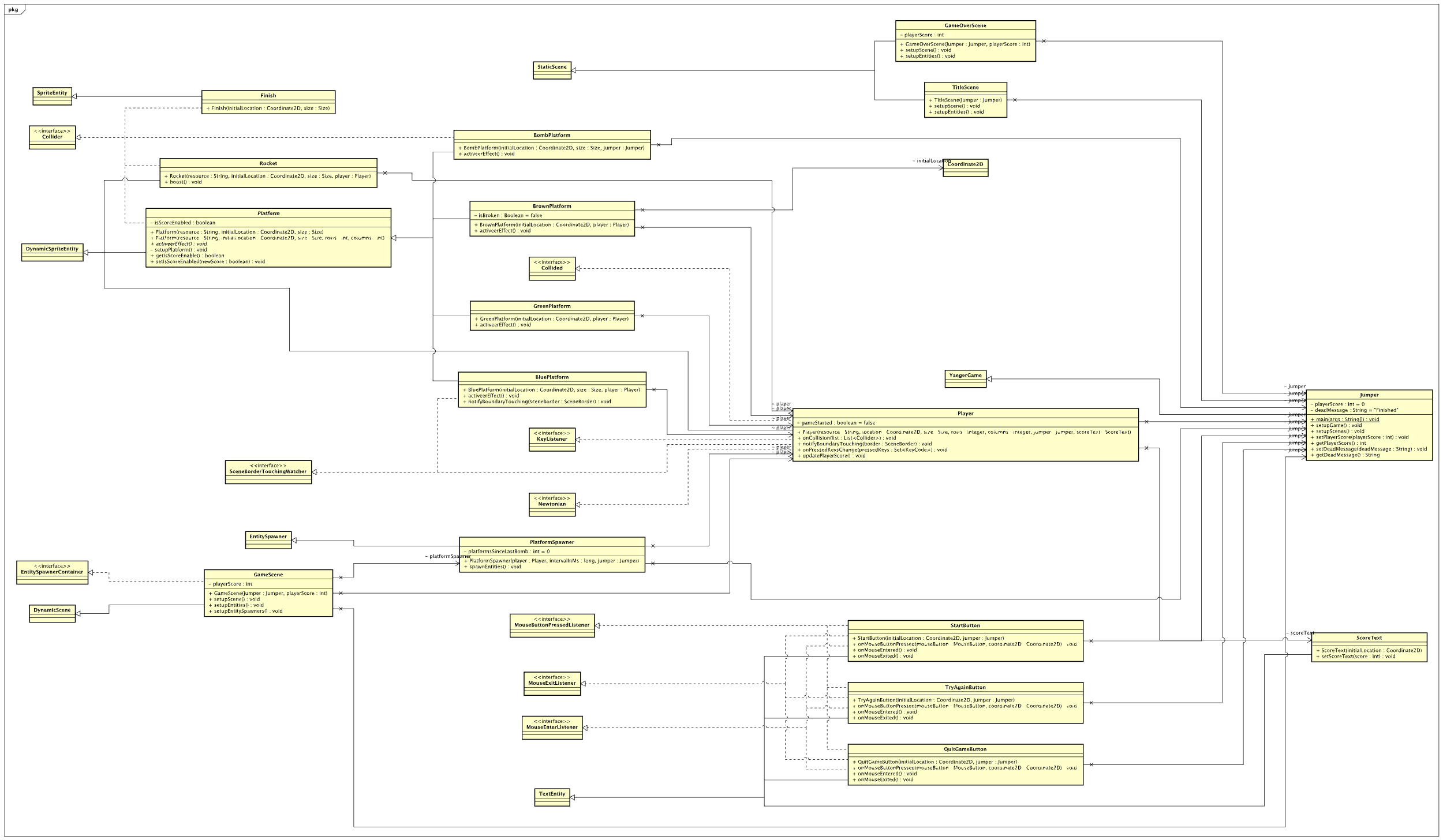
[3 Terugkoppelen MoSCOW 8](#_Toc162552830)

INLEIDING

In dit document worden de klassen toegelicht die zijn gebruikt voor het spel ‘Jumper’ In dit bestand zul je een toelichting terugvinden van de verschillende classes en belangrijke methoden daarvan. Ook is er in dit bestand een klasse diagram terug te vinden die een duidelijk overzicht moet geven over alle klassen en interfaces die zijn gebruikt voor het spel ‘Jumper’.

# Klassen diagram

Het klassen diagram is geëxporteerd als SVG. In de zip folder van dit project is ook de afbeelding te vinden mocht het onduidelijk zijn in dit document



# Toelichting klassen diagram

In dit hoofdstuk worden de belangrijke klassen en methode omschreven.

## Buttons

### StartButton

De startbutton extend de klasse TextEntity en implementeert MouseButtonPressedListener, MouseEnterListener, MouseExitListener

Hiermee kan het spel gestart worden

### QuitButton

De QuitButton extend de klasse TextEntity en implementeert MouseButtonPressedListener, MouseEnterListener, MouseExitListener

Hiermee kan het spel gestopt worden

### TryAgainButton

De TryAgainButton extend de klasse TextEntity en implementeert MouseButtonPressedListener, MouseEnterListener, MouseExitListener

Hiermee kan het spel opnieuw gestart worden als de speler af is.

### Finish

De finish extendsSpriteEntity en implementsCollider.

Als de speler deze entity aanraakt zal het spel gestopt worden met een resultaat als winnen

### Speler

De speler extends DynamicSpriteEntity en implements KeyListener, SceneBorderTouchingWatcher, Newtonian, Collided

Met de methode onCollision word er gekeken van welk type het is, finish, platform of een raket. Als de speler een van deze aanraakt word er een bepaalde handeling uitgevoerd.

De methode updatePlayerScore, update de score van de speler als hij een nieuw platform heeft aangeraakt.

### Rocket

De Rocket extends DynamicSpriteEntity implements Collider

Als de speler een raket raakt wordt de methode boost() uitevoerd en word de speler ver de lucht in geschoten en word daarna de enity verwijderd uit het spel.

### Platform

Platform is een abstracte klasse die extendsDynamicSpriteEntity en implementsCollider

Is platform abstracte klasse heeft een boolean isScoreEnabled waarbij word bijgehouden of het nodig mogerlijk is om een score te krijgen van het platform

De methode activeerEffect() is een abstracte klasse die voor elk verschillende platform andere functie heeft.

### BluePlatform

BluePlatform extendsPlatform implementsSceneBorderTouchingWatcher

Dit platform verplaatst zichzelf van links naar rechts van het scherm zogauw hij de zijkanten heeft aangeraakt. Hierop word ook de methode activeerEffect gedaan. Als deze word uitgevoerd word de speler de lucht in geschoten

### GreenPlatform

GreenPlatform extendsPlatform

Dit platform gaat alleen van boven naar beneden de scene. Hierop word ook de methode activeerEffect gedaan. Als deze word uitgevoerd word de speler de lucht in geschoten.

### BrownPlatform

BrownPlatform extendsPlatform

Dit platform gaat ook van boven naar benenden de scene. Als de speler dit platform aanraakt dan word de activeerEffect() uitevoerd. Bij deze zal het platform breken en is het niet langer mogerlijk dat de speler nog een keer gebruik maakt van dit platform. Ook word de speler de lucht in geschoten.

### BombPlatform

De BombPlatform extendsPlatform

Dit platform gaat ook van boven naar beneden in de scene. Als de speler dit platform aanraakt dan word de activeerEffect() uitevoerd. Bij deze zal de bom ontploffen en gaat de speler dood. De speler word naar het gameover scene gestuurd.

### PlatformSpawner

De PlatformSpawner extendsEntitySpawner

Deze klasse genereerd alle platforms die er zijn. De platformen worden op een random locatie op de X as gezet boven aan de scene. Belangrijk detail is dat er altijd 4 normale platformen moeten komen voordat er weer een nieuwe bom platform mag komen. Dit is gedaan om te zorgen dat het niet te moeilijk word voor de speler.

### GameOverScene

De GameOverScene extendsStaticScene

Deze klasse geeft aan wanneer het spel klaar is, er zijn 3 mogerlijkheden voor de speler om het spel te beindigen. Dat is door de finish te raken en te winnen, van alle platformen op de grond te vallen. En door een bom aan te raken waardoor je afgaat.

### GameScene

De GameScene extendsDynamicScene implementsEntitySpawnerContainer

Dit is de klasse waar het spel zich afspeeld hier worden de entitys aangemaakt in op de goede locatie gezet.

### TitleScene

De TitleScene extendsStaticScene

Dit is het beginscherm waar de speler op aankomt. Hier is de keuze om het spel te starten of om het spel af te sluiten.

### ScoreText

De ScoreText extendsTextEntity

Deze klasse laat zien aan de speler hoeveel punten hij heeft gemaakt. Via de methode setScoreText word de text weergegeven op het scherm en de nieuwe score geupdate.

### Jumper

De Jumper extendsYaegerGame

Dit is de Main klasse van het spel. Hierin word het spel aangemaakt en de scenes neergezet. Ook word hierbijgehouden wat de precise score en de deadmessage is.

# Terugkoppelen MoSCOW

|  |  |
| --- | --- |
| Nummer | Omschrijving |
| MH1 | Er zijn minimaal 8 eigenklassen geïmplementeerd: QuitGameButton, StartGameButton, TryAgainButton, Finish, Player, Rocket, BluePlatform, BrownPlatform, GreenPlatform, Platform, BombPlatform, PlatformSpawner, GameOverScene, GameScene, TitleScene, ScoreText, Jumper |
| MH2 | Zelfgebouwde methoden hebben comments. \*Behalve de getters en setters. |
| MH3 | Er wordt colision interface gebruikt op de speler, Rocket, en de platformen |
| MH4 | Er wordt overerving toegepast binnen de zelfgemaakte klasse Platform en bijvoorbeeld BrownPlatform |
| MH5 | Er wordt Polymorfie toegepast aan de klasse Platform en bijvoorbeeld BrownPlatform |
| MH6 | Van de klassen is vanuit de naam te bedenken wat die klassen kan inhouden. |
| MH7 | De code is zelf geschreven is overal uit te leggen. |

