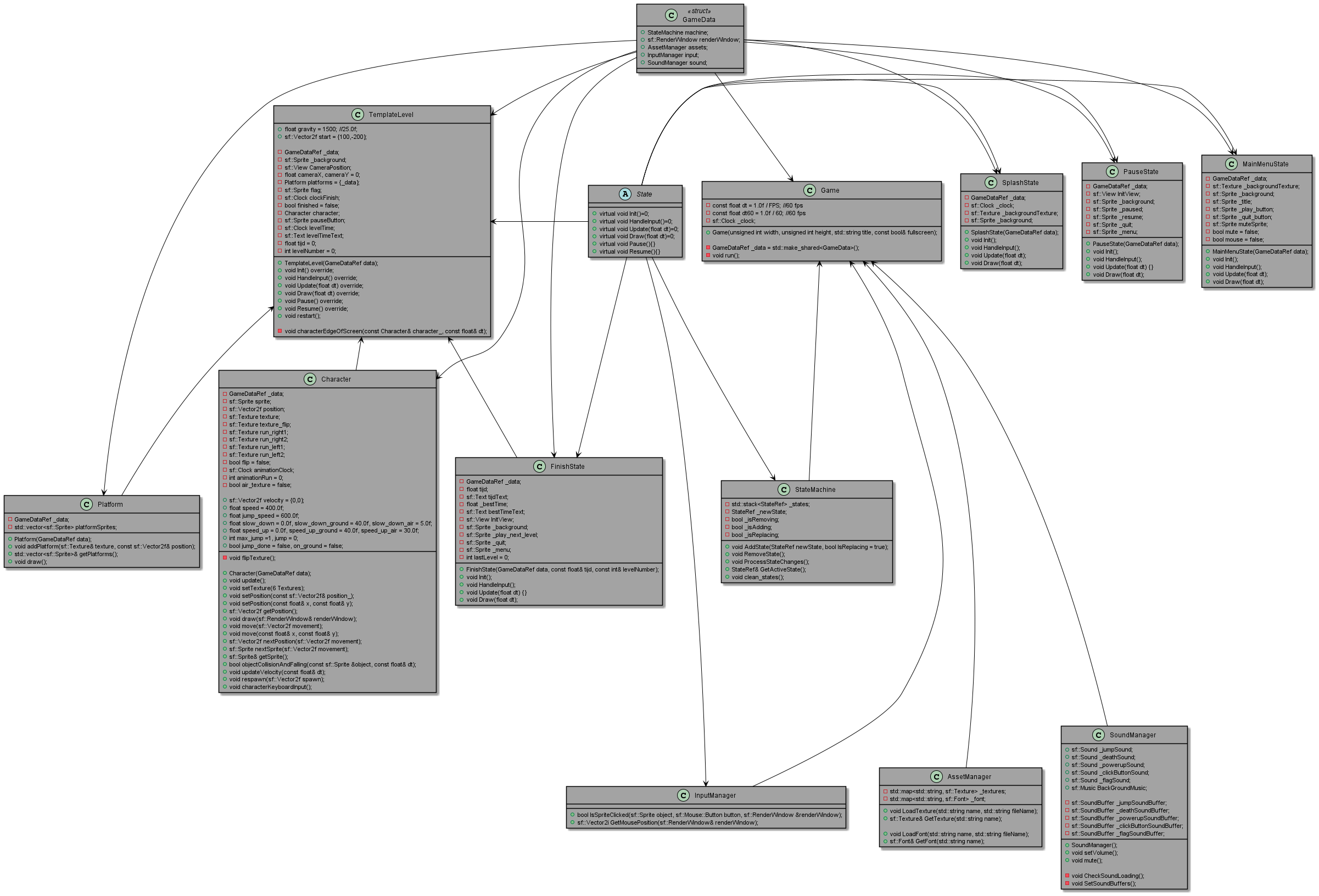
Technisch Verslag:

**Daan Kerbusch, Roel Stierum en Joris Kunkeler**

28-01-21

## Klassendiagram:



Hier een link naar het klassendiagram als het niet leesbaar is:<https://imgur.com/a/vbR42Ys>

## 

## Keuzes:

We hebben vrijwel geen onduidelijke keuzes gemaakt. Het klassendiagram geeft een duidelijke weergave van de klassenstructuur. Ook zijn er geen complexe algoritmen in de code aanwezig.

## Doxygen:

We hebben doxygen toegevoegd aan alle .hpp bestanden, om onze structuur zo duidelijk en begrijpbaar mogelijk te maken. In de .cpp bestanden staan comments om de code nog een keer/verder toe te lichten. Zo is de code goed te (her)gebruiken door anderen.

## Extra uitleg:

De collision tussen de character en de platformen wordt getest in een functie van de character class. Die functie geeft een boolean terug. Als de character met geen enkel platform collide zal deze boolean false returnen. Als de boolean false is wordt bij de y van de velocity de zwaartekracht waarde afgetrokken, zodat de character valt. Hoe langer de character valt hoe sneller hij dus valt omdat elke keer de zwaartekracht waarde van de y velocity wordt afgetrokken.

De character positie wordt geupdate door middel van de velocity vector. Eerst wordt de velocity geupdate. Aan het eind, als we alles hebben gedaan met de velocity, roepen we updateVelocity() functie aan in de character class. Dit zal de positie van de charater verplaaten en zo nodig de texture omdraaien.

Voor het maken van een nieuw level moet de code van TemplateLevel .cpp (wel moet de include van de hpp worden gewijzigd) en hpp worden gekopieerd naar de nieuwe level bestanden (cpp en hpp). Daarin moet de levelNumber member worden aangepast naar de eerstvolgende. Ook moet in DEFINITIONS.hpp het aantal levels worden verhoogt.