

Retro VideoGame Sphaeristerium

Student: Pim Horeman

Studentnummer: 3032469

Kernmodule-1, GDV

Docenten: Aaron Oostdijk, Valentijn Muijers

Inlever datum: 19-09-2019

Concept & Twist:

Voor mijn Retrogame met heb ik gekozen voor een fusie tussen Arkanoid en Pinball. Ik begon met arkanoid en hoe ik dat wilden gaan veranderen en toen kwam ik al snel op een idee om niet de bal te laten beginnen vanaf het midden, wat gevolgen bracht op een combinatie met pinball.

De twist van het spel is dat het gecombineerd is. Je maakt gebruik van beide spellen in 1. Daarnaast is Arkanoid een 2d spel en dit zet ik nu in een 3d omgeving. Pinball is officieel een arcade machine die nu in een virtuele omgeving gebracht wordt.

Doordat het spel een 3d spel is, heb je ook meteen te maken met een Vector3 in plaats van een Vector2 (in de meeste gevallen). Wat het programmeren weer interessant maakt.

Code Patterns

Ik heb voor dit project gebruik gemaakt van de Observer Pattern.

Observer Pattern:

Het Observer Patterns is in de game geïmplementeerd door middel van interfaces en actions.

Ik maak gebruik van 1 interface (IReset) wat voor nu gebruikt wordt met de ball class. Hierbij kan je aangeven waar een object gereset moet worden in een vector3 positie.

Ik heb veel gebruik gemaakt van Actions in het spel wat het afvuren van events heel makkelijk maakte. Ik zie het als: een class schreeuwt een word en een andere class luistert alleen of dit bij hem hoort en als dat zo is dan activeert die een functie.

Recap:

Ik vond het lastig om te weten wanneer je interfaces nou echt moet toepassen. Actions daarentegen gaat mij een stuk makkelijker af. Ik heb beide wel gebruikt en ik kan wel het nut zien van interfaces.

Code Structuur

Hier wordt er informatie gegeven over classes die interessant waren in het project. Niet alle classes zullen worden benoemd.

Brick

De Brick class wordt gebruikt voor de prefab 'brick' hierin zit een dictionary wat rekening houdt met wat voor enum brick het is en welke kleur eraan verbonden zit. Door een FSM zijn er verschillende staten waarin de brick zich kan bevinden. Wanneer de brick geraakt wordt zal de kleur veranderen.

BrickSpawner

De BrickSpawner class is een aparte class.

We geven in een int array cijfers mee van 0 tot 4. Deze cijfer zijn getekent in een rij op de x as en een rij op de y as binnen de int array. Daarna koppelen de brick enum aan de cijfers en spawnen ze in rijen. Hierdoor als je het cijfer 1 neerzet krijg je een rode brick te zien en als je een 3 neerzet een groene brick.

Ball

De Ball class houdt zich alleen bezig met de bal en de physics die eromheen spelen.

De ball heeft een aantal publieke functies die aangeroepen kunnen worden vanuit andere classes. Het enige wat de ball doet is physics en force op zich laten komen. Hierdoor beweegt de bal door het level.

Als de bal de paddle raakt dan zal er een berekening plaats vinden op welke plek hij terecht kwam en hoe die er vanaf moet kaatsen.

ScoreManager

De ScoreManager class regelt de score in het spel. Er is een lokale score en een lokale highscore. De highscore wordt verzonden naar de unity playerprefs functie die eventueel later op te halen is. De playerprefs heeft ook een aparte key die prive is zodat niet iemand de playerprefs kan aanpassen.

IReset

IReset is een interface wat ervoor zorgt dat je een object kan resetten naar een vector3 positie. De vector3 positie heeft een getter en setter om ook de positie aan te passen.

SpringLauncher

De SpringLauncher class wordt gebruikt om tijdens het starten van het spel de bal af te vuren in de game. Hier wordt gebruikt gemaakt van de spatiebalk om dit te activeren. De Ball class wordt ook opgevraagd om een van de publieke functies te gebruiken om force mee te geven als de bal afgeschoten wordt.

De SpringLauncher gebruikt een plunger(GameObject) en trekt deze eerst naar achteren en dan veert die terug naar boven en schiet de bal af.

ParticleManager

De ParticleManager class wordt geactiveerd via Actions van buiten af. Een ander script geeft een enum mee met wat de type particle het wordt en een transform voor de positie.

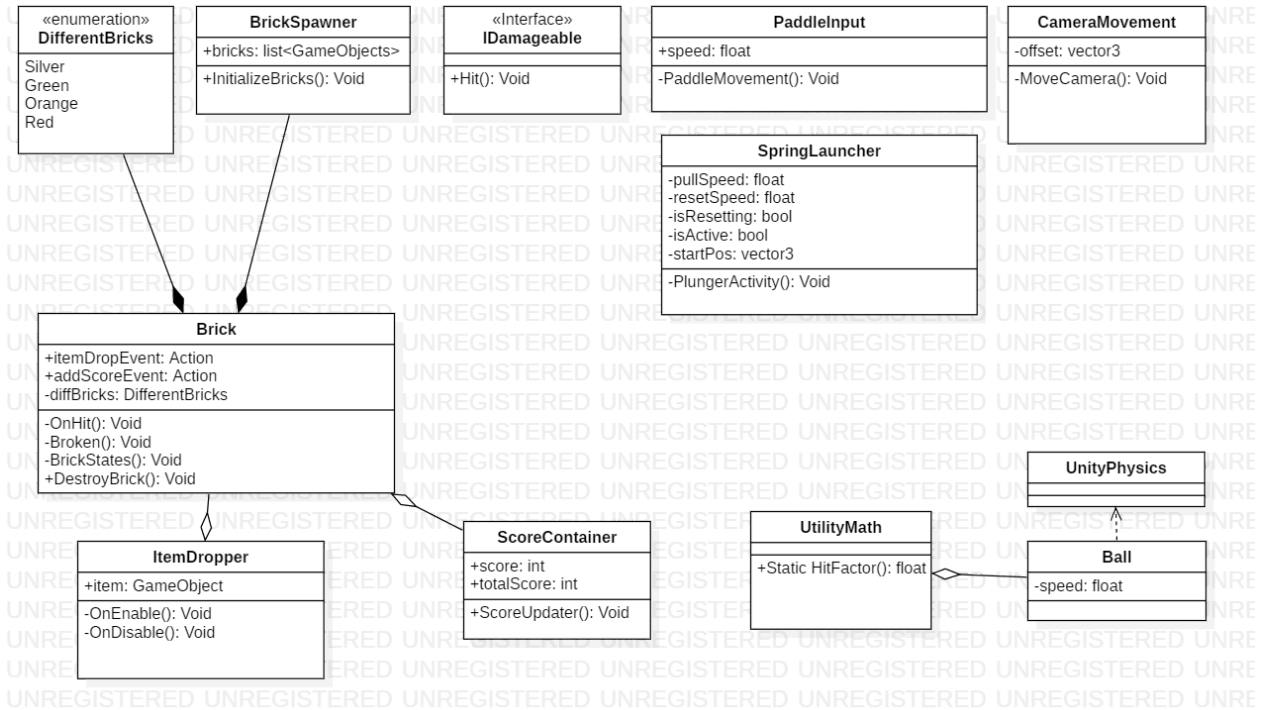
Daarna wordt de particle spawned. Het ontvangt alleen actions.

UtilityMath

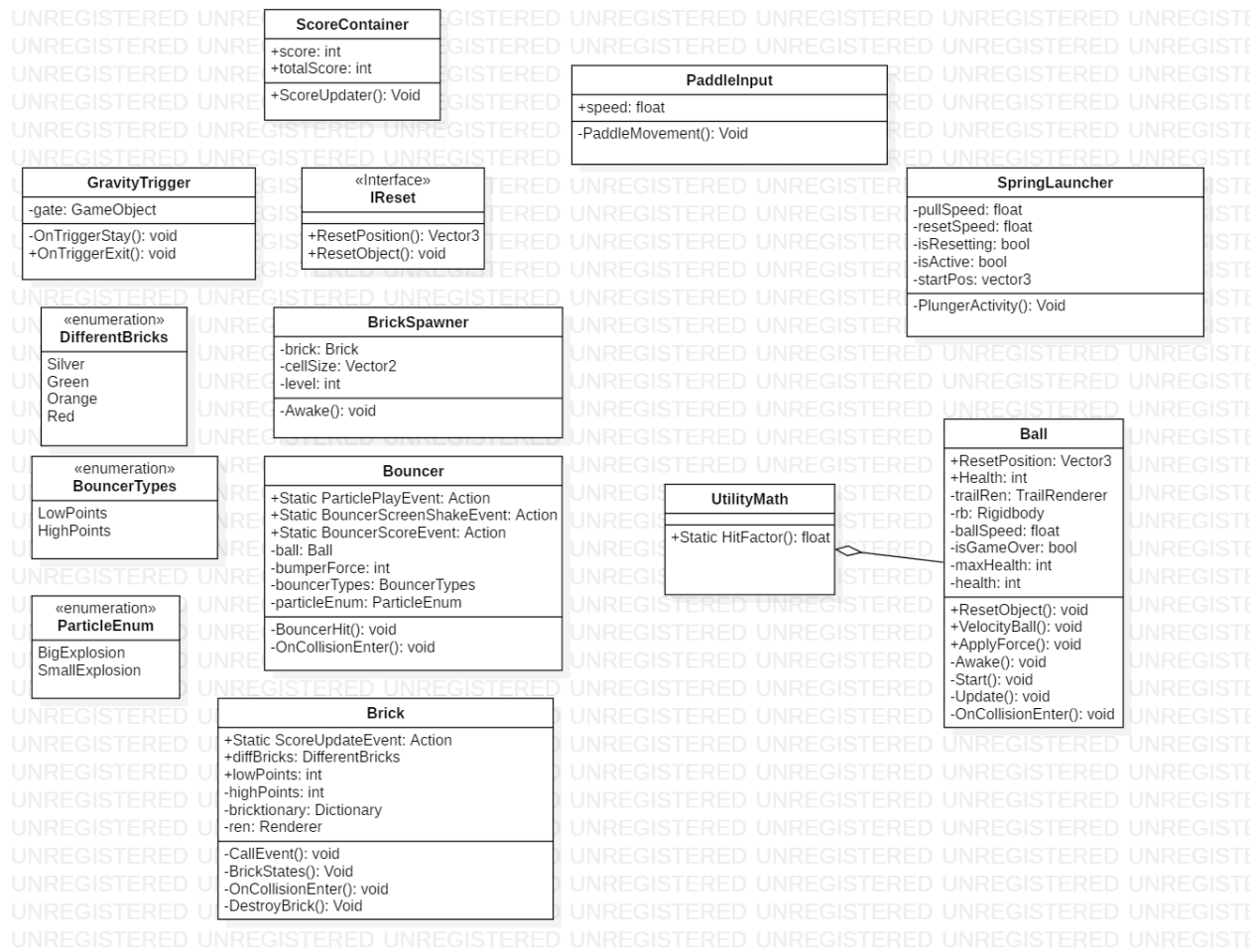
UtilityMath class is een statische class.

Hierin wordt een berekening gemaakt in dit geval speciaal voor de bal. Hierin wordt de lengte van een object min de deling van 2 posities berekent.

UML Class Diagram van code structuur



Niet compleet UML met nieuwe classes:



UML Activity Diagram

