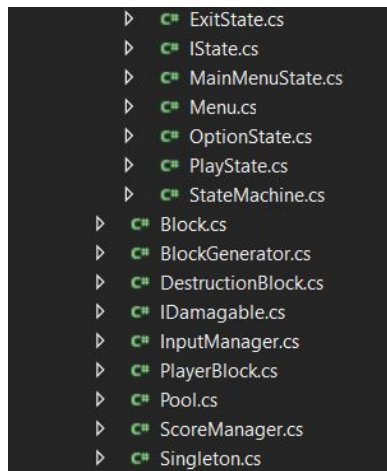


Wheatris  
Geoffrey Hendrikx  
3028316  
Kernmodule Game Development  
19/09/2019

Het Concept.

Het concept is gebaseerd op Tetris alleen dan met een hele grote twist. In tetris moet je blokken op elkaar zetten om zo een rij vrij te spelen. In dit spel moet je juist zorgen dat je zoveel mogelijk tetris steentjes houdt. Na een aantal stenen verwisselen de rollen en gaat een AI proberen random steentjes weg te halen. Dit doet de AI door gewoon zijn steentje op jouw steentje te gooien met zwaartekracht.

### Toelichting Code Structuur



StateMachine zorgt in dit spel de afwisseling van menu panels.

IState alles wat een state nodig heeft aan functies en events staat hierin.

ExitState

MainMenuState eerste menu wordt gemaakt door deze class.

Menu

OptionState Fullscreen en volume hier.

PlayState Naar de playstate

---

### Blocks

Block Generator deze klas krijgt een generic mee en op basis daarvan spawned die een block.

Destruction Block Deze class gebruik ik om AI blocks te spawnen.

IDamagable een van mijn interfaces die ik heb gebruikt. Deze interface gebruik ik voor objecten damage te geven.

Player Block niet helemaal hetzelfde als de destruction block. deze worden aangemeld bij de input manager zijn delegate en zijn dus ook volledig bestuurbaar.

---

### InputManager Observer pattern

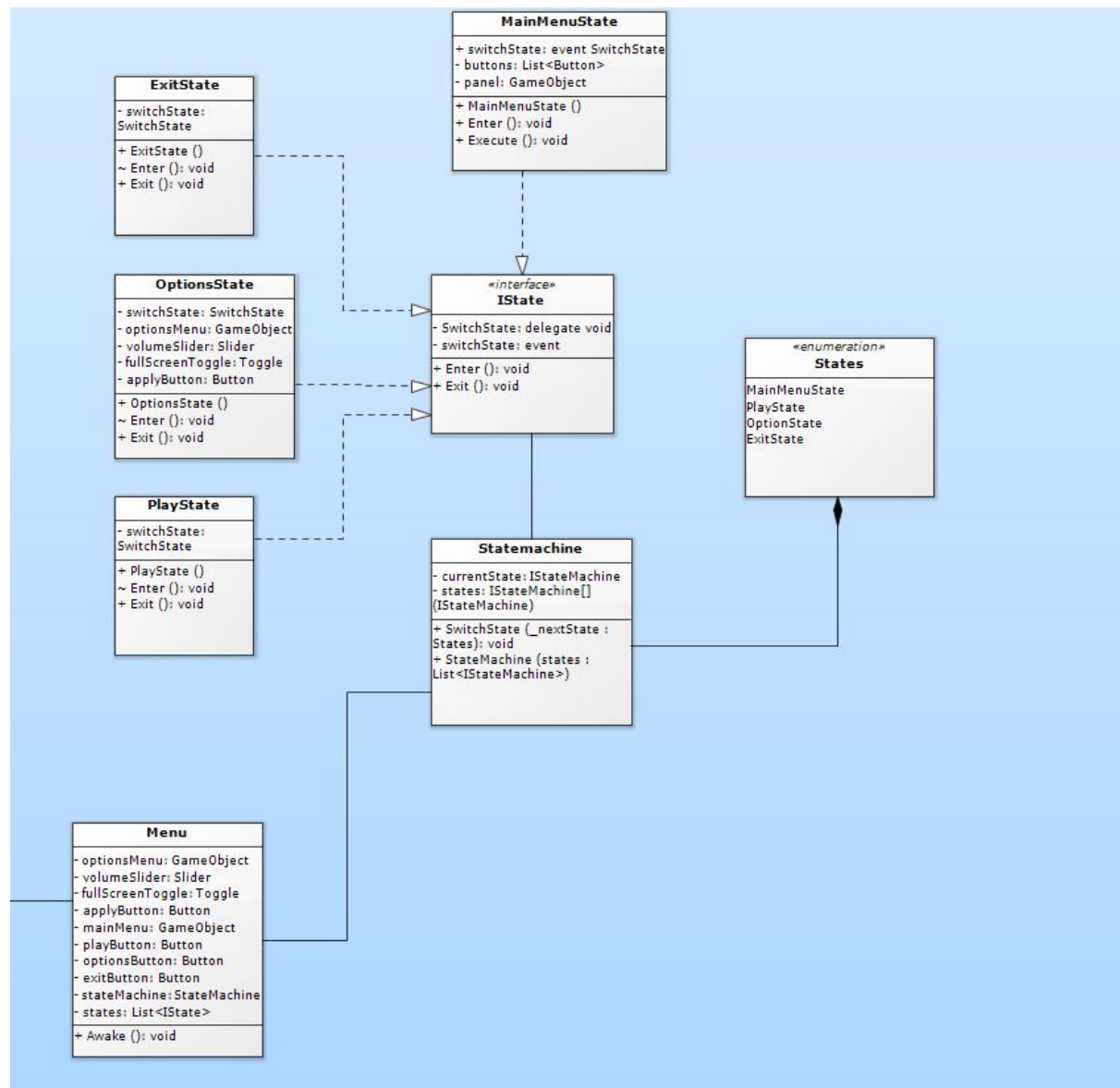
Pool object pooling om zo efficiënt mogelijk te werken.

ScoreManager - Update the GUI score and shows the end screen

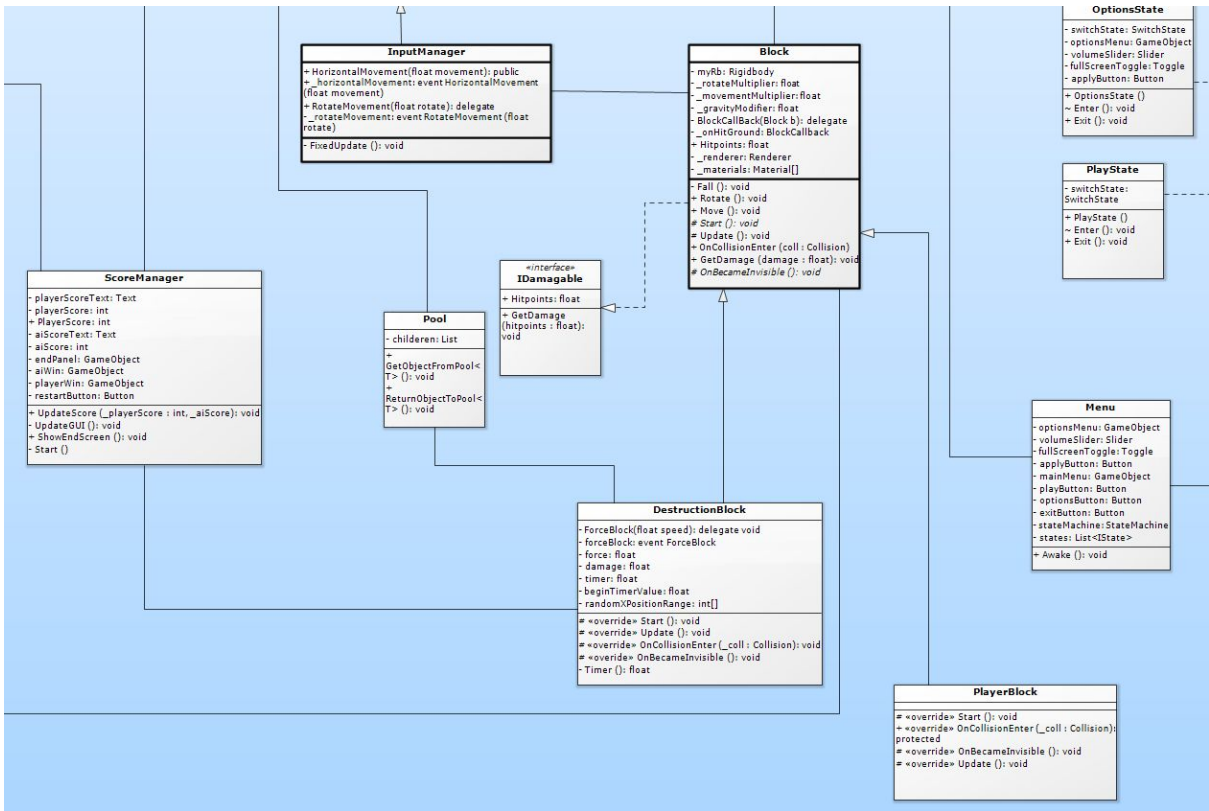
Singleton - zorgt ervoor dat er maar 1 instance in de game is. de classes InputManager Pool en ScoreManager hebben ze

UML Ik deel mijn UML in 3 foto's op omdat die simpelweg te groot is.

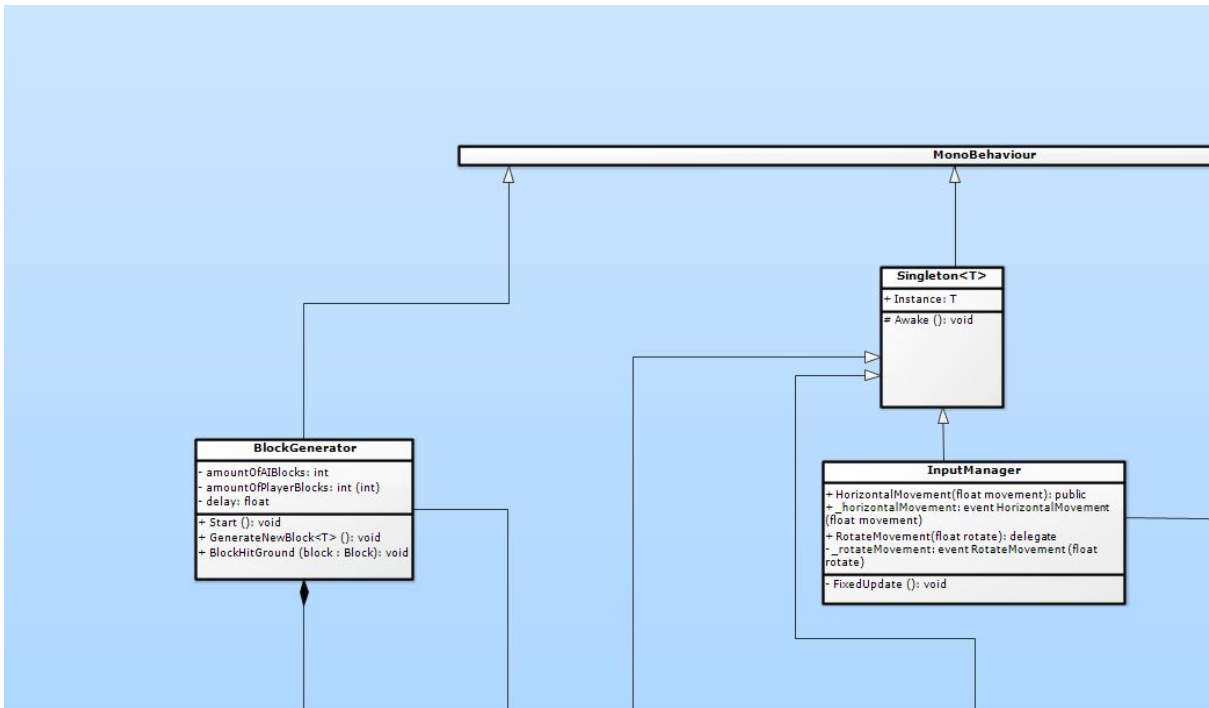
## StateMachine



blocks + score



BlockGenerator



Activity diagram

