Design patterns

Student: Thomas Uijlen

Inhoud

[inleiding 3](#_Toc114658675)

[Class Diagram 4](#_Toc114658676)

[ontwerpkeuzes en motivatie 6](#_Toc114658677)

[Event Queue 6](#_Toc114658678)

[Decorator pattern 6](#_Toc114658679)

[Factory pattern 6](#_Toc114658680)

[Command Pattern 6](#_Toc114658681)

# inleiding

Voor de opdracht wilde ik een terrain editor implementeren. Het terrein bestaat uit zelf geïmplementeerde heightmap terrain. Dit terrein kan ingame aangepast worden met een brush. Een brush kan meerdere modus hebben die makkelijk gewisseld kunnen worden.

Verder moet de brush ook een undo functie bevatten, waarbij alle aanpassingen van een brush stroke teruggedraaid kunnen worden.

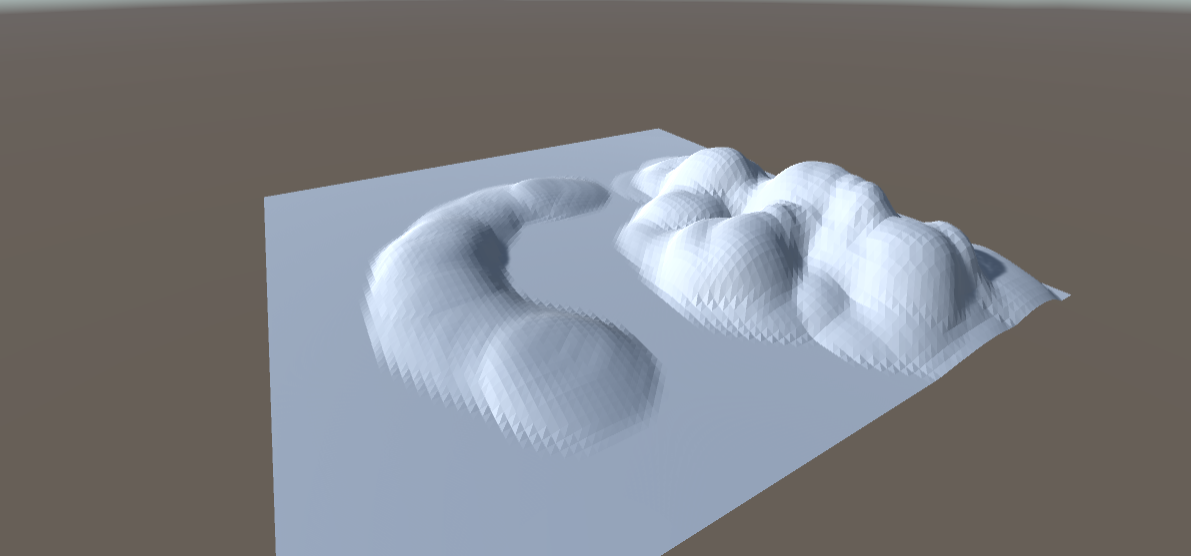
Voor deze opdracht wilde ik 4 patterns implementeren:

* Command Pattern
* Event Queue Pattern
* Decorator Pattern
* Factory Pattern

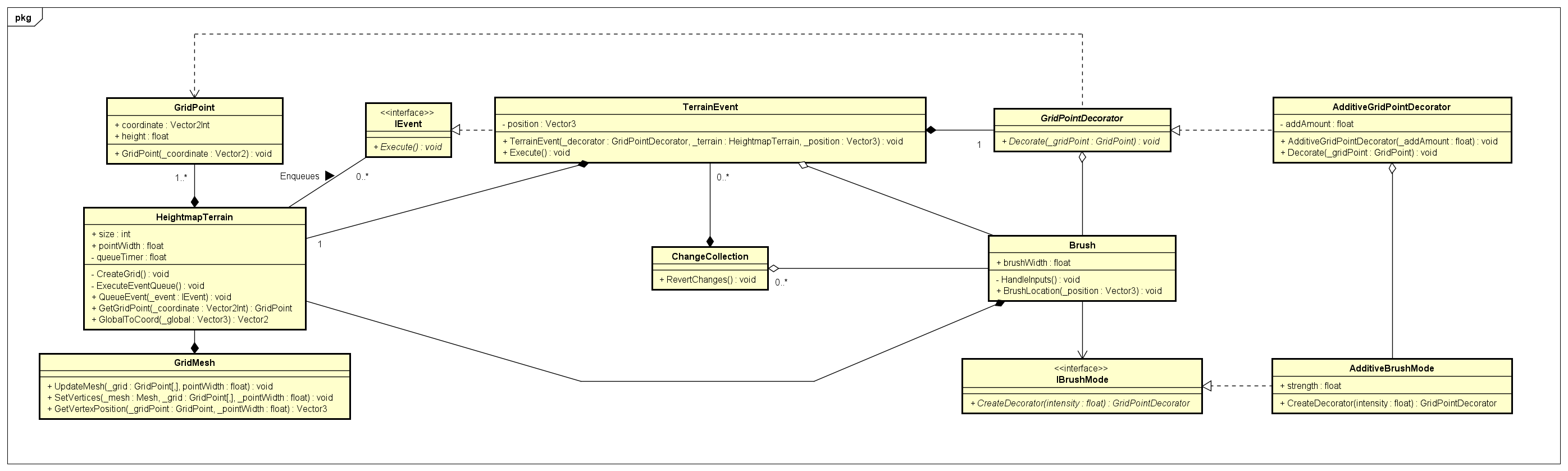
## Besturingen

In de Github repository is een build te vinden in de Build folder. Hieronder zijn de controls:

* W: Vooruit
* A: Links
* S: Achteruit
* D: Rechts
* Q: Naar beneden
* E: Naar boven
* Rechtermuisknop: Camera draaien
* Linkermuisknop: Brush



# Class Diagram



|  |  |
| --- | --- |
| Class | Beschrijving |
| HeightmapTerrain | Heightmap terrein. Bestaat uit een grid van GridPoints. De grootte van de grid kan ingesteld worden met de size variabel. De afstand tussen de GridPoints kan ingesteld worden met de pointWidth.  Heeft een ingebouwde event queue om events af te handelen. Deze events worden met een bepaalde interval afgehandeld. Na het afhandelen van de events wordt de GridMesh geüpdatet. |
| GridPoint | Een punt van de heightmap. Heeft een coördinaat en een height. |
| GridMesh | Converteert een lijst met GridPoints naar een lijst met Vertices. Met deze vertices wordt vervolgens een mesh opgebouwd om het terrain te renderen. |
| IEvent | Een interface voor een generieke event. |
| TerrainEvent | Een Event die het terrein aanpast. Bevat een GridPointDecorator. |
| GridPointDecorator | Een GridPointDecorator past een GridPoint aan. Bevat zowel een Decorate als UnDecorate functie. |
| AdditiveGridPointDecorator | Verhoogt de height variabel van een GridPoint. |
| Brush | Een simpele Brush. Handelt inputs en schiet een raycast voor de locatiebepaling van de brush. |
| IBrushMode | Een interface voor een brush modus. Via de IBrushMode kunnen GridPointDecorators opgevraagd worden. |
| AdditiveBrushMode | Deze brush mode creëert en retourneerd AdditiveGridPointDecorators. |

# ontwerpkeuzes en motivatie

Hier zal kort per gekozen design pattern toegelicht worden waarom deze gekozen is en wat het resultaat daarvan is.

## Event Queue

Als het terrein aangepast wordt moet de mesh opnieuw geconstrueerd worden. Dit kost relatief veel tijd. Vandaar dat het niet slim is om bij elke aanpassing of bij elke frame de mesh te updaten.

Vandaar dat ik gekozen heb voor een event queue. Aanpassingen worden in een queue opgeslagen. Vervolgens wordt deze event queue om een bepaalde interval uitgevoerd. Nadat de queue is uitgevoerd wordt de mesh geüpdatet. Hierdoor update de mesh niet elke frame, maar bijvoorbeeld elke 0.1 seconden.

Het resultaat hiervan is dat het aanpassen van het terrein veel minder zwaar is. Met elke frame updaten dropt de framerate naar onder de 60 fps. Bij het updaten met interval blijft de fps boven de 250++.

## Decorator pattern

Voor het aanpassen van het terrein maak ik gebruik van decorators. Dit zorgt ervoor dat het terrein heel snel op allerlei manieren aangepast kan worden. De decorator pattern werkt ook mooi samen met de volgende pattern.

## Factory pattern

De brush modus zijn geïmplementeerd als decorator factories. De Brush kan aan de actieve modus een GridPointDecorator opvragen, waarbij vervolgens de juiste decorator geretourneerd word.

Dit zorgt ervoor dat het effect van de brush modus volledig is losgekoppeld van de brush zelf. De brush weet niet wat hij doet of wat hij aanpast, hij weet alleen dat hij decorators doorgeeft aan het terrein.

Deze pattern maakt het ook extreem makkelijk om nieuwe brush modus toe te voegen of om de functionaliteit van een bestaande brush aan te passen of uit te breiden.

## Command Pattern

De command pattern word gebruikt voor de undo functionaliteit. TerrainEvents worden opgeslagen in een stack. Als de undo knop wordt ingedrukt wordt de meest recente changes van de stack afgehaald en ongedaan gemaakt.

# bijlagen

Github link: <https://github.com/ThomasUijlen/GameArchitectureDesignPatterns>