

Division Engine

Plan van aanpak

Auteurs:

Nick Dekker 14051079

Nick Schulte

Marthe Veldhuis 14103907

Tobi van Westerop

21-11-2017

Document versie 1.0

Inhoudsopgave

Inleiding	3
Opdrachtbeschrijving	4
Eisen en randvoorwaarden	5
Eisen	5
Randvoorwaarden	5
Afbakening	5
Aanpak	7
Producten	8
Organisatie	9
Planning	10

1. Inleiding

Dit document functioneert als plan van aanpak voor de ontwikkeling van de game engine "Division". Als deel van de minor 3D-engine design - de basis. Er wordt rekening gehouden met een fictieve klant, die deze eisen heeft opgesteld. Zo wordt er gewerkt naar een doel om de klant tevreden te maken met het uiteindelijke resultaat.

In het project wordt gestreefd naar een 3D game engine te bouwen met bijbehorende documentatie, die werkelijk kan worden gebruikt bij het ontwikkelen van een game. De engine zal het proces van een game maken vereenvoudigen en de developer tijd schelen.

In dit document wordt eerst een opdrachtbeschrijving gegeven. Daarna zijn de eisen en randvoorwaarden vastgesteld, waaraan het eindproduct zal voldoen. Ook wordt de aanpak verduidelijkt door de ontwikkelmethode en organisatie vast te stellen die door het team gebruikt zal worden. Vervolgens is een lijst met op te leveren producten te vinden in het hoofdstuk producten. Ten slotte zal de planning worden belicht.

2. Opdrachtbeschrijving

Dit project wordt in opdracht van de minor “3D engine design - de basis” uitgevoerd. Het doel van deze minor is om inzicht te krijgen in de basis van het ontwerpen en implementeren van een 3D engine. Het is de bedoeling dat de groep binnen 1 blok een engine oplevert dat is ontworpen met behulp van UML en ontwikkeld met C++ en DirectX 9. Er is een aantal minimale functies dat in de engine geïmplementeerd moet zijn zodat de projectgroep een voldoende kan halen. Ondanks dat een engine nooit af is, is het ons doel om genoeg functionaliteit te realiseren zodat er een goed product wordt opgeleverd. De uiteindelijke engine moet worden ondersteund door eenvoudig te begrijpen documentatie. Er zal worden gestreefd naar een uitbreidbaar systeem.

3. Eisen en randvoorwaarden

3.1 Eisen

Voor dit project heeft de klant een aantal eisen opgesteld waaraan voldaan moet worden. Deze eisen vormen een basis voor de gamewereld, die kan worden uitgebreid met extra features wanneer deze gevormd is. De eisen zijn als volgt.

- Ondersteuning voor meerdere windows, zodat de engine als level-editor benut kan worden
- Input van muis en keyboard, om de gebruiker te laten communiceren met de wereld
- Scene loader, om externe levels in te laden
- Skybox (lucht)
- Ground (terrain)
- Objecten (entities) om tussendoor te kunnen lopen, ook moeten deze textures kunnen hebben
- Logfile die het verloop van de engine weergeeft (info, warnings, error)
- Sandbox interface (voor expert-users)
- Simpele interface (voor beginners)

Dit zijn eigenschappen waarover de engine zelf beschikt. Code die hiervoor opgeleverd wordt zal ook documentatie bevatten, waarmee gebruikers van de engine wijs uit kunnen worden.

3.2 Randvoorwaarden

Naast de eisen zijn er een aantal randvoorwaarden waar rekening mee moet worden gehouden tijdens het maken van de engine. Deze zorgen ervoor dat indirecte implicaties worden voorkomen, zoals het vastlopen van het programma. De randvoorwaarden zullen ook dienen als functionaliteit om uitbreiding van de code mogelijk te maken. Ten slotte zijn er voorwaarden die voor dit project zijn opgesteld volgens de minor. Alle voorwaarden zijn:

- Abstracte renderer
- Geen memory leaks in het programma
- C++ voor de opbouw van de engine
- DirectX9 voor de 3D ontwikkeling
- Gebruiksgemak
- Uitbreidbaarheid
- Onderhoudbaarheid

3.3 Afbakening

De volgende punten hoeven tijdens dit blok niet te worden geïmplementeerd.

- Audio
- Physics

- Light sources
- Shaders
- Collision detection
- Een game

4. Aanpak

4.1 Algemene aanpak

Om een engine te ontwerpen zal er eerst goed gekeken moeten worden naar het verloop van een game. Op basis van de resultaten zal een ontwerp worden gerealiseerd in UML. Deze UML diagrammen zullen leiden tot code in C++.

Om gebruiksvriendelijkheid voor de klant te verzekeren, zal gekeken worden vanuit het perspectief van een game-designer. Zo wordt er constant rekening gehouden met zaken als uitbreidbaarheid en documentatie.

4.2 Methode

De engine is opgebouwd uit verschillende, duidelijk te onderscheiden onderdelen. Deze onderdelen kunnen daardoor in incrementen worden opgeleverd. Door deze aard van het project zal er met een agile development methode worden gewerkt om het project te managen. De methode Scrum zal worden gebruikt om sprints van 1 week realiseren. De backlog met user stories wordt op Visual Studio Team Services (VSTS) bijgehouden.

Voor de code zal met version control worden gewerkt. Dit zal worden gedaan via GitHub, waar branches aangemaakt worden om nieuwe features toe te voegen. Deze kunnen later worden gemerged naar de development branch en na testen naar de master branch. Zo wordt ervoor gezorgd dat de bestaande code blijft werken op de master branch.

Ten slotte wordt de volgende richtlijn voor de stijl van de code aangehouden (<http://geosoft.no/development/cppstyle.html>) om te zorgen dat de gehele code overzichtelijk en makkelijk te lezen blijft.

4.3 Organisatie

De projectgroep bestaat uit vier studenten: Nick Dekker, Nick Schulte, Marthe Veldhuis en Tobi Westerop. De groep zal drie keer per week een bijeenkomst houden om gezamenlijk de voortgang te bespreken en samen aan het project te werken. Eén keer per week zal er een presentatie worden gegeven over de voortgang in de klas.

5. Producten

De volgende producten worden aan het einde van het project opgeleverd.

Product	Beschrijving	Deadline
Plan van aanpak	Het huidige document voor de aanpak van het project	week 2
UML klassendiagram	Het ontwerp van de engine met alle functies Zorgt voor verduidelijking van de code-structuur in combinatie met de documentatie	week 7*
Engine	De complete library met alle code waaruit de engine bestaat	week 7
Documentatie API	Beschrijvingen van alle functies	week 7
Handleiding sandbox interface	Toelichting hoe gebruikt kan worden gemaakt van de engine voor ervaren gebruikers	week 7
Handleiding simpele interface	Toelichting hoe gebruikt kan worden gemaakt van de engine voor onervaren gebruikers	week 7

Tabel 1: producten

*Een opmerking over het UML klassendiagram is dat deze initieel in week 3 moet worden afgerond zodat deze kan worden gebruikt tijdens de ontwikkeling van de engine. Echter in de realiteit zal deze constant kleine veranderingen ondergaan, waardoor deze pas in week 7 compleet af is.

6. Planning

Onze planning is ingericht met 7 sprints. Elke nieuwe sprint wordt op woensdag gestart na de ingeplande expert vergadering. Hiervoor wordt uiteraard de voorgaande sprint afgerond. Tijdens de eerste sprint zal Nick Schulte niet deelnemen, maar zal in de volgende sprints deze gemiste tijd inhalen. Voor sprint 6 is extra tijd genomen omdat rekening wordt gehouden met de kerstvakantie. Wij zijn niet van plan om tijdens deze periode aan het project te werken.

Week	Data	Sprint
week 1	15 nov - 22 nov	sprint 1
week 2	22 nov - 29 nov	sprint 2
week 3	29 nov - 06 dec	sprint 3
week 4	06 dec - 13 dec	sprint 4
week 5	13 dec - 20 dec	sprint 5
week 6	20 dec - 10 jan	sprint 6
week 7	10 jan - 17 jan	sprint 7