

Projectdocument Minecraft Mod Builder

Projectgroep "Twintro"

11 december 2015

Inhoudsopgave

1	Probleemstelling	2
2	Productbeschrijving	2
3	Requirements analyse	3
3.1	Functional requirements	3
3.2	System requirements	3
4	Projectaanpak	4
5	Taakverdeling	5
5.1	Organisatie	5
5.2	Technisch	5
6	Planning	6

1 Probleemstelling

Minecraft¹ is een computerspel dat erg populair is, mede dankzij zogenaamde *mods*, dit zijn aanpassingen in het spel die door derden worden ontwikkeld. Het is mogelijk deze mods te ontwikkelen met behulp van MCP². Dat is een toolkit die wordt onderhouden door een van de ontwikkelaars van Minecraft, maar die niet officiëel wordt ondersteund. Mods werken alleen voor een specifieke versie van het spel en het tegelijk gebruiken van meerdere mods geeft vaak problemen. Om mods beter te laten samenwerken en het ontwikkelen eenvoudiger te maken is er Minecraft Forge³.

Wanneer gebruikers een mod in een nieuwe versie willen gebruiken, moeten ze dus eerst wachten tot MCP en eventueel Forge is aangepast en daarna tot de maker van de specifieke mod een update hiervoor uitbrengt. Ook vereist het maken van mods een uitgebreide ontwikkelomgeving en een redelijke kennis van de programmeertaal Java.

2 Productbeschrijving

Minecraft Mod Builder is een systeem waarmee gebruikers zonder programmeerervaring mods voor Minecraft kunnen ontwikkelen. Door middel van een editor met grafische interface kan de gebruiker mods samenstellen. Deze mods kunnen vervolgens geëxporteerd worden naar bestanden en door andere gebruikers ingeladen worden in het spel. De editor kan de bestanden ook weer inlezen om aanpassingen te maken. Om de mods in het spel in te laden wordt gebruik gemaakt van een generieke mod. Die heeft veel mogelijkheden en aan de hand van de bestanden wordt het gedrag bepaald.

Het systeem bestaat dus uit twee onderdelen; namelijk de grafische editor en een generieke mod. Het eerste onderdeel is vooral zichtbaar voor gebruikers en is alleen nodig om de mods te ontwikkelen. Het tweede onderdeel is nodig om de mods daadwerkelijk te gebruiken. De geëxporteerde mods kunnen zonder aanpassing in nieuwere versies van het systeem worden ingeladen, zodat de makers zelf geen updates hoeven uit te brengen.

¹<https://nl.wikipedia.org/wiki/Minecraft>

²Mod Coder Pack, <http://www.minecraftforum.net/forums/mapping-and-modding/minecraft-tools/1260561-toolkit-mod-coder-pack-mcp>

³http://www.minecraftforge.net/wiki/Minecraft_Forge

3 Requirements analyse

3.1 Functional requirements

1. Must have: de gebruiker kan nieuwe *blokken* (bouwstenen van de wereld) met gewenst uiterlijk en gewenste eigenschappen toevoegen.
2. Must have: de gebruiker kan nieuwe *items* (voorwerpen die gebruikt kunnen worden) met gewenst uiterlijk en gewenste eigenschappen toevoegen.
3. Must have: de gebruiker kan nieuwe *crafting recipes* (recepten om items uit andere items te maken) definiëren en toevoegen.
4. Should have: de gebruiker kan structuren toevoegen aan de manier waarop de wereld wordt gegenereerd.
5. Should have: de gebruiker kan definiëren wat er gebeurt bij bepaalde gebruikersacties of gebeurtenissen in het spel.
6. Should have: de gebruiker kan gebruikerscommando's definiëren.
7. Should have: de gegenereerde bestanden kunnen zonder moeite in toekomstige versies van het spel gebruikt worden.
8. Could have: de gebruiker kan programmeercode toevoegen voor geavanceerde logica.
9. Could have: de gebruiker kan *particles* (visuele effecten) met gewenst uiterlijk toevoegen.
10. Could have: de gebruiker kan geluidseffecten toevoegen.
11. Could have: de gebruiker kan de afbeeldingen die het uiterlijk van onderdelen bepalen binnen het programma aanpassen.
12. Won't have: de gebruiker kan aangepaste blokmodellen definiëren.
13. Won't have: de gebruiker kan gebruikersinterfaces binnen het spel toevoegen.

3.2 System requirements

1. Must have: de generieke mod wordt geladen via de *Minecraft Forge Mod Loader* (eenvoudige en meest gebruikte manier om mods de installeren).
2. Should have: het product maakt alleen gebruik van vrije software.
3. Should have: de editor werkt cross-platform.
4. Should have: het lezen en schrijven van bestanden gebeurt in het JSON-formaat.
5. Could have: de geavanceerde logica wordt geïmplementeerd met behulp van de scriptingtaal Lua.

4 Projectaanpak

We zullen de methodologie van SCRUM gebruiken bij het uitvoeren van dit project. Dit breekt het totale project op in kleinere sprints waarin werkende delen van het product worden ontwikkeld. Dit stelt ons in staat de software flexibel uit te breiden. De sprints duren zeven dagen, zodat we in een wekelijkse vergadering de voortgang kunnen bespreken en zo nodig de planning aanpassen.

Het platform waarop het product zal werken is Java, omdat dit ook is waarop Minecraft werkt, hierdoor zijn er voor gebruikers geen extra systeemvereisten. De ontwikkelomgeving die we gebruiken is Eclipse, omdat de toolkit van Forge daarmee goed samenwerkt⁴. Om samen aan de code te werken, gebruiken we het versiebeheersysteem Git. Na het toevoegen van een werkend deel en na belangrijke wijzigingen doen we een *push*. Voordat we ergens aan beginnen te werken doen we een *pull*. Hierdoor werkt iedereen aan de meest recente werkende versie en kunnen we, als dat nodig is, wijzigingen gemakkelijk terugdraaien.

Om bekend te worden met de MCP/Forge API en de mogelijkheden te onderzoeken, beginnen we met het ontwikkelen van een aantal voorbeelden van mods. Aan de hand hiervan kunnen we daarna de generieke mod implementeren en tegelijk de gebruikersinterface van de editor die daarbij nodig is ontwerpen. Uiteindelijk zullen we de generieke mod en editor koppelen en de functionaliteit verder uitbreiden.

⁴Minecraft Forge source installation, <http://www.minecraftforge.net/wiki/Installation/Source>

5 Taakverdeling

5.1 Organisatie

- Projectleider: Arjan
- Notulist: Lode

5.2 Technisch

- Gebruikersinterface van editor
 - Stijn
 - Lode
 - Jorke
- Koppeling tussen editor en mod
 - Arjan
 - Mike
- Implementatie van generieke mod
 - Jorke
 - Mike
 - Arjan
- Website
 - Lode
 - Stijn

6 Planning

Week	Activiteiten	Deadlines
49	projectdocument afmaken begin aan website maken editor ontwerpen mod implementeren	projectdocument (vrijdag)
50	tussenproduct tonen website afmaken tussenpresentatie voorbereiden	tussenproduct (vrijdag)
51	mod en editor koppelen	website & tussenpresentatie (woensdag)
52	functionaliteit uitbreiden	
1	testen handleiding maken	
2	eindverslag maken website bijwerken flyer/tutorial maken	
3	product afronden presentatie voorbereiden demonstratie voorbereiden	eindverslag, werkend systeem, website, flyer (vrijdag)
4	symposium	presentatie & demonstratie

Tabel 1: De weken lopen tot woensdag, omdat op die dag de wekelijkse vergadering plaatsvindt.