Plan van aanpak

PLUG

THAT

PLUMBER

Versie 1.3

**Groep 1**

Maarten van den Hoek – 0837626

Felix van Leeuwen – 0839339

Tom Nieuwenhuys – 0840825

Andra Veraart – 0835868

Rotterdam, 19 januari 2011

# Inhoudsopgave

[Projectinhoud 3](#_Toc278985797)

[Achtergrond 3](#_Toc278985798)

[Probleemstelling 3](#_Toc278985799)

[Doelstelling 3](#_Toc278985800)

[Resultaat 3](#_Toc278985801)

[Afbakening 3](#_Toc278985802)

[Risico’s 4](#_Toc278985803)

[Het werk per fase 5](#_Toc278985804)

[De Initiatieffase 5](#_Toc278985805)

[Definitiefase 5](#_Toc278985806)

[Ontwerpfase 5](#_Toc278985807)

[Voorbereidingsfase 5](#_Toc278985808)

[Realisatiefase 6](#_Toc278985809)

[Nazorgfase 6](#_Toc278985810)

[De beheersplannen 7](#_Toc278985811)

[Tijdsplan 7](#_Toc278985812)

[Initiatieffase 7](#_Toc278985813)

[Definitiefase 7](#_Toc278985814)

[Ontwerpfase 7](#_Toc278985815)

[Voorbereidingsfase 8](#_Toc278985816)

[Realisatiefase 8](#_Toc278985817)

[Nazorgfase 8](#_Toc278985818)

[Totaal aantal uren 8](#_Toc278985819)

[Kwaliteitsplan 9](#_Toc278985820)

[Informatieplan 10](#_Toc278985821)

[Organisatieplan 11](#_Toc278985822)

[Voortgangsbewaking 12](#_Toc278985823)

# Projectinhoud

In dit hoofdstuk wordt het *waarom*, *hoe*, *wat wel* en *wat niet* van dit project beschreven.

## Achtergrond

In het kader van INFPRJ02 zullen de leerlingen van INF1 in groepjes van drie tot vijf leerlingen binnen acht weken een game ontwikkelen.

## Probleemstelling

Mensen vervelen zich, en willen graag via internet, in hun eentje, een spelletje spelen. Eén van de meest populaire spelvormen van dit moment is het Tower Defence genre. Hoe kan een Mazing Tower Defence spel ontwikkeld worden?

## Doelstelling

Het doel is om een zogenaamd *mazing tower defence* spel te ontwikkelen.

## Resultaat

Aan het eind van dit project levert de projectgroep een *tower defence* spel op dat in een browser gespeeld kan worden. De afmetingen van het spel zullen maximaal 700 bij 700 pixels zijn. Bij het spel wordt een HTML-pagina geleverd waarin het spel geplaatst wordt.

Het spel zal in Java ontwikkeld worden.

## Afbakening

De projectgroep is niet verantwoordelijk voor de server waarop het spel uiteindelijk speelbaar gaat zijn.

## Risico’s

### Performance problemen

Een risico is dat de eindgebruiker de game niet zonder performanceproblemen kan spelen, omdat de eindgebruiker een computer heeft die niet goed meer kan presteren. De game kan de computer dusdanig belasten, dat de gebruiker stotteringen te zien krijgt, of dat het spel helemaal niet te spelen is.

Om te bepalen of dit daadwerkelijk een probleem zal vormen zal het spel onder anderen getest worden op wat oudere computers, zoals de publieke computers van de Hogeschool.

Eén manier om de performance te meten is het controleren van de lengte van de zogenaamde heartbeat. Dit is een methode die steeds na een vast aantal milliseconden wordt uitgevoerd. Wanneer blijkt dat de tijd tussen iedere beat langer wordt dan de vooraf bepaalde tijd, dan is het duidelijk dat niet alle berekeningen op tijd uitgevoerd kunnen worden.

Om het spel te “wapenen” tegen performance problemen, kunnen we onderzoeken of het mogelijk is om bepaalde functies minder vaak uit te voeren wanneer blijkt dat de beats te lang duren. Op die manier zou bijvoorbeeld de vloeiendheid van animaties opgeofferd kunnen worden voor een betrouwbaar speeltempo.

### Problemen door oudere versie van Java

Een ander risico is dat het gebruik van verschillende Java-versies door verschillende eindgebruikers problemen kan veroorzaken doordat bepaalde functionaliteiten niet uitgevoerd kunnen worden.

Een oplossing hiervoor zou kunnen zijn dat er d.m.v. een check gekeken wordt welke Java-versie de client op zijn computer heeft. Als de versie niet dezelfde is als de laatste versie, moet er een suggestie gedaan worden om de nieuwe Java-versie te downloaden. Zolang dit niet gebeurd is, kan het spel ook niet gespeeld worden.

# Het werk per fase

In dit hoofdstuk bespreken we voor iedere fase het werk dat gebeurd is, of nog moet gebeuren.

## De Initiatieffase

Toen duidelijk werd dat er een game ontwikkeld moest worden, heeft de projectgroep verschillende speltypes en -genres overwogen. Uiteindelijk is gezamenlijk gekozen voor een spel van het type *mazing tower defence*.

Verder is er, na enkele proeven met verschillende platformen, besloten te gaan ontwikkelen in Java.

## Definitiefase

Tijdens de kick-off voor dit project is duidelijk geworden dat er een spel ontwikkeld moet worden dat door één persoon gespeeld kan worden. Het spel moet kunnen functioneren als een onderdeel van een HTML-pagina.

Verder is uit de modulewijzer voor het vak INFPRJ02 duidelijk geworden dat het spel niet meer dan 700 bij 700 pixels groot mag zijn.

Het spel moet *meaningful play* bieden, in de vorm van vooruitgang in score, levels en/of verhaallijn. Ook worden creativiteit en het gebruik van technische oplossingen (zoals complexe wiskunde en/of kunstmatige intelligentie) meegenomen in de beoordeling.

## Ontwerpfase

De ideeën en richtlijnen die uit de eerste twee fasen naar voren zijn gekomen, zullen omgezet moeten worden in een spelontwerp. Hierin komen onder anderen de volgende zaken aan de orde: de verhaallijn, de spelregels en ontwikkeling tijdens het spel, ontwerpschetsen en de technische architectuur van het spel.

## Voorbereidingsfase

Zodra er een spelontwerp ligt, kan er bepaald worden uit welke delen het werk zal bestaan dat tijdens de realisatiefase zal moeten worden uitgevoerd, om het project tot een goed einde te brengen.

Tijdens deze fase zullen ook ontbrekende kennis of materialen aangevuld worden. Op die manier wordt er voor gezorgd dat alle benodigdheden binnen de projectgroep aanwezig zijn, voordat met de daadwerkelijke ontwikkeling begonnen wordt.

## Realisatiefase

Na de eerder genoemde voorbereidingen kan begonnen worden aan het ontwikkelen van het spel. Allereerst zal een prototype van het spel gemaakt worden: een versie die vooral is gericht op de technische basisprincipes waarop het spel gebaseerd is.

Vervolgens zal er naar een bètaversie toe gewerkt worden. Deze versie moet alle geplande functionaliteiten bezitten, maar er kunnen op dat moment nog fouten in het spel zitten.

Vanaf dat moment wordt er gewerkt om zoveel mogelijk fouten op te sporen, en in volgorde van noodzaak proberen op te lossen.

## Nazorgfase

Na het opleveren van het spel zal het getest worden door alle medestudenten. Tijdens deze periode zal er een trailer gemaakt worden om het spel aan te prijzen. Ook zal er een presentatie voorbereid worden, waarbij het spel aan de medestudenten en de projectdocent zal worden gedemonstreerd.

# De beheersplannen

Dit project kent verschillende facetten die ieder door een eigen beheersplan beschreven wordt.

## Tijdsplan

Het project kent een bepaald tijdsplan. Hieronder wordt per fase aangegeven welke tussenproducten de fasen op moeten leveren en wanneer de fases afgerond dienen te zijn.

Begindatum project: 18 november 2010

Einddatum project: 28 januari 2010

### Initiatieffase

Begindatum: 18 november 2010

Einddatum: 26 november 2010

Producten binnen de initiatieffase met het benodigde aantal uren:

* samenwerkingscontract (3 uren);
* plan van aanpak (15 uur);
* (goedgekeurd) spelvoorstel (15 uren).

23 november wordt het spelvoorstel besproken met de docenten zodat er groen licht gegeven kan worden.

26 november dient het definitieve plan van aanpak ingeleverd te worden.

### Definitiefase

Begindatum: 18 november 2010

Einddatum: 26 november 2010

Het voornaamste van de definitiefase in dit project is het verzamelen van specifieke eisen voor dit project.

### Ontwerpfase

Begindatum: 23 november

Einddatum: 3 december

Producten binnen de ontwerpfase met het benodigde aantal uren:

* spelontwerp:
  + verhaallijn (5 uren);
  + spelregels (10 uren);
  + ontwikkeling tijdens het spel (15 uren);
  + ontwerpschetsen (20 uren);
  + technische architectuur van het spel (10 uren).

3 december dient het spelontwerp te worden opgeleverd.

### Voorbereidingsfase

Begindatum: 3 december 2010

Einddatum: 10 december 2010

Producten binnen de voorbereidingsfase met het benodigde aantal uren:

* document met daarin de ontleding van het spel in componenten met de specifieke eigenschappen van de componenten (40 uur).

10 december dient dit document klaar te zijn. Vanaf dan kan er begonnen worden met programmeren.

### Realisatiefase

*Deel a*

Begindatum: 10 december 2010

Einddatum: 7 januari 2011

Producten binnen de realisatiefase deel a met het benodigde aantal uren:

* beta versie van het spel (70 uren);
* demonstratie (10 uren).

7 januari dient de beta versie van het spel te worden opgeleverd door middel van een demonstratie.

*Deel b*

Begindatum: 7 januari 2011

Einddatum: 21 januari 2011

Producten binnen de realisatiefase deel b met het benodigde aantal uren:

* final versie van het spel (40 uren);
* trailer (15 uren);
* projectdossier (5 uren).

21 januari dienen alle producten te worden opgeleverd.

### Nazorgfase

Begindatum: 21 januari

Einddatum: 28 januari

Producten binnen de nazorgfase met het benodigde aantal uren:

* eindpresentatie (20 uren);
* trailer (20 uren).

28 januari wordt de eindpresentatie gehouden.

### Totaal aantal uren

Totaal aantal geplande uren: 343 uren.

Er zullen naar alle waarschijnlijkheid tot ongeveer 100 uur extra bijkomen voor vergaderingen en management.

Totaal aantal uren: 343 + 100 = 445 uren.

## Kwaliteitsplan

Het project bevat vanuit zichzelf al een aantal eisen waaraan moet worden voldaan. Om aan deze eisen te voldoen, moeten we de eisen direct in ons spelontwerp verwerken. Voor de oplevering wordt nog een test gedaan, waaruit zal moeten blijken of aan de projecteisen is voldaan.

Alle code moet compileren en mag geen *runtime* fouten veroorzaken. Alle functies die gerelateerd zijn aan de *heartbeat* van het spel moeten afgerond kunnen worden binnen het tijdsbestek van één *heartbeat*. Op die manier voorkomen we dat de speler merkbare vertraging in de *gameplay* ervaart.

Voorgaand aan het schrijven van algoritmes zal eerst onderzoek gedaan worden naar vergelijkbare algoritmes. Zodat er niet wordt nagedacht over problemen die al door andere informatici zijn opgelost. Dit bespaard tijd en komt ten goede van de efficiëntie van de door ons geschreven algoritmes. Vooral bij het kortstepadalgoritme is het van groot belang dat deze aan een van buitenaf gestelde standaard voldoet (bijvoorbeeld Dijkstra’s algoritme).

We zullen een testperiode in moeten calculeren om te testen of aan alle eisen is voldaan. Deze periode zal na de oplevering van de bètaversie plaats vinden. Tijdens deze periode zal ook het zoeken naar fouten in de code, die veroorzaken dat het spel niet goed kan functioneren in bepaalde spelsituaties, echt van start gaan.

## Informatieplan

Er wordt gecodeerd in de programmeertaal Java. Er wordt gezorgd voor een duidelijke opmaak van de code. Daarnaast wordt geprobeerd om herhaling van code te voorkomen. Elk object dat wordt aangemaakt, krijgt zijn eigen klasse. In deze klassen heeft elk attribuut een get en een set methode, bovendien moet er bij elke methode een korte uitleg komen over de desbetreffende methode. Hiervoor wordt de JavaDoc notatie gebruikt.

Voor het opslaan van bestanden, wordt de extensie .docx gebruikt.

Er wordt gebruik gemaakt van SVN om data op een centrale server op te slaan, zo kunnen alle bestanden eenvoudig worden gedeeld. Bovendien kan er gemakkelijk een oudere versie opgehaald worden, in geval dat het programma niet langer blijkt te werken. Er zijn echter wel een aantal regels voor het gebruik van SVN.

* Vóór het committen van een document of van code, wordt er eerst een update gedaan zodat er niks overschreven wordt;
* Code wordt pas gecommit als duidelijk is dat de nieuwe code werkt;
* Bij het committen wordt er een eenduidige omschrijving gegeven van wat er veranderd is. Op deze manier kan er gemakkelijk nagegaan worden wat er allemaal gebeurd is en kunnen fouten makkelijker gevonden worden.

## Organisatieplan

De projectgroep bestaat uit de volgende leden:

* Maarten van den Hoek
* Felix van Leeuwen
* Tom Nieuwenhuijs
* Andra Veraart – projectleider

Binnen de groep heeft iedereen zijn eigen taken en daardoor zijn eigen stuk code te schrijven. In de meeste gevallen zal dit in aparte gevallen kunnen, maar soms zal een bestaande klasse uitgebreid moeten worden.

Momenteel zijn de onderdelen verdeeld in:

* Het pathfinding algoritme, dit wordt geschreven door Felix.
* Het kunnen plaatsen en verkopen van torens, dit wordt gedaan door Maarten en Tom.
* Het schieten van de torens, vooralsnog zonder collision detection, dit wordt gedaan door Andra.

Onderling contact verloopt via de geplande vergaderingen en contacturen. Daarbuiten zal normaal gesproken email gebruikt worden. Bij zaken die zo snel mogelijk geregeld moeten worden (minder dan 24 uur) heeft telefonisch contact of sms de voorkeur.

## Voortgangsbewaking

Bij dit project gaan we gebruik maken van voortgangsreportage`s om te controleren of de originele planning gehaald wordt. De planner gaat dit bij houden. Elke week wordt er bij de opdrachtgever een voortgangsrapportage ingeleverd en ook bij de vergaderingen wordt voortgang gecontroleerd. Dat betekent dus dat minimaal 3 keer in de week een controle plaatsvindt. De projectleider zal bij vertraging ingrijpen en eventueel bijsturen als het niet goed loopt.