



# Examenportfolio

**Roc Ter AA**

De vloer is lava

**Embedded fitness**

**Deelnemer:** Sam Hendriks  
Beukenstraat 229  
5753 GS Deurne  
0627333668  
77199@roc-teraa.nl

**School:** ICT College – ROC Ter AA  
Keizerin Marialaan 2  
5702 NR Helmond  
Tel: 0492 – 507900  
Postbus 490  
5700 AK Helmond

**BPV-docent:** Thomas Sprong  
t.sprong@roc-teraa.nl

**Leerbedrijf:** Embedded fitness  
Berkveld 1  
5709 AE Helmond  
06 53708878

**Praktijkopleider:** Joël van Huijkelom  
joel@embeddedfitness.nl

**Opleiding:** Applicatie ontwikkelaar, 25187, 2017

**Periode:** Dit project is uitgevoerd in de periode van  
4-5-2020 tot 29-5-2020

## Contents

Examenportfolio.....	1
Inleiding .....	7
PvB opdracht.....	7
Examenafspraken .....	8
B1-K1-W1: Stelt de opdracht vast .....	8
B1-K1-W2: Levert een bijdrage aan het projectplan .....	9
B1-K1-W3: Levert een bijdrage aan het ontwerp.....	9
B1-K1-W4: Bereidt de realisatie voor .....	10
B1-K2-W1: Realiseert (onderdelen van) een product .....	11
B1-K2-W2: Test het ontwikkelde product.....	11
B1-K3-W1: Optimaliseert het product.....	12
B1-K3-W2: Levert het product op .....	12
B1-K3-W3: Evalueert het opgeleverde product .....	12
P1-K1-W1: Onderhoudt een applicatie.....	13
P1-K1-W2: Beheert gegevens.....	13
Oriëntatie.....	14
Brainstormen.....	14
Concept.....	14
Documenten.....	14
Onderzoek.....	14
Programma van eisen.....	15
Projectplan.....	15
Functioneel ontwerp .....	15
Technisch ontwerp .....	16
Vorbereiding realisatie .....	16
Benodigdhedenlijst.....	16
Configuratie ontwikkelomgeving .....	16
Testen ontwikkelomgeving .....	16
Realisatie.....	17
Testen .....	17
Acceptatietest .....	17
Opleveren .....	17

Evaluatie .....	17
Onderhoud.....	17
Development.....	19
Tools required .....	19
Git Flow.....	19
Branches .....	19
Branch Naming .....	19
Pull requests.....	20
How do Pull requests work and what are they good for? .....	20
Guidelines .....	20
Creating a game .....	21
Adding your game .....	21
Setting your game data.....	22
GetGameControls().....	22
GetGameDescription() .....	22
GetGameIcon() .....	22
GetGameBehavior() .....	22
Creating custom ui .....	22
UI & Template .....	22
Icons & Button Icons.....	22
Logging.....	23
WriteToConsole.....	23
WriteToUiConsole .....	23
Inleiding .....	25
Brainstorm .....	25
Concept.....	25
Analyse .....	25
Verzamelde informatie .....	26
Pitch .....	26
Documenten.....	26
Afspraken .....	27
Eindresultaat .....	27
Impact.....	28
Embedded fitness.....	28
Het team .....	28

Inleiding .....	30
Project assignment .....	30
Project activiteiten .....	30
Project grenzen .....	31
Moscow .....	31
Producten .....	32
Project organisatie .....	32
Planning .....	33
Wat word er gebruikt .....	33
De planning .....	33
Documentatie .....	33
Realisatie .....	34
Opleveren.....	34
Risico's .....	34
Inleiding .....	36
Project management .....	36
Project details .....	36
Project omschrijving .....	36
Requirements.....	37
Use case diagram .....	37
Wireframe.....	38
Flowchart .....	39
Technische Eisen .....	41
Hoe gaat het werken .....	41
Voorbeeld.....	41
Componenten en modules .....	42
Diagrammen.....	43
Klassen diagram.....	43
Activiteiten diagram .....	44
Relationeel datamodel .....	45
Beslissingen.....	45
Afspraken.....	46
Grenzen .....	46
Plan van Aanpak.....	46
Haalbaarheidsstappen .....	47

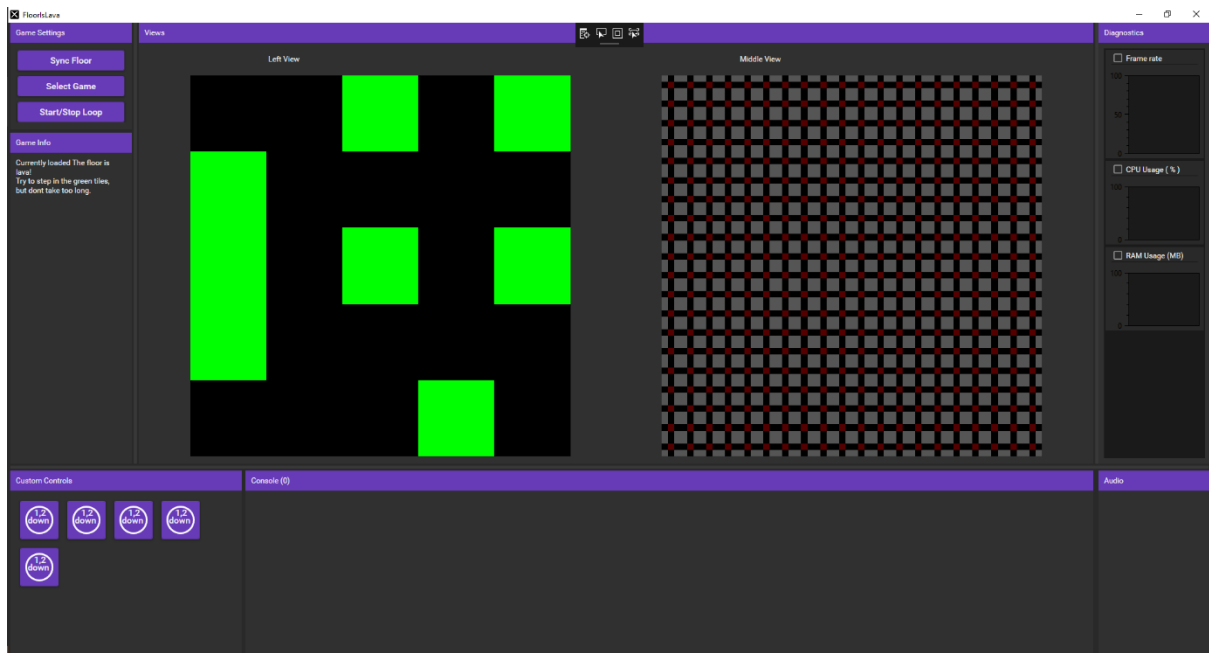
Test plan .....	47
Benodigheden .....	49
Werkomgeving .....	49
Inleiding .....	58
Opleveren .....	58
Product .....	58
Process .....	58
Inleiding .....	60
Het product .....	60
Vorbereiding .....	60
Documentatie .....	60
Realisatie .....	61
Oplevering .....	61
Conclusie .....	61
Inleiding .....	64
Digitale opslagplaats .....	64
Inlog gegevens .....	65
Versiebeheer .....	65
Source code .....	66
Aanpassingen/Wijzigingen .....	74
Specificaties .....	74
Kennis .....	74
Gebruikte hardware .....	74
Gebruikte software .....	74
Bijlagen .....	84
CRPR lijst .....	84
WBS .....	84
Trello .....	84
Dankwoord .....	85

## Inleiding

Mijn naam is Sam Hendriks en ik doe de opleiding Applicatie- en mediaOntwikkeling. Ik ga mijzelf vooral focussen op C# en embedded systems en hier mee doorleren. Het bedrijf waar ik stage bij ga lopen is het Embedded fitness in helmond. Hier maken zij meerderen interactieve games die focussen of beweging en leren. Daarnaast maken ze ook programma's en simulaties voor verschillende soorten sporten. Hier ga ik aan meerdere opdrachten werken voor het bedrijf zelf en ik ga aan mijn afstudeer opdrachten werken van school.

## PvB opdracht

De PvB opdracht die ik heb gekozen is "De vloer is lava", Dit is een spel op een interactieve vloer. Het doel van het spel is om zolang mogelijk op de groene tegels te blijven staan. Het spel zelf begint doormiddel van een start signaal te geven. Dit word aangegeven door 3 verschillende kleuren te laten knipperen. Als eerst word de kleur rood vertoond gevolgd door oranje en als laatst groen. Zodra het start signaal gegeven is begint het spel, tijdens het spel worden er willekeurige tegels geselecteerd die groen worden. Hierop kan de speler staan, na een aantal seconden worden de overige tegels rood. Indien de speler op een van deze tegels staat of niet aanwezig is op de vloer word er een eind signaal gegeven. Dit word aangetoond door de vloer meerderen keren rood te laten knipperen. Als dit voorbij is word er een tegel geel en dit dient als de reset tegel, zodra de speler hierop staat word het spel gereset. "De vloer is lava" is deel van een aansturings programma("Lightspace"). Dit programma stuurt de vloer aan zoals het aansturen van de ledjes en het uitlezen van de sensoren. "De vloer is lava" geeft door aan "lightspace" welke ledjes er veranderd moeten worden, daarnaast kan het spel ook vragen of er sensoren geactiveerd in bepaalde tegels zijn.



Figuur 1 De vloer is lava demo

## Examenafspraken

### ***B1-K1-W1: Stelt de opdracht vast***

#### **De kandidaat interpreteert en/of oriënteert zich op de opdracht:**

Om erachter te komen wat ik wil gaan doen ga ik een brainstorm sessie houden. Hierbij ga ik een concept schrijven in het programma van eisen van wat ik wil gaan doen. Daarbij komt een tabel van losse ideeën die eventueel in het project kunnen. Het concept en de tabel ga ik bespreken met de opdrachtgever via een pitch en dan kan hij eventuele toevoegingen geven.

Om te kijken hoe ik het project wil uitvoeren ga ik kijken naar het concept. Door naar het concept te kijken kan ik kijken welke manieren ik het project het beste kan maken. Hierbij komt kijken in welke taal ik het wil programmeren, welke planning tools ik wil gebruiken, en in welke omgeving het project word gerealiseerd. Dit word allemaal vast gelegd in het programma van eisen.

Om te kijken met welke onderdelen ik dit wil realiseren ga ik de al bestaande onderdelen bekijken. Zoals het bekijken van de interactieve vloer en meer informatie hierover vragen krijg ik een beter zicht met wat ik ga werken. Hierdoor kan ik ook kijken of er extra onderdelen nodig zijn en wat er nodig is om het project te realiseren.

#### **De kandidaat verzamelt benodigde informatie via communicatie met een betrokkene:**

Om informatie te krijgen over het project ga ik een interview houden met de klant. In het interview ga ik vragen met wat ik moet gaan werken, wat het verwachten eindresultaat is, welke eisen er in het project moeten komen, en de afspraken die tussen ons gemaakt moeten worden.

#### **De kandidaat verzamelt benodigde (aanvullende) informatie door onderzoek (bijv. eventuele oude software, concurrerende software, documenten, stroomdiagram):**

Na het interview met de klant heb ik gevraagd voor eventuele oude software/documenten die beschikbaar waren, daarnaast heb ik een onderzoek uitgevoerd of er concurrerende software/documenten zijn.

#### **De kandidaat analyseert de verkregen informatie en legt dit vast in een programma van eisen, verduidelijkt met schema's:**

Alle verkregen informatie ga ik eerst bekijken, daarna bepaal ik wat er relevant is voor het project zelf en leg dit vast in het programma van eisen.

#### **De kandidaat brengt in het programma van eisen de (on)mogelijkheden van het gevraagde in kaart:**

Om de (on)mogelijken te krijgen geven wij eerst een pitch aan de klant van wat wij willen maken. Na deze pitch kan de klant input geven over wat zij hier graag bij willen hebben. deze (on)mogelijkheden leggen wij vast in het programma van eisen. Hierbij komt te staan of deze (on)mogelijkheden uit te voeren zijn of niet.

#### **De kandidaat brengt de impact voor de betrokkenen in kaart in het programma van eisen:**

In het programma van eisen komt op het einde te staan wat dit project voor impact heeft op de betrokkenen. Zoals wat zij kunnen doen met het project wanneer dit is afgerond.



**De kandidaat communiceert het programma van eisen met de opdrachtgever en vraagt om goedkeuring:**

Als het programma van eisen is afgerond ga ik naar de opdrachtgever toe en vraag voor goedkeuring. Dit word aangegeven met een handtekening op het document.

## ***B1-K1-W2: Levert een bijdrage aan het projectplan***

**De kandidaat beschrijft zijn eigen projectdoelstellingen in het projectplan:**

Via het programma van eisen word er een projectdoelstelling gemaakt in het projectplan. Hier ga ik kijken wat het uiteindelijke doel is van het project en hoe ik dit wil bereiken.

**De kandidaat beschrijft zijn uit te voeren projectactiviteiten (waaronder zijn inzet, middelen en ontwikkelmethode) in het projectplan:**

In het projectplan beschrijf ik hoe ik bij het uiteindelijke doel kom. Dit doe ik door alle activiteiten in het project op te schrijven.

**De kandidaat plaats zijn projectactiviteiten in chronologische volgorde gekoppeld aan tijd (planning) in het projectplan:**

In het projectplan heb ik beschreven wanneer ik welke activiteiten ga uitvoeren. Dit doe ik aan de hand van een WBS document. Hierin staat hoelang ik verwacht te werken aan een onderdeel en hoelang ik er aan heb gewerkt. Langs deze tabel staat een GANNT tabel waarin terug te vinden is in welke week een onderdeel word uitgevoerd.

**De kandidaat geeft in het projectplan in zijn planning aan wanneer hij wat terug koppelt en aan wie:**

In het project plan staat een bulletlist met de namen van de programmeurs die deelnemen aan het project. Onder elke naam staat wat zijn verantwoordelijkheden zijn en de functie van de programmeur.

**De kandidaat stemt zijn bijdrage aan het projectplan af met de projectleider en vraagt om goedkeuring:**

Zodra het projectplan is afgerond word deze gepresenteert aan de opdrachtgever. Daarna word er om goedkeuring gevraagd van het document dit word aangegeven met een handtekening.

## ***B1-K1-W3: Levert een bijdrage aan het ontwerp***

**De kandidaat stelt requirements op, en legt ze vast in het functioneel ontwerp:**

Met behulp van het programma van eisen worden alle requirements benoemt in het functioneel ontwerp.

**De kandidaat bepaalt de prioriteit van de requirements:**

Door het bekijken van de requirements ga ik de prioriteit plaatsen van iedere requirement die in het project aanwezig is.

**De kandidaat past in het functioneel ontwerp schematechnieken toe ter verduidelijking van de requirements:**

Om een duidelijk beeld te krijgen van de prioriteiten van de requirements worden deze geplaatst in een MoSCoW lijst.

**De kandidaat legt in het functioneel ontwerp de eisen van de user interface vast d.m.v. schetsen:**

Om de eisen van de user interface vast te leggen word er een schema gemaakt waarin te zien is wat de gebruiker allemaal kan doen.

**De kandidaat legt het functioneel ontwerp met een korte toelichting voor aan de opdrachtgever ter goedkeuring:**

Het functioneel ontwerp word opgeleverd aan de opdrachtgever en er word gevraagd om goedkeuring van het document. Dit zal worden aangegeven met een handtekening.

**De kandidaat vertaalt het functioneel ontwerp naar technische specificaties en legt ze vast in het technisch ontwerp:**

Aan de hand van het functioneel ontwerp word er een technisch ontwerp geschreven. Het technisch ontwerp neemt functionelen specificaties en zet deze om naar technische specificaties.

**De kandidaat maakt in het technisch ontwerp een relationeel datamodel:**

In het technisch ontwerp word er uitgelegt hoe de data in het project verstuurd, ontvangen en opgeslagen word. Om dit te verduidelijken word er een dataflowdiagram(DFD). schema toegevoegd, hierin word getoont hoe de onderdelen in dit project met elkaar communiceren.

**De kandidaat past in het technisch ontwerp schematechnieken toe ter verduidelijking van de technische specificaties en/of werking van de applicatie:**

In het technisch ontwerp staan een aantal diagrammen ter verduidelijking van de technische specificaties. Hieronder vallen een activiteiten diagram, klassendiagram en een DFD.

**De kandidaat legt het technisch ontwerp met een korte toelichting voor aan de opdrachtgever ter goedkeuring:**

Het technisch ontwerp word opgeleverd aan de opdrachtgever en er word gevraagd om goedkeuring van het document. Dit zal worden aangegeven met een handtekening.

### ***B1-K1-W4: Bereidt de realisatie voor***

**De kandidaat inventariseert de benodigde onderdelen voor de realisatie:**

Voordat ik aan het realiseren van het project begin ga ik eerst een document opmaken waarin alle benodigdheden voor het project instaan. In de benodigdheden lijst staat welke onderdelen ik nodig zal hebben om dit project succesvol te laten slagen.

**De kandidaat installeert en configureert de ontwikkelomgeving:**

Na het inventariseren van de benodigdheden word de ontwikkelomgeving geconfigureert. De veranderingen die worden toegepast en de versie waarin gewerkt word worden gedocumenteert in het kwaliteitshandboek.

**De kandidaat test de ontwikkelomgeving:**

Zodra de ontwikkelomgeving geconfigureert is word deze getest door een test programma aan te maken. Hierin worden functies getest die nodig zijn voor het project zelf.

**De kandidaat documenteert de instellingen en wijzigingen:**

De instellingen en wijzigen die zijn gemaakt voordat het realiseren begint worden gedocumenteert in het kwaliteitshandboek.

***B1-K2-W1: Realiseert (onderdelen van) een product***

**De kandidaat inventariseert de werkzaamheden en plaatst deze in chronologische volgorde in een planning:**

Voordat het programmeren begint word er een trello pagina aangemaakt. In de trello pagina worden alle werkzaamheden geplaatst en gelabeld. Daarnaast word er bij elke werkzaamheid een nummer gezet en word deze zo op volgorde geplaatst.

**De kandidaat realiseert een applicatie volgens de goedgekeurde ontwerpdocumentatie:**

Er word een applicatie gemaakt die voorzien is van elke eis die voorheen is benoemd.

**De kandidaat voorziet de broncode van functioneel commentaar:**

Bij elke unieke functie en block broncode word er een stuk commentaar geplaatst die uitlegd wat het uitvoerd.

**De kandidaat administreert zijn werkzaamheden:**

Alle onderdelen die afgerond zijn worden in de trello verplaats naar een andere tabel, hieraan kan worden gezien wat er af is en niet. In trello kan ik zien wat er nog gedaan moet worden, waar aan gewerkt word, en wat er af is. Daarnaast word er in een logboek bijgehouden wat er is uitgevoerd wanneer.

***B1-K2-W2: Test het ontwikkelde product***

**De kandidaat legt zijn wijze van testen vast in een testplan:**

Het testplan word opgesteld in het technisch ontwerp, hierin zet ik hoe en wat ik wil gaan testen.

**De kandidaat stelt testcases op:**

In de testcases komt te staan wat ik ga testen, wat het verwachten eindresultaat is, wat het resultaat is, en nog een eventuele opmerking.

**De kandidaat voert de testcases uit:**

Na het maken van de applicatie worden de testcases uitgevoerd. Hierin zet ik wat het resultaat is van het testen. Bij het testen voer ik elke keer de hele lijst uit, hieraan kan ik zien of er een probleem is ontstaan bij een onderdeel als er bij een ander onderdeel iets is opgelost.

**De kandidaat legt de resultaten van de testcases vast:**

De resultaten van de testcases worden langs het verwachten testresultaat geplaatst.

**De kandidaat bepaalt of op basis van de resultaten aanpassingen nodig zijn en legt zijn bevindingen vast:**

Aan de hand van de resultaten uit de testen zet ik bij de opmerkingen neer of er een aanpassing nodig is. Daarnaast worden deze aanpassingen vermeld in de Change request problem report(CRPR) lijst en de versiecontrol.

**De kandidaat past indien nodig n.a.v. de testresultaten de applicatie en documentatie aan:**

Indien er veranderingen komen in het gehele project vanwege de testresultaten worden de documenten aangepast. Deze aanpassingen worden vastgelegd in de crpr lijst.

### ***B1-K3-W1: Optimaliseert het product***

#### **De kandidaat stelt een acceptatietest op voor zijn applicatie:**

Na het testen van de applicatie ge ik een acceptatietest opzetten. Hierin word vermeld wat mijn applicatie hoort te doen. Bij ieder onderdeel komen 3 vakken te staan, 2 hiervan kunnen worden aangekruist. Deze vakken zijn OK en NOK, deze vakken geven aan of het onderdeel goed is(OK) of dat het niet voldoende is(NOK). In het 3<sup>de</sup> vak kunnen opmerkingen gezet worden, deze opmerkingen kunnen verbeterpunten zijn of wijzigingen voor dat onderdeel.

#### **De kandidaat laat de acceptatietest uitvoeren door een of meer gebruikers en begeleidt de gebruikers tijdens het uitvoeren hiervan:**

Na het opzetten van de acceptatietest laat ik meerdere personen de acceptatie test uitvoeren op mijn applicatie.

#### **De kandidaat bepaalt welke informatie van gebruikers hij kan verwerken in de applicatie, beschrijft voor deze aanpassingen zijn werkzaamheden en voert deze aanpassingen door:**

Na het bekijken van iedere acceptatietest van de gebruikers ga ik beslissen welke opmerkingen gebruikt kunnen worden in de applicatie. Deze aanpassingen worden daarna uitgevoerd en vastgelegd in de CRPR lijst.

#### **De kandidaat werkt de documentatie van de applicatie bij:**

Indien het project veranderd is of dat de documentatie verouderd is word deze bijgewerkt. Deze aanpassingen worden vermeld in de CRPR lijst.

### ***B1-K3-W2: Levert het product op***

#### **De kandidaat demonstreert het product aan de hand van het programma van eisen, en vraagt om goedkeuring van de opdrachtgever:**

Het product word opgeleverd aan de opdrachtgever, daarna word er gevraagd om goedkeuring van het product.

### ***B1-K3-W3: Evalueert het opgeleverde product***

#### **De kandidaat evalueert het product met de betrokkenen (wat is goed gegaan, verbeterpunten) en legt dit vast in een evaluatieverslag:**

Na het opleveren van het product ga ik samen met de opdrachtgever het product evalueren. De evaluatie word vastgelegd in het evaluatieverslag.

#### **De kandidaat evalueert het proces met de betrokkenen (opleverproces, wat is goed gegaan, verbeterpunten) en legt dit vast in een evaluatieverslag:**

Na het evalueren van het product ga ik samen met de opdrachtgever het process evalueren. De evaluatie word vastgelegd in het evaluatieverslag.

#### **De kandidaat laat zijn evaluatieverslag accorderen:**

Als het evaluatieverslag is ingevuld word deze ondertekent door de opdrachtgever.

***P1-K1-W1: Onderhoudt een applicatie***

**De kandidaat analyseert en categoriseert incidentmeldingen (4x), en vraagt hierbij indien nodig om meer informatie:**

Indien er incidentmeldingen zijn worden deze bekeken en gecategoriseerd in de CRPR lijst. Indien nodig zal er om extra informatie gevraagd worden rond om het incident.

**De kandidaat legt de mogelijke oplossingen voor de incidenten vast en toetst of de oplossingen uitgevoerd kunnen worden (rekening houdend met eerder gemaakte afspraken of contracten, de werking van het product en beschikbare middelen):**

Bij de incidenten die indien aanwezig zijn word er gekeken naar goede oplossingen die niet in contrast komen met de afspraken die zijn gemaakt. Indien die niet aanwezig, maar er zijn nog andere oplossingen beschikbaar. Dan word er eerst gevraagd voor toestemming bij de opdrachtgever of dit een mogelijke oplossing kan zijn.

**De kandidaat voert de aanpassingen door en documenteert deze:**

De aanpassing word zodra goedgekeurd doorgevoert en word daarna gedocumenteert in de CRPR lijst en de versiecontrol.

**De kandidaat houdt de betrokkenen op de hoogte van de voortgang:**

De voortgang word wekenlijks besproken, indien dit niet een mogelijkheid is word dit via een mailtje aangegeven hoe het project ervoor staat.

***P1-K1-W2: Beheert gegevens***

**De kandidaat controleert gegevens van de applicatie (functionele-, technische- en testdocumentatie en bronbestanden) op juistheid, volledigheid en bruikbaarheid:**

zodra het project is opgeleverd en is gevalueert worden de documenten gecontroleert op juistheid, volledigheid, en bruikbaarheid.

**De kandidaat documenteert en archiveert gegevens van de applicatie (in een kwaliteitshandboek of digitaal archief). Hierbij noteert hij de specificaties, inloggegevens, databasegegevens, aanpassingen, eisen en wijzigingen en voert versiebeheer door:**

Alle specificaties, inloggegevens, databasegegevens, aanpassingen, eisen en wijzigingen komen in een kwaliteitshandboek te staan. Daarnaast komt hier ook een versiebeheer in te staan samen met een versiecontrol.

## Oriëntatie

### Brainstormen

Tijdens mijn brainstorm sessie heb ik een aantal ideeën gekregen voor spellen die van toepassing waren op de interactieve vloer. Hieronder vallen “De vloer is lava”, “Simon says”, “Stoelendans”, en “Plaat rennen”. Uit deze spellen heb ik “De vloer is lava” gekozen. Voor mij leek dit een uitdagend project om veel van te leren. De brainstorm sessie zelf was uitgevoerd met 4 personen in totaal. Iedereen had hiervoor ideeën opgeschreven en daarna hebben we individueel gekeken welke projecten ons het leuk zouden lijken (figuur 2).

### Concept

Aan de hand van de brainstorm sessie heb ik een concept gemaakt van het spel “De vloer is lava”. Het idee van het spel is om op de groene vakjes te blijven staan. Deze vakjes wisselen om de zoveel seconden. De overige vakken worden een aantal seconden van tevoren rood. Indien de speler op een van de overige vakken staat terwijl deze rood zijn is het spel afgelopen (figuur 3).

### Documenten

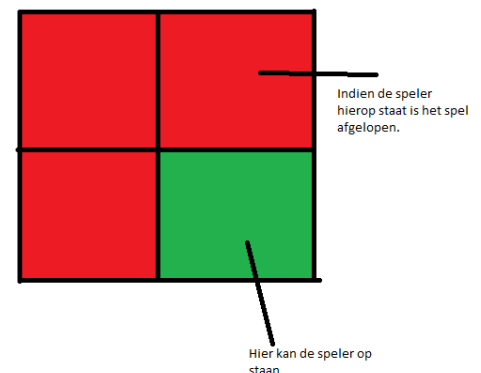
Om informatie op te doen over het project heb ik gevraagd voor eventuele documenten. Omdat dit een nieuw project was en Embedded fitness nog niet eerder met de interactieve vloer heeft gewerkt, was er momenteel nog geen documentatie of andere software waar ik naar kon kijken. Later is hier verandering in gekomen en is er een document gekomen voor het aansturings programma. In dit document stond beschreven hoe je een spel in het aansturings programma kon maken.

### Onderzoek

Via onderzoek ben ik vooral op nieuws artikelen gekomen over wat er mogelijk is met een interactieve vloer. Daarnaast waren er nog velen aantal webpagina's met reclame over een interactieve vloer en wat de mogelijkheden hiervan waren. Ook werden de diensten die het bedrijf kon geven beschreven. De manier van het aansturen en eventuele applicaties van de interactieve vloer waren niet aanwezig.



Figuur 2 Gekozen ideeën



Figuur 3 Concept

## Programma van eisen

Het programma van eisen staat mijn brainstorm sessie, het concept, de documentatie, het onderzoek, de (on)mogelijkheden, afspraken, en de impact van het project beschreven. Hierin staat hoe ik op mijn idee ben gekomen en hoe ik aan de bijbehorende informatie ben gekomen.

Ik heb het project in het kort uitgelegd als concept, met dit concept heb ik een pitch gehouden voor de opdrachtgever. Na de pitch heb ik voor feedback gevraagd, goedkeuring van het project, en daarna eventuele informatie zoals documenten en oude software. Deze pitch staat in dit document beschreven en de afspraken die hieruit zijn gekomen.

Naast dit heb ik gekeken naar de impact op de betrokkenen zoals de opdrachtgever, eventuele klanten van de opdrachtgever, en mijzelf als programmeur.

## Projectplan

In het projectplan heb ik de het project doel beschreven en hoe ik hier ga komen door alle projectactiviteiten op te noemen. Verder heb ik de project grenzen opgenoemd waarbij ik ga kijken wat ik wel en niet kan doen, zoals aan een onderdeel werken van de hardware. Daarnaast staat er ook in vermeld wat de uiteindelijke producten zijn als het project volledig is afgerond. Na de product vermelding heb ik een Alinea waarin staat uitgelegd wat ieder teamlid zijn/haar taken en verantwoordelijkheden zijn.

Een belangrijk deel van het projectplan is de planning, waarbij ik heb uitgelegd hoe ik de planning heb gemaakt en wat ik ga gebruiken om de voortgang bij te houden. Ik ga mijn planning maken aan de hand van een WBS. In de WBS laat ik zien wanneer ik een projectactiviteit uitvoer en hoelang ik denk dat ik erover ga doen. Daarnaast om bij te houden waarmee ik bezig ben en wat ik nog te doen heb ga ik trello gebruiken.

Als laatste onderdeel van het projectplan heb ik de risico's beschreven. Wat zijn de showstoppers van mijn project. Wat ga ik als eerste testen om ervoor te zorgen dat er niet teveel tijd is verloren aan het uiteindelijk oplossen indien het fout gaat.

## Functioneel ontwerp

In het functioneel ontwerp heb ik beschreven wat de functies zijn van mijn project. Hierin staat ook beschreven wat de functie is van ieder teamlid en waar zij verantwoordelijk voor zijn. Er word ook beschreven wat de project details zijn hier word er gekeken naar de projectnaam, het bedrijf, en de aannemer.

Na de projectdetails word er gekeken naar de functionele details van het project in een projectsomschrijving. Hierin komt te staan wat het spel gaat doen en de functies die het spel hiervoor gebruikt.

Hieronder word er gekeken naar de requirements van het project, deze requirements worden dan in een MoSCoW lijst gezet. Hierdoor worden de prioriteiten van elk onderdeel duidelijk vertoont.

Als laatst in dit document worden er een aantal diagrammen en wireframes vertoont over het project zelf. Een aantal diagrammen zijn de usecasediagram en de dataflowdiagram(DFD).

## Technisch ontwerp

Het technisch ontwerp is een technische vertaling van het functioneel ontwerp met een aantal extra onderdelen. Hier word er gekeken naar de technische eisen van het spel zoals hoe gaat het spel werken.

Daarnaast ga ik ook kijken welke componenten en modules er nodig zijn en hoe deze samenwerken in het spel zelf.

Na het vastleggen van de componenten en modules worden er een aantal diagrammen bekeken zoals de klassendiagram en de activiteitendiagram. In de klassendiagram komt te staan welke onderdelen met elkaar verbonden zijn en wat voor onderdelen ze bij zich hebben. bij de activiteitendiagram word er gekeken welke acties worden uitgevoerd na bepaalde handelingen of events.

Een groot onderdeel van het technisch ontwerp is het relationeel datamodel. Hierbij heb ik gekeken hoe de data verstuurd, ontvangen, en opgeslagen word. Dit word verduidelijkt met een DFD en een schema van hoe de data eruit ziet als deze word verzonden naar de interactieve vloer.

Als laats worden beslissingen over het project, afspraken tussen de teamleden, de grenzen, een stukje plan van aanpak, en het testplan beschreven.

## Vorbereiding realisatie

### ***Benodigdhedenlijst***

In de benodigdhedenlijst heb ik alle benodigdheden benoemd die ik nodig zal hebben om dit project te kunnen laten slagen.

### ***Configuratie ontwikkelomgeving***

Voor de configuratie van de ontwikkelomgeving heb ik gekeken naar het document dat mij was aangeleverd. In dit document voor het aansturingsprogramma stond beschreven in welke versie en welke taken er voldaan moesten worden om een spel te maken. Deze taken heb ik uitgevoerd om mijn ontwikkelomgeving op te zetten.

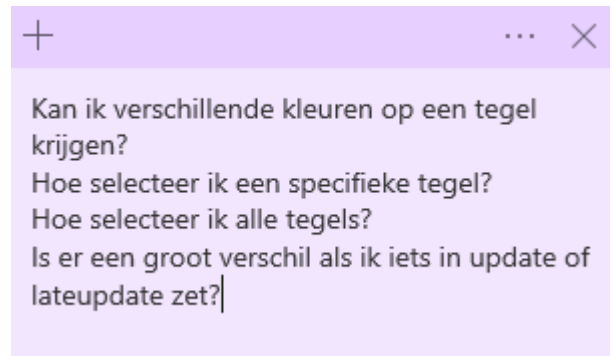
### ***Testen ontwikkelomgeving***

Voor het testen van de ontwikkelomgeving heb ik een testspel aangemaakt om een aantal functies uit te proberen.



## Realisatie

Bij het begin van de realisatie waren de zelfgemaakte functies die ik moest gebruiken om een spel te maken moeilijk om goed te kunnen gebruiken. Deze functies waren belangrijk voor het uitlezen van de sensoren en het aansturen van de ledjes op de interactieve vloer. Om dit op te lossen heb ik een aantal vragen opgeschreven en heb ik de programmeur van het aansturings programma deze vragen gesteld (figuur 4). Nadat deze vragen beantwoord waren ging het programmeren wat gemakkelijker. Het combineren van bepaalde functies ging nog steeds nog niet altijd perfect. Uiteindelijk is de applicatie afgerond en was deze klaar voor testen.



Figuur 4 Vragen aan programmeur

## Testen

Het testen van de applicatie is gedaan aan de hand van een testcase. Hierin stonden alle eisen waaraan de applicatie moest voldoen. Langs de eisen stond de verwachten uitkomst van de testen, daarnaast stond wat het echte resultaat was. Aan de hand van de testresultaten zijn er wat aanpassingen gemaakt en opnieuw getest.

## Acceptatietest

In het acceptatietest heb ik een aantal functies neer gezet welke de applicatie zou moeten behalen. Langs deze functies staan 2 vakjes met OK en NOK om aan te geven of deze functie goed (OK) of niet goed (NOK) is. Hierlangs staat nog een vak met opmerkingen over de functies. De acceptatietest is daarna aan testers gegeven die niet aan het project werken. Deze testers testen de applicatie aan de hand van de acceptatietest. Aan de hand van wat zij ervaren kunnen zij de vakken invullen met opmerkingen over de functies, zoals wat zij anders zouden willen of extra erbij. Aan de hand van deze resultaten ga ik kijken of deze aanpassingen gebruikt kunnen worden.

## Opleveren

Het opleveren van het project werd gedaan door met de opdrachtgever naar het volledig project te kijken. Hierbij heb ik de applicatie gepresenteert en heb ik uitgelegd hoe deze werkt. Daarna hebben wij gekeken naar het werkproces.

## Evaluatie

Bij de evaluatie hebben wij eerst gekeken naar het project zelf en hoe het gemaakt is, of deze aan de eisen voldoet, en wat kon er beter. De resultaten hiervan zijn vastgelegd in het evaluatieverslag. daarnaast hebben we naar het werkproces in het algemeen gekeken zoals hoe heb ik gewerkt, op welk tempo, en wat kon er beter de volgende keer.

## Onderhoud

Voor de onderhoud van de applicatie en documentatie heb ik een meerdere dingen bijgehouden. Hieronder vallen een CRPR lijst, een Versiecontrol, de trello, een gitkraken repository, en gekeken naar het verouderen van documenten. Op het einde van het project heb ik alle documenten bekeken of deze nog volledig en correct waren.

## Aansturing programma Embedded fitness

---

**Project:** De vloer is lava  
**Client:** Embedded fitness  
**Project Number:** 1.1

**Auteur:** Sam Hendriks  
**Date:** 25-5-2020  
**Version:** 1.1

---

---

The undersigned declare their agreement with the content of this Project Plan document

**Client**

***Initial Seen:***

***Initial Seen:***

**Project Manager**

Date: \_\_\_\_\_

Place: \_\_\_\_\_

Date:

Place:

## Development

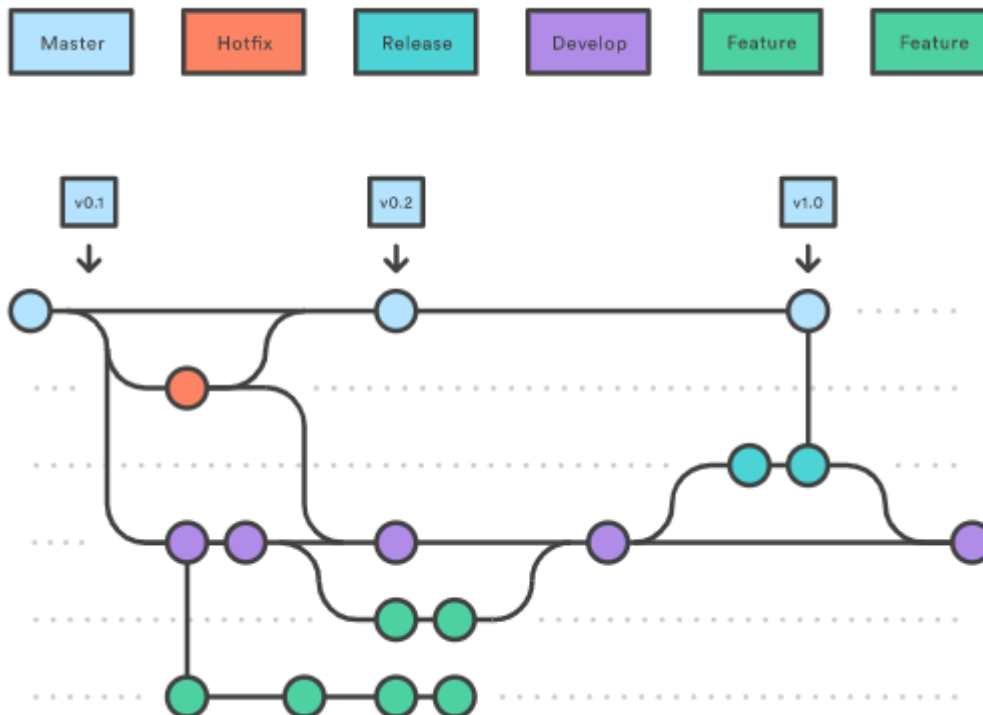
### Tools required

C# & XAML capable IDE (Visual studio / VSCode recommended)

git (GitHub desktop client / gitKraken recommended)

The target framework of the application is .NET Framework 4.8.

### Git Flow



### Branches

Lightspace development follows the git-flow workflow.

master Master branch. Pushes can be made using pull requests.

dev Develop branch. Pushes can be made using pull requests.

A full guide describing the git-flow workflow can be found [here](#).

### Branch Naming

Branches must follow the following naming strategy:

feature: feature/"project-name"-"short-implementation-description"

example: feature/SR0-added-hardware-support

release: release/"release\_version"

example: release/0.0.1

hotfix: hotfix/"project-name"-"short\_bug\_name"

example: hotfix/SR0-fix-nullreference

## Commit Messages

Commit messages should explain the largest impact change made in the new commit.

An example for how to do this is as follows. Imagine this hotfix branch fixed an issue where a nullreference occurred due to an error when loading a project resource.

hotfix/SR0-fix-nullreference

example: Fixed a nullreference occurring when loading resource.

If the commit/branch is relevant or an update to an Issue, include #[number] in the commit message to link the commit to the relevant issue.

example: Fixed a nullreference occurring when loading resource. #0

## Pull requests

Pull request are required to be reviewed by at least one other collaborator. After the check has passed it will be possible to merge the branch. A pull request should always be Squashed when merged.

### How do Pull requests work and what are they good for?

Pull requests are a step in between committing(& pushing) and merging. You push your branch with it's changes to the repository, and you create a pull request to the Develop branch (dev). The pull request will go into review, and you can invite other collaborators to review a pull request.

Pull requests should be reviewed by others as a quality gateway to check your code for possible errors or mistakes before it will be pushed onto the main development branch. This is to prevent (most of the) errors from ever entering the main development cycle and possibly interrupting the entire project workflow.

## Guidelines

When creating a project, there are certain guidelines to follow. These are important for maximizing performance, and minimizing issues.

- Only update your render visual if it has changed since your last tick.

This means that if your visual render has no changes, it does not need to be updated and pushed to the UI. You need to keep track of this yourself, because comparing the visuals every frame can get expensive very fast. Render visual found as gameRender in TileManager.cs and can be called using SetRenderGraphic() and GetRenderGraphic().

- Be careful when working with different threads.

The Core Loop, Main (UI) and Performance Profiling threads are all separate. This means you can't simply access all their functionality from your game class. Any needed functionality should be available to you, and otherwise you can make a new [Issue](#) with the desired feature.

- When removing or changing main program functionality, discuss with other collaborators.

If you are making changes to main program functionality, you may have solved all issues on your end, however other people might need or be working with said functionality. Discuss with other collaborators to work it out, and if needed work together in a [Pair Programming](#) manner.

## Creating a game

To create a new game, Create a new folder in the Games folder of the LightSpace\_WPF\_Engine Solution.

In this folder you can create a class. This class should inherit from RunningGameBehavior. Within this class certain standard functions can be used for creating your game:

- Start (public override void Start())

Will be triggered on creating the object.

- Update (public override void Update())

Will be triggered every frame of the game.

- LateUpdate (public override void LateUpdate())

Will be triggered every frame of the game after the entire update step has finished.

There are 2 more functions that can be used, though these are risky since they are able to break the flow of the Core Loop if used incorrectly.

- Destroy (Destroy())

Can be triggered to remove the object from the BehaviorManager, thus breaking possibly the last reference. (**dangerous**)

- OnDestroy (public override void OnDestroy())

Will be triggered before the object gets destroyed. (**dangerous**)

Information on setting game data, specifically your Behavior can be found [here](#).

## Adding your game

To add your game to the collection of existing games, we use the GameList class (Models/Models/GameList.cs) and the GameName enum (Models/Enums.GameName.cs).

First off, add your game to the GameName enum (if multiple new games are created at once, discuss with other developers in order to avoid creating duplicate values). After you've added your game to GameName, start filling in the Get functions within GameList.

## Setting your game data

### GetGameControls()

GetGameControls is the function that loads in your custom controls. (information on creating this found [here](#). Simple add the case for your GameName entry, and return a new object of the type of control you made. If you have no custom controls, you can return null.

### GetGameDescription()

GetGameDescription loads a string that is used within the UI to give a short description about what your game does, how it functions, or what it's used for.

### GetGameIcon()

GetGameIcon gets a bit more complicated. To use your image, you need to add it to the Solution Resources. First off, drag the image you wish to use into Media/GameIcons. The current sizing standard is 64\*64 Pixel .png file.

Once you've added your image, Right-Click the solution and go to the Resources section. Drag your newly imported image into the Resources. this should create a new entry with the name of your file.

Back in GameList you should now be able to return Properties.Resources.<ResourceName> directly.

### GetGameBehavior()

GetGameBehavior is the function that loads your game. If you've made your game according to the previous wiki entry, you should have a class that inherits from RunningGameBehavior. Simply return a new object of that class type in this method, and your game should be loaded.

## Creating custom ui

### UI & Template

Custom UI can be made for your game. The main UI already has a spot reserved for your custom UI, and the previously mentioned GameList will handle the references for you [here](#). Create a new UserControl within (Wpf/Views/UserControls/CustomControls).

the CustomGameControlTemplate.xaml and CustomGameControlTemplate.xaml.cs files are template files you can use as a guide to create your own custom game UI.

### Icons & Button Icons

Icons for UI can be found on [Modern Ui Icons](#), or new custom path data icons can be created using [this](#) website or any other svg path editor. (respective licenses can be found [here](#))

## Logging

Within the application, logs and errors can be handled through the `ConsoleLogger`. There are two main functions used here.

- `ConsoleLogger.WriteLine(object data, string message, Exception exception, bool unique)`
- `ConsoleLogger.WriteLine(object data, string message, Exception exception, bool unique, Exception exception)`

### **WriteToConsole**

Allows you to write directly to the console window that opens up at the start of the launcher. Any exceptions passed into this function will automatically be written into a log file saved at `%AppData%/Local/LightSpace_WPF_Engine`.

A single file will be saved **per** launch of the application if errors are logged, and the file is named after the millisecond the error occurs. Anything passed to the console via this way could be logged if the `SaveExceptionsOnly` bool is set to **false**, however do note that many writing actions will take it's toll on the application.

### **WriteToUiConsole**

Allows you to write directly to the User Interface console situated below the rendering views. This message is also directly sent to the `WriteToConsole` in the background, so that errors (or any message in case of `SaveExceptionsOnly` being **false**) appears in the general log and can be logged to a file.

Using this action is **slower** than `WriteToConsole` and should only be used to inform the user about critical information regarding the application.

## Programma van eisen Embedded fitness

---

**Project:** De vloer is lava  
**Client:** Embedded fitness  
**Project Number:** 1.0

**Auteur:** Sam Hendriks  
**Date:** 25-5-2020  
**Version:** 1.1

---

---

The undersigned declare their agreement with the content of this Project Plan document

**Client**

***Initial Seen:***

***Initial Seen:***

**Project Manager**

Date: \_\_\_\_\_

Place: \_\_\_\_\_

Date: \_\_\_\_\_

Place: \_\_\_\_\_





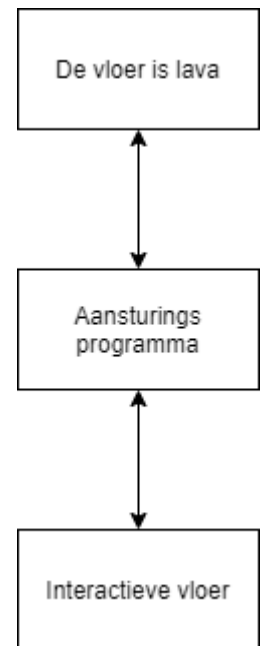
## Verzamelde informatie

Alle verzamelde informatie over het project en eventuelen documenten of oude software.

### Pitch

#### De presentatie

In de pitch hebben wij een presentatie gegeven aan de klant over wat wij precies willen maken. Hierin is in voort gekomen dat wij een spel("De vloer is lava") willen maken die een interactieve vloer aanstuurt met behulp van een aansturings programma. Verder hadden wij de voorkeur om in visual studio te werken om het spel te maken. Om dit te laten werken moet visual studio met het besturings programma kunnen communiceren. Dit gaat makkelijk omdat het besturings programma en de game in hetzelfde Visual studio project gemaakt kunnen worden. De games hebben een aparte folder waarin gewerkt kan worden die dan communiceert met het aansturings programma. Om de planning bij te houden hebben we eerst een projectplan gemaakt om een planning te maken. Daarnaast gaan we trello gebruiken om bij te houden waar we zijn in het project en hoeveel nog gedaan moet worden. Om te zorgen dat iedereen er bij kan maken wij gebruik van een onedrive en github gecombineerd met gitkraken. Dit zorgt voor een dubbel backup en dat je kunt zien wat wanneer is gebeurt.



#### Feedback

Na de presentatie heb ik gevraagd voor eventuele toevoegingen aan het spel. Omdat het een nieuw project is en hier nog niet eerder aan is gewerkt was ik vrij in het kiezen van wat ik wilde doen. Toch was er een toevoeging die de klant er graag bij zou willen hebben. de klant zou graag willen hebben dat de speler kan zien waar hij/zij op het moment staat. Dit kan voor de speler handig zijn om in een tegel te blijven en zelfs om het spel te debuggen.

#### Documenten

Momenteel zijn er geen documenten beschikbaar vanwege dat er nog niet eerder aan gewerkt is. De software(het aansturings programma) waarop het spel gaat werken is gemaakt door embedded fitness. Het aansturings programma stuurt de interactieve vloer aan. Daarnaast geeft het aansturings programma de input en outputs aan van de vloer. Het aansturings programma zelf is gemaakt in visual studio. Over de software van embedded fitness is een klein beetje documentatie. Hierin word vooral gekeken naar de specificaties waarin het aansturings gemaakt word. Daarnaast word er ook uitgelegd hoe je een spel kan maken op het aansturings programma. De documentatie staat allemaal op github en word in een aparte Word document geplaatst(Documentatie-aansturing).

## Afspraken

Hierin leggen we afspraken vast die zijn gemaakt tijdens de pitch. Zo hebben we altijd een overzicht van wat er is afgesproken en als er ooit een probleem is met het project kunnen we naar de afspraken kijken of het de bedoeling was of niet. De afspraken die zijn gemaakt staan hieronder.

- Gebruik maken van visual studio
  - De code voor het spel word geschreven in visual studio
- Gebruik maken van Github
  - De code die geschreven is word opgeslagen en beheert in github
- Gebruik maken van Gitkraken
  - De versiebeheer en het opslaan in github word uitgevoerd in gitkraken
- Gebruik maken van onedrive
  - De code word opgeslagen in onedrive
- Gebruik maken van trello
  - De planning en werkzaamheden worden in trello benoemd
- Communicatie tussen spel en het aanstuurprogramma
  - Het spel kan communiceren met het aansturings programma om functies uit te voeren
- Uitlezen sensoren
  - Het detecteren wanneer een persoon op de vloer staat doormiddel van de sensoren
- Aansturen ledjes
  - Het aangeven waar een persoon mag staan en waar niet doormiddel van de ledjes in de vloer
- Game start signaal
  - Het moet voor de gebruiker duidelijk zijn wanneer het spel begint.
- Game end signaal
  - Het moet voor de gebruiker duidelijk zijn wanneer het spel is geëindigt
- RandomSelector voor random veilige locatie op de vloer
  - De veilige locaties op de vloer worden willekeurig uitgekozen
- Versnellen van het spel
  - Elke keer als een persoon op een veilige plek staat zal het spel zichzelf versnellen
- Het bijhouden van de speler zijn positie
  - De vloer gaat kijken naar sensoren die geactiveerd zijn. Als een tegel een sensor actief heeft word op deze tegen een extra kleur toegevoegd zodat de speler weet waar hij/zij zich bevind.
- (Optioneel)Moeilijkheden kiezen
  - Bij het opstarten van het spel kan de gebruiker kiezen op welke moeilijkheidsgraad zij willen spelen.

## Eindresultaat

Het eindresultaat is een interactief spel waar je op de juiste plekken op een interactieve vloer moet gaan staan. De plaatsen waar je kan staan veranderen constant en veranderen elke keer iets sneller totdat je het niet meer bij kan houden. Zodra je af bent geeft de vloer dit aan doormiddel van het laten knipperen van de vloer zelf.

## **Impact**

Hierin komt te staan welke impact het project heeft op de betrokkenen.

### ***Embedded fitness***

De klant krijgt uiteindelijk een nieuw spel waarop verder gebouwd kan worden. Dit kan leiden tot nieuwe spellen die gebaseert worden op het spel dat gemaakt gaat worden. Verder kan de klant het spel nog uitbreiden. Daarnaast is de vloer ook functioneel om geld aan te verdienen zodra het spel af is en kan het spel gebruikt worden in demonstraties voor potentiële klanten.

### ***Het team***

Uit het maken van het spel word veel ervaring opgedaan, dit komt zowel door een andere manier van coderen en het samenwerken met een programmeur op een hoger niveau.

## Project Plan Embedded fitness

---

**Project:** De vloer is lava  
**Client:** Embedded fitness  
**Project Number:** 1.1

**Auteur:** Sam Hendriks  
**Date:** 25-5-2020  
**Version:** 1.1

---

---

The undersigned declare their agreement with the content of this Project Plan document

**Client**

***Initial Seen:***

***Initial Seen:***

**Project Manager**

Date: \_\_\_\_\_

Place: \_\_\_\_\_

Date:

Place:

## Inleiding

Het project “de vloer is lava” is een nieuw spel voor Embedded fitness. Het spel wordt gemaakt in visual studio en gaat via een aansturingen programma dat wordt geleverd door Embedded fitness communiceren met een interactieve vloer. In het projectplan komt te staan wat het project is, wat het doel van het project is, wat er gedaan moet worden om dat doel te behalen, de moscow lijst, wat we uiteindelijk hebben, de planning, en als laatste nog een aantal risico's.

## Project assignment

### Project omgeving

Het project wordt geprogrammeerd in visual studio in een subfolder van het aansturingen programma. Om het project op te slaan gebruiken we github en om veranderingen bij te houden en door te geven gebruiken we gitkraken. Voor de planning wordt er gebruik gemaakt van Trello, dit gaan wij ook gebruiken voor de manier van werken. In trello gaan wij de Scrum werkmethode toevoegen en wekelijks kijken wat er die week af moet zijn.

### Project doel

Het doel van het project is een functioneel spel maken voor een interactieve vloer. Dit spel hoort een element van sportiviteit te hebben en wat beweging. Het spel zelf is een demo voor potentiële klanten die interesse hebben in de interactieve vloer. Met dit spel geef je dan aan wat ongeveer de mogelijkheden zijn en wat de klant zou kunnen verwachten in de toekomst.

## Project activiteiten

- **Hoe behalen we de resultaten die we willen behalen**
  - Om het uiteindelijke doel te behalen hebben wij een planning nodig die na gekomen wordt. Om dit goed uit te voeren maken wij gebruik van Trello, hierin kan je goed zien wat er nog gedaan moet worden en wat al gedaan is. Daarnaast is een redelijke kennis nodig van visual studio en deels van het aansturingen programma zelf.
  - Om kennis op te halen van het aansturingen programma hebben we een gesprek met de programmeur die dit heeft gemaakt. Hiervoor hebben we een interview gehouden met de klant van wat wij in gedachten hadden voor het project. Nadat wij dit hebben gedaan kan de klant aangeven wat zij anders willen en dit wordt dan aan het project zelf veranderd.
  - Om de communicatie zelf goed te onderhouden geven wij dagelijkse updates van de status van het project. Deze communicatie gaat voornamelijk over de mail. Op deze manier indien dit nodig is kunnen wij ook vragen stellen over het project als iets onduidelijk is of omdat we bij een onderdeel hulp nodig hebben.
  - Zodra het project is afgerond moet dit ook getest worden dit gaan we doen aan de hand van een testlijst en testplan. Het testen zelf wordt eerst gedaan door de programmeurs zelf en daarna door mensen die niets van het project afweten. Zo kunnen wij goed zien wat een gebruiker zal doen die nog geen of weinig ervaring heeft gehad met het project.

- **Welke activiteiten moeten behaald worden om het doel te behalen**
  - De project omgeving moet correct worden ingesteld.
    - Visual studio .NET framework 4.8
    - C# & XAML capable IDE (Visual studio / VSCode recommended)
  - Interview met de klant.
  - Interview met de programmeur van het aansturings programma
  - Voldoende documentatie
    - Programma van eisen, projectplan, functioneel ontwerp, technisch ontwerp etc.
  - Er moet een functioneel spel gemaakt worden in het aansturings programma.
  - Het functioneel spel moet getest worden door de programmeur en door mensen die niks van het project weten.

## Project grenzen

- **Wat gaan we doen**
  - Communicatie tussen klant en projectleider.
    - Dit gaan we doen door zoveel mogelijk updates te geven over het project.  
Daaronder valt hoe ver het project is en waar iedereen mee bezig is.
  - De planning van het project nakomen.
  - De afgesproken Moscow lijst uitvoeren.
- **Wat gaan we niet doen**
  - Zonder communicatie aanpassingen maken.
  - Hardware aanpassen of er aan werken.
  - Aan onderdelen van het aansturings programma werken.

## Moscow

De uitgebreide uitleg van de onderstaande items kunt u terugvinden in het Programma van eisen.

- **Must**
  - randomSelector voor willekeurige veilige plaats op de vloer
  - Versnellen van het spel
  - Uitlezen sensoren interactieve vloer
  - Aansturen ledjes interactieve vloer
  - Game start signaal
  - Game end signaal
  - Communicatie tussen spel en het aanstuurprogramma
  - Het bijhouden van de speler zijn locatie
- **Should**
  - Commentaar bij de code(niet functioneel)
  - Gebruik van github en gitkraken(niet functioneel)
  - Gebruik maken van visual studio(niet functioneel)
  - Gebruik maken van onedrive(niet functioneel)
  - Gebruik maken van trello(niet functioneel)
- **Could**
  - MoeilijkheidsGraad uitkiezen
- **Won't**

## Producten

Wanneer het project klaar is hebben we 3 producten die met elkaar het gehele project maken. Het eerste product is het spel zelf. Het spel zelf is het programma dat is geschreven in C# en met functies van het aansturings programma. In het spel staat wat wanneer moet gebeuren. Zoals de random locatie waar je kan staan en waar niet, hiernaast ook wat er gebeurt als je op een locatie staat waar je niet mag staan. Daarnaast zal het spel ook steeds versnellen.

Het tweede product is de applicatie die de interactieve vloer aanstuurt. Deze applicatie zorgt ervoor dat alles wat de spellen van visual studio stuurt ook word uitgevoerd op de interactieve vloer. Zoals wanneer er ledjes aan of uit moeten gaan. Daarnaast zal deze applicatie ook de sensoren uitlezen en dit doorgeven aan de spellen in visual studio.

Het laatste product is de interactieve vloer zelf. De interactieve vloer heeft sensoren en ledjes ingebouwd waarmee hij kan aangeven waar iets zich bevind op dat moment. Om zelf iets uit te voeren moeten er eerst commando's aan de vloer gegeven worden met de aansturings applicatie. De vloer zal zelf doorgeven aan de applicatie waar iets zich bevind op de vloer doormiddel van de sensoren.

## Project organisatie

- **Teamleden**
  - Sam Hendriks
    - Teamleider
    - Kennis in C# en embedded systems
    - Verantwoordelijk voor
      - C# applicatie
      - Documentatie
      - Eindproduct
      - Communicatie
- **De klant**
  - Embedded fitness
- **Communicatie**
  - Wekenlijks zal er een voortgangs gesprek zijn met de klant in hoever wij momenteel zijn.



## Planning

### Wat word er gebruikt

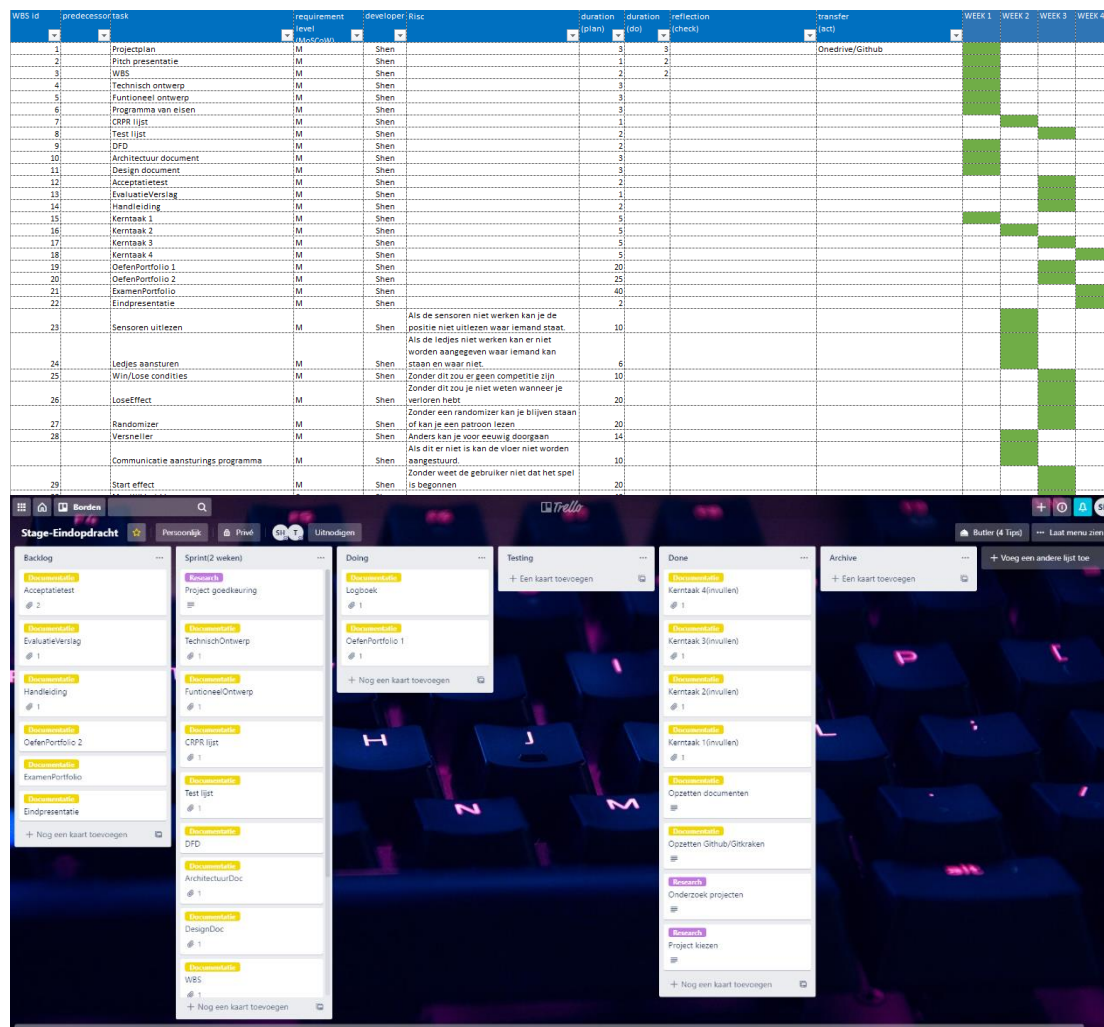
De planning van het project word gedaan met de hulp van een trello board. Hierin staat wat er om de twee weken geplanned staat, wat er nog gedaan moet worden, wat er gedaan word, wat getest word en wat er al af is.

Indien er onvoorziende omstandigheden zijn kan dit ook makkelijk worden aangepast met trello vanwege de flexibiliteit die Trello heeft. Daarnaast is er ook wekelijks een voortgang gesprek van hoe ver het project staat en of er aanpassingen nodig zijn in de plannen.

Bij de onderdelen die gedaan moeten worden kan de prioriteit makkelijk worden aangepast in trello. Dit kunnen we op elk moment doen terwijl de planning nog steeds duidelijk blijft.

### De planning

In de planning kijken we wat we gaan doen en wanneer iets gedaan word. Dit word allemaal uitgevoerd door de teamleider. Hieronder staat de wbs waarin staat wat gedaan moet worden en ook in welke week.



## Documentatie

Bij de documentatie komen een groot aantal delen bij kijken, daarom word hier 2 weken aan gespendeerd. De documentatie zal ook niet in een keer achter elkaar gedaan worden en dan verder naar de realisatie. Het is meer verspreid over heel de planning van het project waarbij meer en deel aan het begin word gemaakt.

## Realisatie

In de realisatie gaan we uitvoeren wat we allemaal hebben besproken met de klant en wat we hebben vastgelegd in de documentatie. In deze periode gaan wij werken aan het maken van het project zelf en hier een functioneel spel uit te krijgen dat samen kan werken met het aansturings programma. Het maken van het spel word in twee weken gedaan.

## Opleveren

Het oplever process is het opleveren van projectdelen en uiteindelijk het geheel in totaal.

Uiteindelijk hebben we een paar oplever momenten deze zijn:

- Opleveren project idee
- Examenafspraken
- Examenportfolio

## Risico's

- **Wat kan er fout gaan**
  - De communicatie tussen visual studio en het aansturings programma voor de interactieve vloer.
  - Slechte communicatie tussen klant en teamleider.
- **Hoe kunnen we dit oplossen**
  - Om de communicatie zo vlot mogelijk te laten verlopen is dat een prioriteit en word dit als een van de eerste dingen getest.
  - Om een goede communicatie tussen klant en teamleider te behouden zorgen we dat we wekenlijks een vast moment hebben om de status van het project te bespreken en eventuelen veranderingen.

## Functioneel ontwerp Embedded fitness

---

**Project:** De vloer is lava  
**Client:** Embedded fitness  
**Projectnumber:**1.0

**Auteur:** Sam Hendriks  
**Date:** 25-05-2020  
**Version:** 1.1

---

---

**Client**

**Initial Seen:**

**Initial Seen:**

**Projectmanager**

**Date:** \_\_\_\_\_

**Place:** \_\_\_\_\_

**Date:**

**Place:**

## Inleiding

In dit document word er gekeken naar de verantwoordelijkheden van ieder teamlid, de project details, een functionele project omschrijving, en een aantal diagrammen.

## Project management

- De leden van het project
  - Sam Hendriks
- De verantwoordelijkheden van ieder lid
  - Sam Hendriks(Team leider)
    - Programmeur
    - C#
    - Planning
    - Communicatie met de klant
    - Communicatie met project gerelateerde personen
    - Documentatie
- Wanneer hebben de projectleden een standaard meeting
  - De meeting word iedere maandag uitgevoerd.

## Project details

Project naam: De vloer is lava

Client naam: Embedded fitness

Naam aannemer: Sam Hendriks

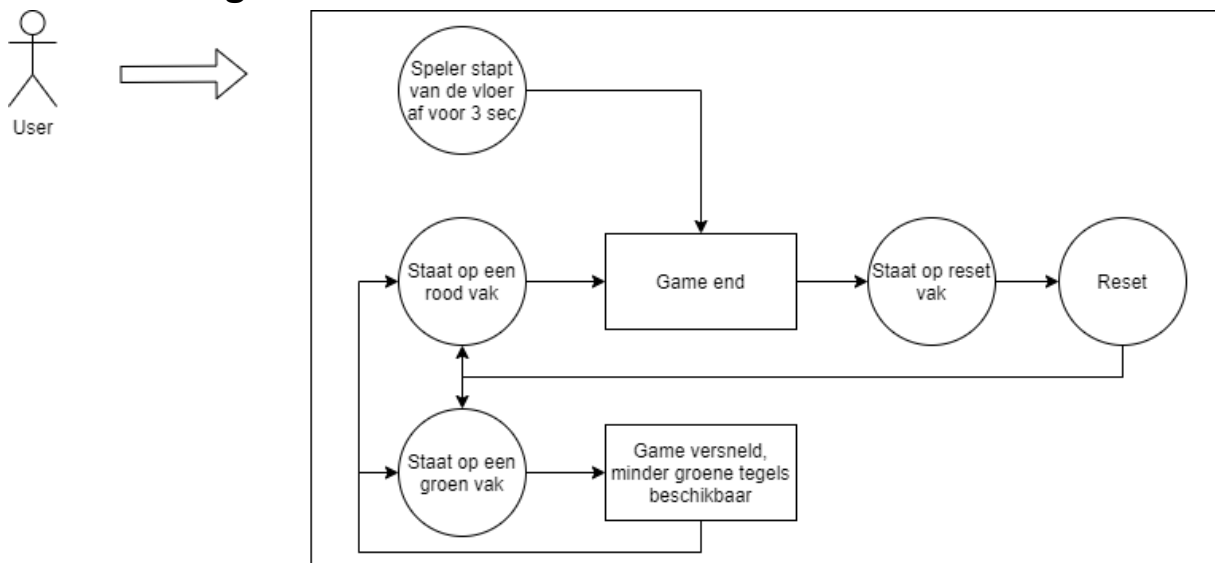
## Project omschrijving

De situatie nu is dat embedded fitness een interactieve vloer heeft waar nog niks mee is gedaan. recent is er een aansturings programma in aanmaak gegaan voor deze interactieve vloer. Momenteel hebben ze nog geen spellen of anderen projecten die op deze interactieve vloer gebruikt kunnen worden. "De vloer is lava" word daarom het eerste spel dat gemaakt word voor de interactieve vloer van Embedded fitness. Het spel zelf word gemaakt in combinatie met het aansturings programma. In het aansturings programma komt een folder te staan met ons spel. Hierin gaan wij aanroepen wat er moet gebeuren en wanneer. Het spel zelf is vrij simpel: zodra het spel start word er eerst een start signaal aangegeven. Dit signaal word duidelijk gemaakt door 1x rood te knippen dan 1x oranje en als laatst 1x groen. Als dit is gebeurt begint het spel. De bedoeling van het spel is om op de groene vakken te staan. Het aantal seconden dat je hiervoor krijgt is 10 it word elke ronde met 0.5% verminderd. Indien je op een van de rode vakken staat is het spel afgelopen. Dit word aangegeven door de hele vloer een aantal keer rood te laten knippen. Indien je op het groen staat gaat het spel verder. De locatie van de groene vakken zijn constant willekeurig. Dit blijft doorgaan totdat de speler af is en het eind signaal is aangegeven. Als een speler van de vloer afstapt word er bij het check moment dit zelfde eindsignaal gegeven. Om aan de speler duidelijk te maken waar zij staan word het deel waar de speler staat geel.

## Requirements

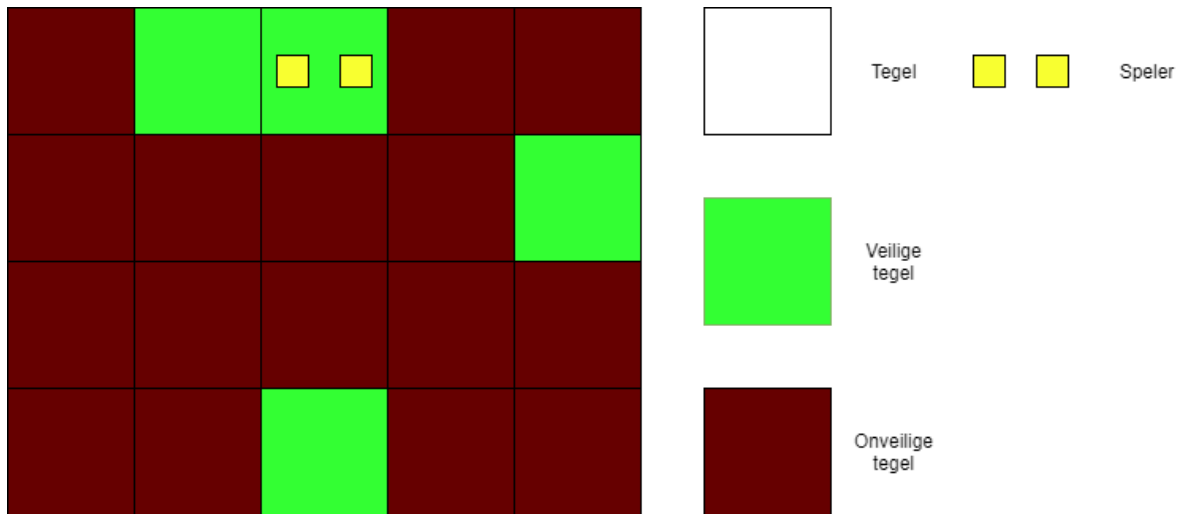
- **Must**
  - randomSelector voor willekeurige veilige plaats op de vloer
  - Versnellen van het spel
  - Uitlezen sensoren interactieve vloer
  - Aansturen ledjes interactieve vloer
  - Game start signaal
  - Game end signaal
  - Communicatie tussen spel en het aanstuurprogramma
  - Het bijhouden locatie speler
- **Should**
  - Commentaar bij de code(niet functioneel)
  - Gebruik van github en gitkraken(niet functioneel)
  - Gebruik maken van visual studio(niet functioneel)
  - Gebruik maken van onedrive(niet functioneel)
  - Gebruik maken van trello(niet functioneel)
- **Could**
  - MoeilijkheidsGraad uitkiezen
- **Won't**

## Use case diagram

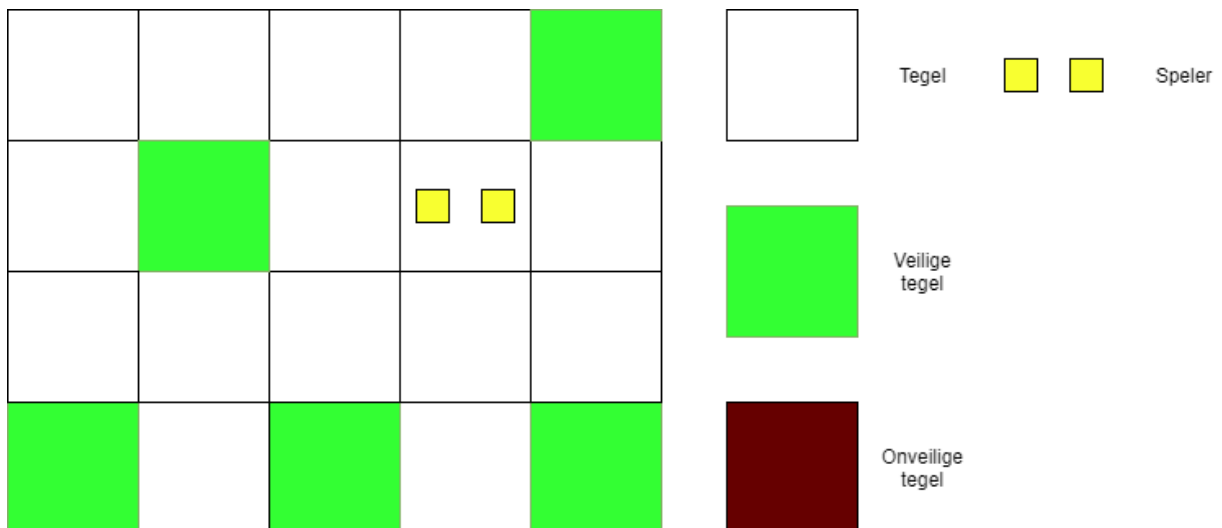


Als het spel aanstaat kan de gebruiker 2 dingen doen, hij kan of op een rood vak gaan staan of op een groen. Als de gebruiker op een rood vak gaat staan dan word de game end signaal uitgevoerd en kan de gebruiker resetten door op een reset vak te gaan staan waardoor deze weer bij het begin aankomt. Indien de gebruiker op een groen vak staat gaat het spel verder en begint het weer opnieuw.

## Wireframe

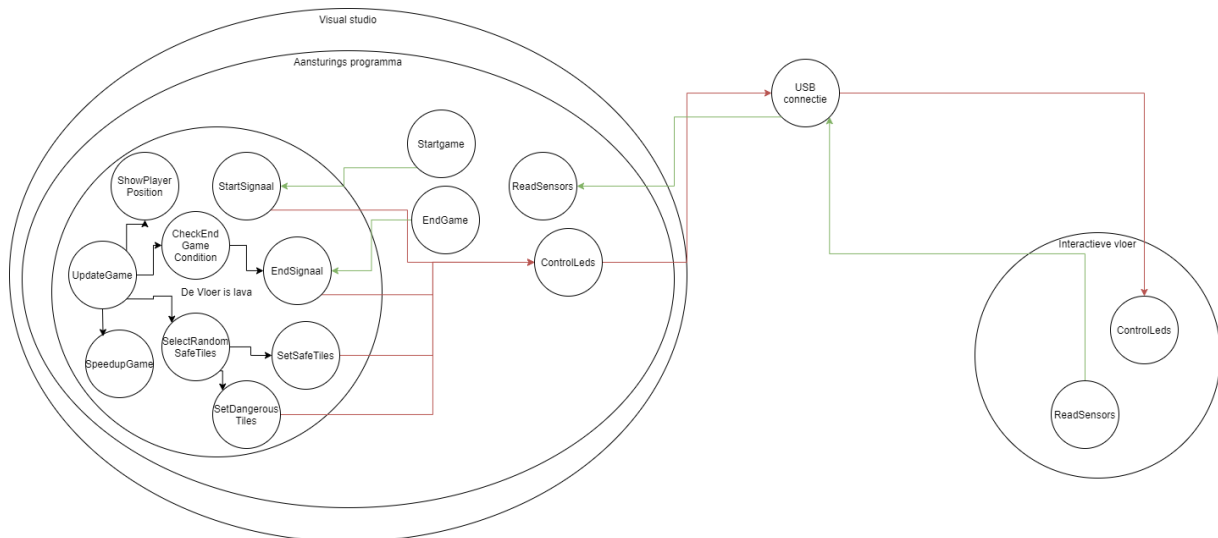


In de wireframe kan je zien hoe het er ongeveer voor de gebruiker uit gaat zien als deze op de vloer staat. Hierbij hebben we normale tegels die alleen gebruikt worden als de veilige tegels worden aangetoond. De veilige tegels zelf waar de gebruiker op kan staan en de onveilige tegels waar de gebruiker niet op kan staan.



Op de afbeelding hierboven kan de speler zien waar hij naartoe moet lopen de “veilige tegels”. De tegel waar de speler nu opstaat zijn nog neutraal.

## Flowchart



In de flowchart word een voorbeeld gegeven hoe de data van een functie naar de interactieve vloer verplaats word en terug. Ook valt er te zien waar welk onderdeel zich bevind, zoals dat het aansturings programma zich in visual studio bevind, het spel “De vloer is lava” zich in het aansturings programma bevind, en dat de interactieve vloer een los staant object is dat op zichzelf functioneert.

## Technisch ontwerp Embedded fitness

---

**Project:** De vloer is lava  
**Opdrachtgever:** Embedded fitness  
**Projectnummer:** 1.0

**Auteur:** Sam Hendriks  
**Datum:** 25-5-2020  
**Versie:** 1.1

---

---

**Opdrachtgever**

***Paraaf gezien:***

Datum: \_\_\_\_\_

Plaats: \_\_\_\_\_

**Projectmanager**

***Paraaf gezien:***

Datum: \_\_\_\_\_

Plaats: \_\_\_\_\_



## Inleiding

Dit technisch ontwerp is voor het project “De vloer is lave”. In dit document gaan we kijken naar de technische eisen in het project, welke componenten in het project zitten, een aantal diagrammen, hoe wij beslissingen regelen, de afspraken wie wat doet, de genzen wat zit er niet in het ontwerp, een deel van het plan van aanpak hierin komt te staan wat de showstoppers zijn, en het testplan. Om dit document goed door te kunnen lezen is er een basis kennis nodig in C# en over diagrammen die gebruikt worden bij programmeren.

## Technische Eisen

Hierin komt te staan hoe een onderdeel moet gaan werken, op welke bepaalde snelheid het programma moet draaien, en welke programmeertaal er word gebruikt.

### *Hoe gaat het werken*

Om het spel te laten werken is er een aansturings programma gemaakt dat de commando's van het spel verwerkt en naar de Interactieve vloer stuurt. Dit word dan uitgevoerd door de vloer zelf. De vloer leest ten alle tijden de sensoren uit en levert deze aan het aansturings programma. Het programma geeft deze door aan het spel en dan kan hierop gereageert worden. Dit moet allemaal in miliseconden worden uitgevoerd zodat het spel niet blijft hangen want een gebruiker kan hier last van hebben. Het aansturings programma en het spel worden allebei gemaakt in visual studio, daarnaast word er de programmeertaal C# gebruikt voor het coderen. Het spel zelf krijgt een GUI doormiddel van het aansturings programma. In het aansturings programma kan ik zelf een GUI maken die dan vertoont word bij het opstarten van het spel.

### Voorbeeld

Om bijvoorbeeld de tegels rechtboven groen te maken worden er een aanstal stappen gemaakt.

Het begint met het aanmaken van een background, de background is een clone van de orginele background. De orginele background is zwart, zodat tegels die niet worden gebruikt ook zwart blijven. De belangrijkste functie om een tegel te kleuren is de DrawRectangle(). Hierin word de informatie van de tegel gezet die word aangepast, zoals de positie en kleur. Deze functie is van het aansturings programma zelf. Op het einde van een update word de draw functie uitgevoerd. Hierin word de background met alle aanpassingen gezet. Als dit is gedaan gaat het aansturings programma de data die verkregen is verwerken en dan doorgeven aan de interactieve vloer.

## Componenten en modules

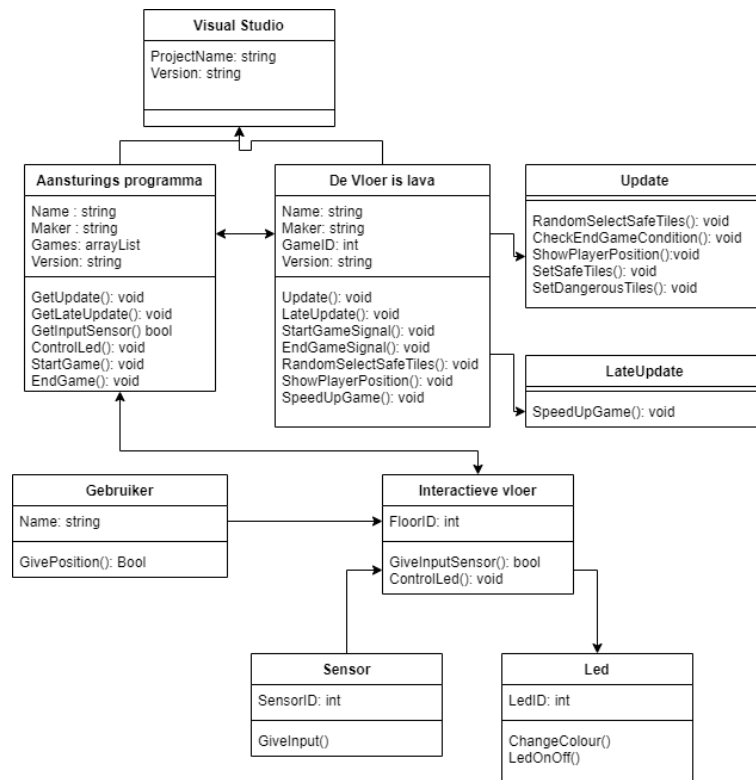
De volgende componenten zijn nodig om ons project functioneel te maken

- Interactieve vloer
  - Sensoren
  - ledjes
- Aansturings programma
  - Gemaakt in visual studio
  - Programmeer taal is C#
- “De vloer is Lava”
  - Gemaakt in visual studio
  - Programmeer taal is C#
  - Gemaakt in het aansturings programma
- Visual studio
  - .NET framework 4.8
  - C# & XAML capable IDE (Visual studio / VSCode recommended)

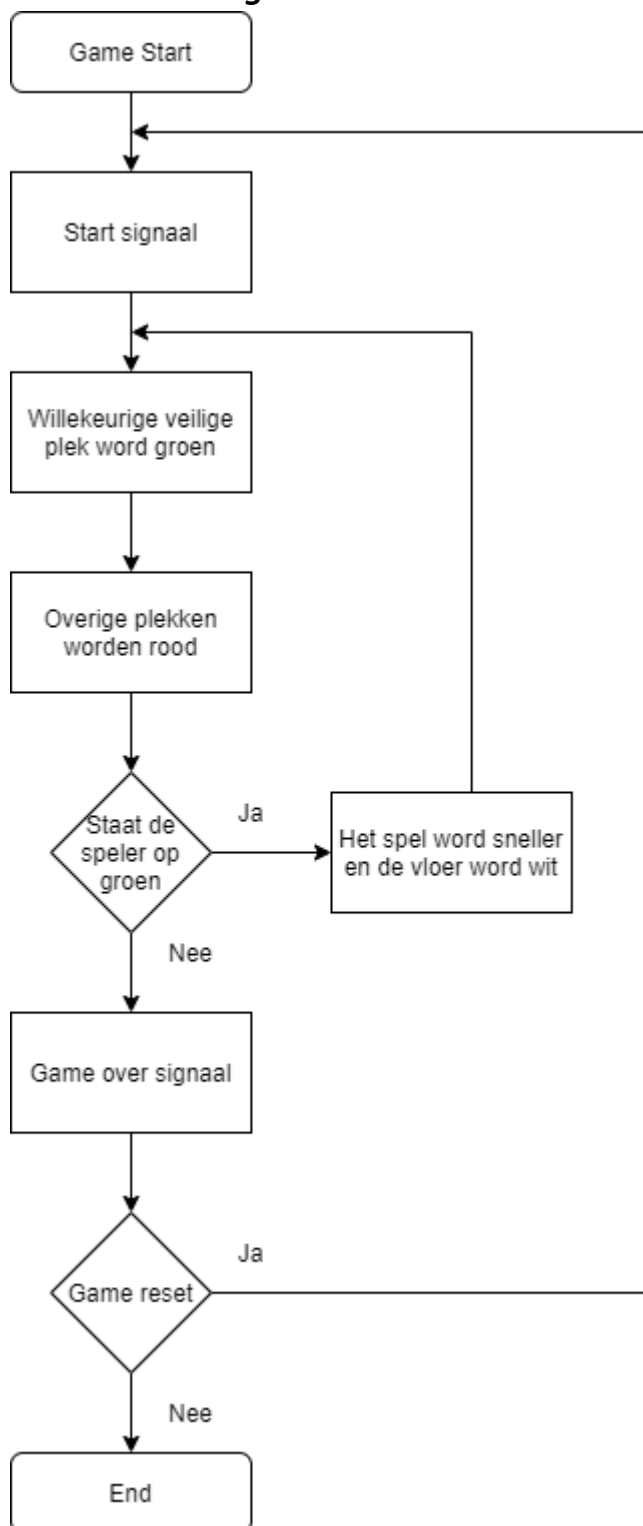
De interactieve vloer werkt samen met het aansturings programma, hierbij geeft het aansturings programma aan wat er moet gebeuren op de vloer zoals welke ledjes welke kleur moeten hebben. de interactieve vloer geeft ook aan wanneer iemand op een plaats staat met de sensoren die erin zitten. Dit geeft de vloer door aan het aansturings programma. Het aansturings programma werk ook samen met “De vloer is lava”. Hierbij geeft het aansturings programma door wat het van de vloer heeft gekregen, maar daarlangs kan “De vloer is lava” doorgeven aan het aansturings programma welke ledjes op welk punt aan moeten. Dit geeft het aansturings programma dan door aan de vloer zodat dit word uitgevoerd. Het aansturings programma en “De vloer is lava” worden allebei gemaakt in visual studio met de programmeer taal C#.

## Diagrammen

### Klassen diagram

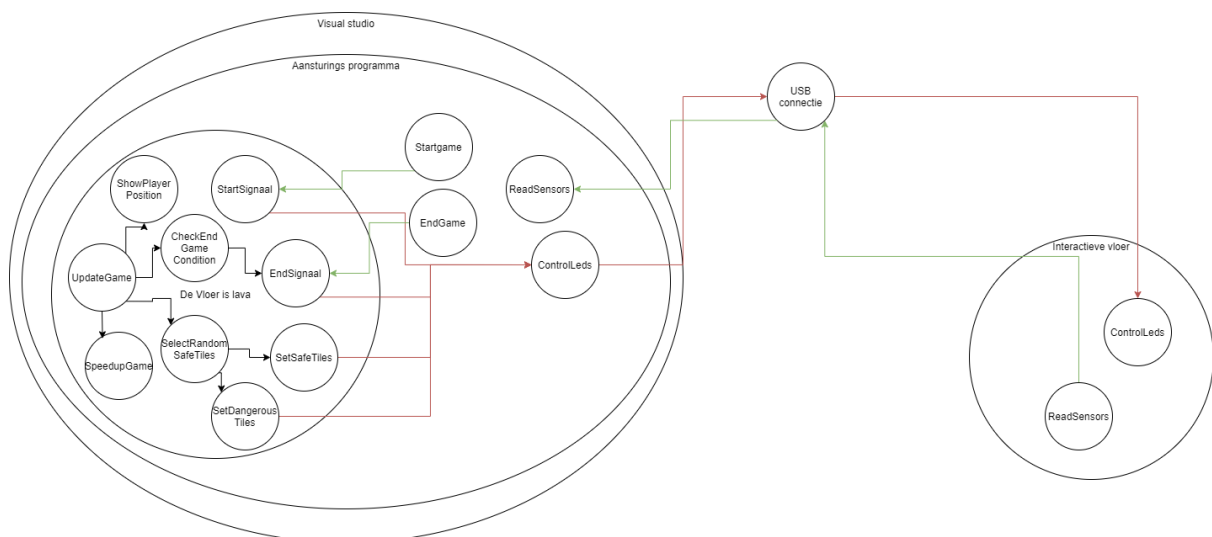


**Activiteiten diagram**



## Relationeel datamodel

In het relationeel datamodel gaan we kijken naar de communicatie tussen onderdelen in het project. Voor de communicatie van “de vloer is lava” naar de interactieve vloer moet “de vloer is lava” eerst communiceren met het aansturings programma. “de vloer is lava” is deel van het aansturings programma hierdoor is de communicatie simpel om uit te voeren. Om met elkaar te communiceren worden bepaalde methodes aangeroepen om functies uit te voeren. “de vloer is lava” stuurt naar het aansturings programma wat er moet gebeuren (welke ledjes welke kleur worden). Zodra het aansturings programma dit heeft ontvangen vertaalt dit het in een code waarmee de interactieve vloer kan werken. De code die het aansturings programma heeft gemaakt wordt via een usb connectie verstuurd naar de interactieve vloer. De interactieve vloer verwerkt deze code en voert deze uit. De interactieve vloer leest zelf ook sensoren uit, deze lezen uit of er ergens iets op de vloer staat. De sensoren geven een signaal als een sensor high of low is, dus deze geven niet constante data aan het aansturings programma. Wanneer er een sensor high of low wordt, gaat er eenzelfde soort stringcode (die het aansturings programma stuurt naar de interactieve vloer) via de usb connectie naar het aansturings programma. Bij het aansturings programma wordt dit verwerkt en doorgegeven aan “de vloer is lava”. Het spel verwerkt de code en reageert hierop met gameEnd of hij speelt verder.



- Versturen bericht
  - Byte 1&2 : Tegelnummer (16 bits = 65536 tegels)
  - Byte 3-66 : 64 bytes voor 64 leds in een tegel [XXBBRRGG] kleuren
- Ontvangen bericht
  - Byte 1 : Tegelgroep aanduiden (0 = tegel 0-31, 1 = tegel 32-63 etc)
  - Byte 2-65 : 32 byte paren waarbij elke paar de 16 capsense bytes (4\*4 sensors) terug zend naar de pc. Dus dan wordt er een batch van 32 tegels meegegeven

## Beslissingen

Indien er een beslissing genomen moet worden over een onderdeel of een keuze dan wordt dit bij de projectleider neergelegd. De projectleider zal dan met onder andere de projectleden en de klant overleggen wat het beste is om te doen. Uit dit overleg zal de teamleider beslissen wat er gaat gebeuren.

## Afspraken

Het project “de vloer is lava” bestaat alleen uit het spel hiervoor is de teamleider verantwoordelijk. De code en de documentatie die hierbij hoort moeten voor 29 mei af zijn zodat deze beoordeeld kunnen worden. Hiervoor word elk document wanneer deze klaar is bekeken door de klant totdat zij met het document akkoord gaan wat word aangegeven door een handtekening. De materialen die nodig zijn voor het project zijn visual studio, het aansturings programma, en de interactieve vloer. Visual studio word ingesteld door de teamleider. Het aansturings programma word gemaakt en geleverd door Rik van Maaren. Als laatst word de interactieve vloer opgezet door Embedded fitness zelf.

## Grenzen

In dit project zijn vooral de technische zaken vastgelegd hierdoor zijn er een aantal zaken niet in dit document vastgelegd. Voor de functionelen delen van het project hebben wij een document het functioneel ontwerp. Hierin staat alles wat met de functie van het project en de modules ervan te maken heeft. In dit document staat ook geen planning dit staat vast in het projectplan.

## Plan van Aanpak

Hier gaan we kijken naar de showstopper van het project. Dit zijn problemen die ervoor kunnen zorgen dat het project niet werkt of het hele project kunnen stil leggen. Onder de showstoppers vallen:

- Er is nog geen werkend aansturings programma
- Communicatie tussen spel en aansturings programma werkt niet
- Het aansturings programma kan niet communiceren met de interactieve vloer

Om dit te voorkomen testen wij deze specifieke showstoppers als eerst zodat we niet midden in het project vast komen te zitten. Hiernaast voeren wij een haalbaarheidstest uit zo weten wij of we het project kunnen afmaken voor onze einddatum. Nadat we dit hebben gedaan kunnen we verder met de rest van het project en komen we niet snel voor onverwachte problemen te staan.

### **Haalbaarheidsstappen**

Om de haalbaarheid te testen worden er een paar onderzoeken uitgevoerd om te kijken of het project haalbaar is. Dit gebeurt in stappen van een kleine test naar een groter doeleinde hiermee kunnen we kijken of het project haalbaar is. Hieronder staat een voorbeeld van hoe zo iets voor ons eruit kan zien.

Haalbaarheidsstappen	
1.	Spel opzetten in het aansturings programma
2.	Simpelen commando's sturen naar het aansturings programma om een ledje aan te zetten en een sensor uit te lezen
3.	Een simpele test doen waarbij het spel via het aansturings programma de vloer bestuurd

Met deze testen zouden wij kunnen zien of wij dit deel van het project kunnen halen in de aangeven tijd. Dit zou op zijn minst een schatting kunnen opleveren waardoor wij verder kunnen gaan met het project. De stappen zoals te zien is zijn kleine makkelijke stappen naar iets moeilijker.

### **Test plan**

Om ons project te testen gebruiken wij een testlijst de testlijst staat in een excel bestand. In dit bestand leggen wij vast wanneer we aan het testen zijn en wat we als uitkomst verwachten. Na het testen schrijven we ook het resultaat op en of het gewenste resultaat behaald is. Bij elk nieuw onderdeel aan code dat geschreven wordt, word heel de testlijst uitgevoerd. Naast dat de programmeurs gaan testen vragen wij ook aan mensen die niks weten van het project om een keer te testen. De personen die getest hebben vullen dan een testraport in die wij kunnen inkijken. Alle delen die getest moeten worden zijn van te voren al in de testlijst gezet. Als een van deze delen getest wordt schrijft de persoon die dat deel test zijn naam erbij zodat alle projectleden weten wie wat getest heeft. Zo zal er geen verwarring komen of iets nog getest moet worden en voor meer informatie weet je dan bij wie je moet zijn.

## Vorbereiding realisatie Embedded Fitness

---

**Project:** De vloer is lava  
**Opdrachtgever:** Embedded fitness  
**Projectnummer:** 1.0

**Auteur:** Sam Hendriks  
**Datum:** 25-5-2020  
**Versie:** 1.1

---

---

**Opdrachtgever**

***Paraaf gezien:***

Datum: \_\_\_\_\_

Plaats: \_\_\_\_\_

**Projectmanager**

***Paraaf gezien:***

Datum: \_\_\_\_\_

Plaats: \_\_\_\_\_



## Benodigheden

De benodigheden voor het project de vloer is lava zijn.

- Aansturings programma
  - Kan met de interactieve vloer communiceren
  - Kan met “de vloer is lava” communiceren
  - Kan de ledjes aansturen
  - Kan de sensoren uitlezen
- Interactieve vloer
  - Heeft multicolour ledjes
  - Heeft druksensoren
  - Kan met het aansturings programma communiceren
- Visual studio
  - Visual studio .NET framework 4.8
  - C# & XAML capable IDE (Visual studio / VSCode recommended)

De bovenstaande items zijn nodig om het project goed uit te kunnen voeren.

Hieronder komen de benodigheden die we gaan gebruiken maar niet nodig zijn om het project goed uit te voeren.

- Git
- Gitkraken
- Trello

## Werkomgeving

De werkomgeving waar het project gemaakt zal worden is op een Msi gpt26qf leopard. Op deze laptop maken wij gebruik van visual studio 2017 met een .NET framework 4.8. Visual studio moet C# en XAML kunnen gebruiken. Dit is nodig voor het aansturings programma zelf.

## Testcase Embedded Fitness

---

**Project:** De vloer is lava  
**Opdrachtgever:** Embedded fitness  
**Projectnummer:** 1.0

**Auteur:** Sam Hendriks  
**Datum:** 25-5-2020  
**Versie:** 1.1

---

---

**Opdrachtgever**

***Paraaf gezien:***

Datum: \_\_\_\_\_

Plaats: \_\_\_\_\_

**Projectmanager**

***Paraaf gezien:***

Datum: \_\_\_\_\_

Plaats: \_\_\_\_\_

gemaakt door de testplanschrijver, naam: Sam Hendriks					gemaakt door de tester, naam: Sam Hendriks	
nr	wat ga je testen	wb	bb	grenswaarden van de testwat verwacht je?	ok	inokopmerking
1	RandomGoodTilesSelector			De game kiest willekeurig de tegels uit waar de speler op kan staan dit word aangegeven door de tegels groen te maken.	x	De tegels worden goed uitgekozen en in een tegelArray gestopt dat hiervoor ontworpen is. Tijdens de update word er via een foreach elke tegel bekeken en de kleur op groen hiervoor gezet. Dit word dan doorgegeven aan het aansturingsprogramma.
2	DecreaseGoodTiles			Het spel geeft bij elke ronde minder tegels weer om op te kunnen staan.	x	Bij elke ronde word er een tegel afgehaald.(dit kan soms anders lijken omdat er nog een mogelijkheid is dat er 2 tegels op dezelfde plek zitten.)
3	GameSpeedIncrease			Het Spel word bij elke ronde iets sneller met een standaard percentage.	x	Bij elke ronde word het spel iets sneller uitgevoerd.
4	ReadActiveSensor			Het spel kan in een tegel sensoren lezen of hier een sensor actief is.	x	De functie van het aansturingsprogramma stond er nog niet(of op de verkeerde plek) in en leed tot een error wanneer dit uitgeprobeert werd.
5	SetLeds			Het spel kan het aanzetten en bewerken van de ledjes.(De kleuren van de ledjes instellen)	x	Alle tegels die worden aangepast kwa kleur worden goed weergeven.
6	GameStartSignal			Het spel geeft een startsignaal geven bij het begin van het spel. Er dit signaal hoort 3 kleuren te geven in de volgorde rood, oranje, en groen.	x	Het word niet uitgevoerd wanneer ik het spel start.
7	GameEndSignal			Het spel geeft een eindsignaal wanneer het spel is afgelopen. Dit doet die door een aantal keer rood te knippen.	x	Wanneer het spel is afgelopen gaat de vloer een aantal keer rood knippen om aan te geven dat het spel voorbij is.
8	PlayerTracker			Het spel detecteert waar de speler staat en maakt een lijn aan de rand van de tegel geel.	x	De lijn die ik bij de tegel wil hebben word niet weergeven.
9	GameReset			Als het spel is afgelopen word er een tegel geel. Als een sensor hierop word geactiveerd dan begint het spel opnieuw.	x	Wanneer ik een aparte tegel wilde pakken crashde het programma.

Figuur 6 Testcase 1

gemaakt door de testplanschrijver, naam: Sam Hendriks					gemaakt door de tester, naam: Sam Hendriks	
nr	wat ga je testen	wb	bb	grenswaarden van de testwat verwacht je?	ok	opmerking
1	RandomGoodTilesSelector			De game kiest willekeurig de tegels uit waar de speler op kan staan dit word aangegeven door de tegels groen te maken.	x	De tegels worden goed uitgekozen en in een tegelArray gestopt dat hiervoor ontworpen is. Tijdens de update word er via een foreach elke tegel bekeken en de kleur op groen hiervoor gezet. Dit word dan doorgegeven aan het aansturingsprogramma.
2	DecreaseGoodTiles			Het spel geeft bij elke ronde minder tegels weer om op te kunnen staan.	x	Bij elke ronde word er een tegel afgehaald.(dit kan soms anders lijken omdat er nog een mogelijkheid is dat er 2 tegels op dezelfde plek zitten.)
3	GameSpeedIncrease			Het Spel word bij elke ronde iets sneller met een standaard percentage.	x	Bij elke ronde word het spel iets sneller uitgevoerd.
4	ReadActiveSensor			Het spel kan in een tegel sensoren lezen of hier een sensor actief is.	x	De functie van het aansturingsprogramma stond er nog niet(of op de verkeerde plek) in en leed tot een error wanneer dit uitgeprobeert werd.
5	SetLeds			Het spel kan het aanzetten en bewerken van de ledjes.(De kleuren van de ledjes instellen)	x	De functie is geplaatst op de juiste plek en kan nu aangesproken worden. Deze functie word ook juist uitgevoerd.
6	GameStartSignal			Het spel geeft een startsignaal geven bij het begin van het spel. Er dit signaal hoort 3 kleuren te geven in de volgorde rood, oranje, en groen.	x	Alle tegels die worden aangepast kwa kleur worden goed weergeven.
7	GameEndSignal			Het spel geeft een eindsignaal wanneer het spel is afgelopen. Dit doet die door een aantal keer rood te knippen.	x	Het word niet uitgevoerd wanneer ik het spel start.
8	PlayerTracker			Het spel detecteert waar de speler staat en maakt een lijn aan de rand van de tegel geel.	x	Wanneer het spel start word het startsignaal weergeven.(er is nog een mogelijkheid dat er een van de kleuren niet vertoont word.)
9	GameReset			Als het spel is afgelopen word er een tegel geel. Als een sensor hierop word geactiveerd dan begint het spel opnieuw.	x	Wanneer het spel is afgelopen gaat de vloer een aantal keer rood knippen om aan te geven dat het spel voorbij is. De lijn die ik bij de tegel wil hebben word niet weergeven.
						Het kleuren van de tegel heb ik op een andere plek in de code neergezet en werkt nu als bedoeld. De speler kan zien waar hij/zij op dat moment staat.
						Wanneer ik een aparte tegel wilde pakken crashde het programma.
						Inplaats van een aparte tegel pakken heb ik een array gemaakt waarin de reset tegel staat. Wanneer de speler op de tegel staat wanneer deze is weergeven start het spel zich opnieuw op.

Figuur 5 Testcase 2

## AcceptatieTest Embedded Fitness

---

**Project:** De vloer is lava  
**Opdrachtgever:** Embedded fitness  
**Projectnummer:** 1.0

**Auteur:** Sam Hendriks  
**Datum:** 25-5-2020  
**Versie:** 1.1

---

---

**Opdrachtgever**

***Paraaf gezien:***

Datum: \_\_\_\_\_

Plaats: \_\_\_\_\_

**Projectmanager**

***Paraaf gezien:***

Datum: \_\_\_\_\_

Plaats: \_\_\_\_\_

**Geschreven  
door, naam: Sam  
Hendriks**

**Getest door,  
naam: Jason  
Pranger**

nr	Wat word er verwacht	ok	nok	opmerking
1	Het spel geeft een start signaal aan voordat het spel begint.	X		Langzamer laten gaan, het flitst nu teveel.
2	Het spel laat eerst de groene tegels zien voordat je de rode tegels ziet.	X		
3	Als je niet op de vloer aanwezig bent geeft het spel een eindsignaal.	x		
4	Als je op een rode tegel staat geeft het spel een eindsignaal.	X		
5	Het spel geeft bij een verlies conditie een eindsignaal	X		
6	Het spel houdt bij waar de speler is doormiddel van een gele lijn aan de binnenkant van een tegel.	X		
7	Het spel laat een resetknop zien na het eindsignaal.	x		Op een tegel zetten, waar de speler niet op staat.
8	Als de speler op de resetknop staat herstart het spel.	x		
9	Het spel word bij iedere ronde iets sneller.	x		
10	Het spel heeft bij iedere ronde een tegel minder.	x		
				Handtekening:  <b>Jason Pranger</b>

**Geschreven  
door, naam: Sam  
Hendriks**

**Getest door,  
naam: Carola**

nr	Wat word er verwacht	ok	nok	opmerking
1	Het spel geeft een start signaal aan voordat het spel begint.	x		Misschien iets langzamer in beeld laten komen, dan kan je het start signaal niet missen.
2	Het spel laat eerst de groene tegels zien voordat je de rode tegels ziet.	x		
3	Als je niet op de vloer aanwezig bent geeft het spel een eindsignaal.	x		
4	Als je op een rode tegel staat geeft het spel een eindsignaal.	x		Misschien kun je in de foute tegel een kruis zetten en dan het eind signaal, dan is het dubbel duidelijk.
5	Het spel geeft bij een verlies conditie een eindsignaal	x		
6	Het spel houdt bij waar de speler is doormiddel van een gele lijn aan de binnenkant van een tegel.	x		
7	Het spel laat een resetknop zien na het eindsignaal.	x		
8	Als de speler op de resetknop staat herstart het spel.	x		
9	Het spel word bij iedere ronde iets sneller.	x		

10	Het spel heeft bij iedere ronde een tegel minder.	x		
				Handtekening: Carola

**Geschreven door, naam: Sam Hendriks**

**Getest door, naam: Rik**

nr	Wat word er verwacht	ok	nok	opmerking
1	Het spel geeft een start signaal aan voordat het spel begint.	x		Duidelijke kleurkeuze door het gebruik van kleuren bekend van stoplichten.
2	Het spel laat eerst de groene tegels zien voordat je de rode tegels ziet.	x		
3	Als je niet op de vloer aanwezig bent geeft het spel een eindsignaal.	x		
4	Als je op een rode tegel staat geeft het spel een eindsignaal.	x		
5	Het spel geeft bij een verlies conditie een eindsignaal	x		
6	Het spel houdt bij waar de speler is doormiddel van een gele lijn aan de binnenkant van een tegel.	x		
7	Het spel laat een resetknop zien na het eindsignaal.	x		Misschien handig om hier wat informatie over te geven in de beschrijving van je spel.
8	Als de speler op de resetknop staat herstart het spel.	x		

<b>9</b>	Het spel word bij iedere ronde iets sneller.	x		Is moeilijkheid instelbaar? Zou een leuke toevoeging kunnen zijn.
<b>10</b>	Het spel heeft bij iedere ronde een tegel minder.	x		
				Handtekening: Rik



## Opleverproces Embedded Fitness

---

**Project:** De vloer is lava  
**Opdrachtgever:** Embedded fitness  
**Projectnummer:** 1.0

**Auteur:** Sam Hendriks  
**Datum:** 25-5-2020  
**Versie:** 1.1

---

---

**Opdrachtgever**

***Paraaf gezien:***

Datum: \_\_\_\_\_

Plaats: \_\_\_\_\_

**Projectmanager**

***Paraaf gezien:***

Datum: \_\_\_\_\_

Plaats: \_\_\_\_\_

## Inleiding

In dit document ga ik kijken naar hoe het opleverproces is gegaan. hierbij komt het opleveren van het product aan bot en het uitleggen en bewijzen van het werkproces.

## Opleveren

### ***Product***

Voor het opleveren van het product hebben we eerst de broncode doorgelopen. Hierbij hebben we gekeken naar welke functie er gebruikt worden en wanneer ze gebruikt worden. Verder hebben we gekeken naar de structuur van de code of deze wel netjes is opgesteld. Daarna heb ik een demonstratie gegeven van het product door een demo versie aan te zetten. Bij de demo versie kun je al een idee krijgen hoe het spel er uiteindelijk uit komt te zien.

### ***Process***

Bij het opleveren van het process hebben we gekeken hoe ik te werk ben gegaan en wat ik allemaal heb kunnen leveren.

## Evaluatie verslag Embedded fitness

---

**Project:** De vloer is lava  
**Client:** Embedded fitness  
**Project Number:** 1.0

**Auteur:** Sam Hendriks  
**Date:** 25-5-2020  
**Version:** 1.0

---

---

The undersigned declare their agreement with the content of this Project Plan document

**Client**

***Initial Seen:***

***Initial Seen:***

**Project Manager**

Date: \_\_\_\_\_

Place: \_\_\_\_\_

Date: \_\_\_\_\_

Place: \_\_\_\_\_

## Inleiding

In dit document gaan we terug kijken naar hoe het project is verlopen. We gaan ook kijken naar hoogtepunten en verbeterpunten in elk onderdeel van het project. Verder gaan we ook kijken naar hoe het product zelf er nu uit ziet en of dit is hoe wij het voor ogen hadden.

## Het product

Het uiteindelijke product is precies wat wij in gedachten hadden. Het is een functioneel spel wat voldoet aan de specificaties die wij hebben bedacht bij het begin. Het spel geeft eerst een start signaal en daarna begint het spel zelf. Als de speler in de groene vakken staat is hij/zij veilig. Zodra de speler op een rode tegel staat of van de vloer afgaat word komt het eindsignaal van het spel. Ook als de speler op de resetknop staat van het spel speelt het spel zich opnieuw af. Het spel zelf blijft moeilijker worden bij elke ronden en er komen minder groene tegels elke ronden. Daarnaast word de speler op de vloer zelf bijgehouden waar die zich bevind, dit word aangegeven met een gele lijn aan de binnenkant van een tegel. Overall is het spel een succes op zichzelf.

## Vorbereiding

Bij de voorbereiding liep alles vlot op het begin er werd hard gewerkt en documenten stonden klaar. Bij de voorbereiding hebben we een pitch gehad met de klant over wat zij zouden willen hebben. Hierop hebben wij input gegeven en zijn wij op deze basis het idee verder gaan uitwerken

- Wat ging er goed
  - Er werd meteen hard gewerkt
  - Goed communicatie
- Wat ging er minder goed
  - Opgdracht was bij het begin niet heel duidelijk

## Documentatie

Tijdens het documenteren zijn er een aantal problemen omhoog gekomen. In de periode dat hieraan gewerkt werd kwamen er persoonlijke problemen met de teamleider die hierdoor niet goed geconcentreerd verder kon werken. Dit heeft het project een achterstand opgelopen. Voor de rest van het documenteren waren de documenten goed ingevuld en is hier feedback op gegeven. De feedback werd daarna gebruikt op het document en werd hierop aangepast.

- Wat ging er goed
  - Feedback op de documenten werd gebruikt
- Wat ging er minder goed
  - Achterstand door persoonlijke redenen

## Realisatie

Het realiseren van het project ging bij het begin erg vlot, in deze periode werd er goed gewerkt aan het project. Omdat de manier van werken en de code die gebruikt werd redelijk nieuw waren duurde het iets langer om het project goed op tempo te houden.

- Wat ging er goed
  - Goede communicatie
  - Onderdelen goed afgerond
- Wat ging er minder goed
  - Het tempo was traag

## Oplevering

De oplevering zelf was goed verlopen de uitleg die gegeven was is duidelijk geleverd. Daarnaast is er goed vertoont wat de situatie was in bepaalde momenten. Verder zijn de documenten goed op orde gebracht.

- Wat ging er goed
  - Duidelijk overgebracht
  - Goede uitleg
- Wat ging er minder goed
  - Er kon wat meer informatie toegepast worden

## Conclusie

Het project zelf is geslaagd, wat er uiteindelijk gemaakt moest worden is afgerond en volgens de eisen die zijn opgesteld. De documentatie kon wat sneller gemaakt worden en de concentratie erbij houden is een aandachtspunt.

## KwaliteitsHandboek verslag Embedded fitness

---

**Project:** De vloer is lava  
**Client:** Embedded fitness  
**Project Number:** 1.0

**Auteur:** Sam Hendriks  
**Date:** 25-5-2020  
**Version:** 1.0

---

---

The undersigned declare their agreement with the content of this Project Plan document

**Client**

***Initial Seen:***

***Initial Seen:***

**Project Manager**

Date: \_\_\_\_\_

Place: \_\_\_\_\_

Date: \_\_\_\_\_

Place: \_\_\_\_\_

## Contents

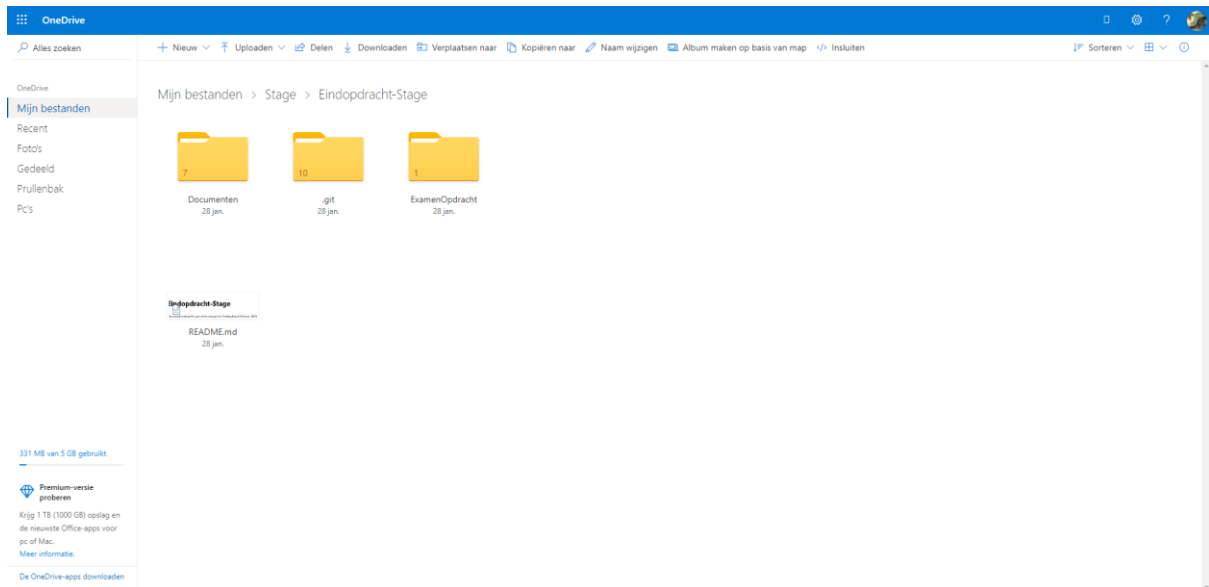
Inleiding .....	64
Digitale opslagplaats.....	64
Inlog gegevens.....	65
Versiebeheer .....	65
Source code.....	66
Aanpassingen/Wijzigingen.....	74
Specificaties .....	74
Kennis .....	74
Gebruikte hardware .....	74
Gebruikte software.....	74

## Inleiding

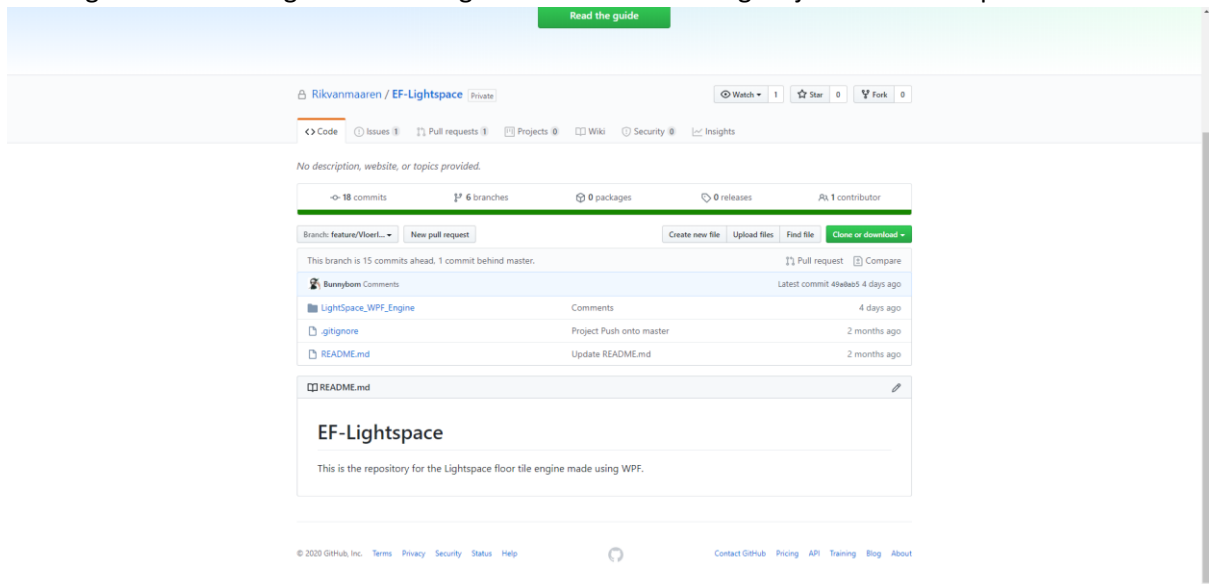
In het kwaliteitshandboek staan alle details waar de documenten en de sourcecode zijn opgeslagen. Daarnaast staat hier ook in welke specificaties bepaalde onderdelen nodig hebben om goed te functioneren. De source code zelf is geschreven in visual studio in C#.

## Digitale opslagplaats

De sourcecode en de documenten zijn opgeslagen op een onedrive. Hierdoor blijft in het geval van een hardware probleem de code veilig opgeslagen.



Ook staat de sourcecode nog een keer opgeslagen op github. Dit geeft extra zekerheid voor als in het kleine geval er iets zou gebeuren met git of de onedrive er nog altijd een ander copie is.



Figuur 7 Onedrive opslagplaats

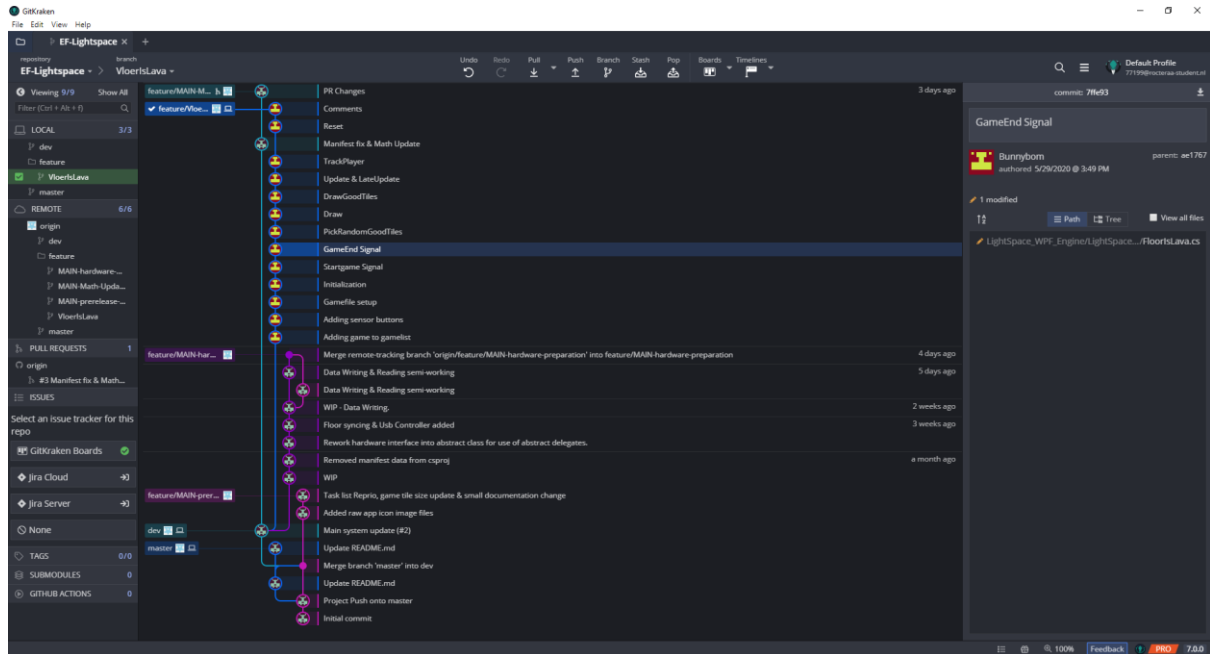


## Inlog gegevens

- Onedrive
  - [77199@roc-teraa.nl](mailto:77199@roc-teraa.nl)
  - 7Tfzm75M
- Github
  - [77199@roc-teraa.nl](mailto:77199@roc-teraa.nl)
  - 7Tfzm75M

## Versiebeheer

Voor de versiebeheer heb ik gitkraken gebruikt om bij te houden op welke versie ik zit.



## Source code

Hierbij komen een aantal screenshots van de sourcecode, onder iedere screenshot staat welk onderdeel van de code het bij hoort.

```
public class FloorIsLava : RunningGameBehavior
{
    private Bitmap backgroundImage;
    private int lightAmount;
    private int sensorAmount;
    private int gameFieldWidth;
    private int gameFieldHeight;
    private List<Tile> GoodTiles = new List<Tile>();
    private List<Tile> resetTile = new List<Tile>();
    private float counter = 0;
    private int counterStart = 0;
    private int counterEnd = 0;
    private int colorNumber = 0;
    private double timeProgression = 10;
    private int goodTilesAmount = 10;
    private bool firstTime = true;
    private bool startGame = true;
    private bool setReset = false;
    private bool gameOver = false;
    private bool playerInGoodTile = false;
```

Figuur 8 variables

```
// Initialized the game, gets the floor size and gets the name and game description
public override void Start()
{
    // Initialized the ticks used in game, gets the Game details and description, gets the floorsize
    base.Start();
    Game.Get.CoreLoop.TicksPerSecond = 10;
    GameName = GameName.FloorIsLava;
    Description = GameList.GetGameDescription(GameName);
    var tiles = Game.Get.TileManager.Tiles;
    GameFieldTileSize = new Vector2(tiles.GetLength(0), tiles.GetLength(1));

    // Gets the lightAmount and the sensorAmount from the tileManager
    lightAmount = Game.Get.TileManager.GetLightAmount();
    sensorAmount = Game.Get.TileManager.GetSensorAmount();

    // Calculates the gameFieldWidth and height
    gameFieldWidth = GameFieldTileSize.X * lightAmount;
    gameFieldHeight = GameFieldTileSize.Y * lightAmount;

    resetTile.Add(tiles[0, 0]);

    // Calculates the amount of goodTiles used.
    goodTilesAmount = GameFieldTileSize.X * GameFieldTileSize.Y / 2;

    // Gets the background and draws it black
    backgroundImage = new Bitmap(gameFieldWidth, gameFieldHeight);
    backgroundImage.DrawRectangle(Vector2.Zero(), gameFieldWidth, gameFieldHeight, true, 2, Colors.Black());
    Game.Get.TileManager.SetRenderGraphic(backgroundImage);
}
```

Figuur 9 Initialization

```
// The first update, used to select the goodTiles before being drawn
public override void Update()
{
    if (!startGame && !gameOver)
    {
        counter += Time.deltaTime;

        // After a specified time the goodTiles are selected
        if (counter > timeProgression || firstTime)
        {
            PickRandomTiles();
            counter = 0;
            firstTime = false;

            // Decreases the time in between rounds, every round
            if (timeProgression > 3)
            {
                timeProgression = timeProgression - (timeProgression * 0.05);
            }
        }
    }
}
```

Figuur 10 Update

```
// Is the second update, the main part of the game is handled here.
public override void LateUpdate()
{
    var tempBG = backgroundImage.Clone() as Bitmap;

    // Performs the StartGame method multiple times
    if (startGame && !gameOver)
    {
        counter += Time.deltaTime;
        if (counter > 1)
        {
            counter = 0;
            StartGame(tempBG);
        }
    }

    // Performs the GameOver method multiple times
    else if (gameOver && !setReset)
    {
        counter += Time.deltaTime;

        if (counter > 0.1)
        {
            counter = 0;
            if (counterEnd < 16)
            {
                GameOver(tempBG);
            }
            else
            {
                setReset = true;
            }
        }
    }
}
```

Figuur 11 Lateupdate part 1

```
// This if draws the goodTiles and the red background, it also tracks the player on the floor
else if (!setReset)
{
    counter += Time.DeltaTime;
    // After a period of time this if statement will be executed
    if (counter > timeProgression - 1)
    {
        var allTiles = Game.Get.TileManager.Tiles;
        playerInGoodTile = false;

        // Draws all tiles Red on the Background
        foreach (var tile in allTiles)
        {
            var pos = new Vector2(tile.Position.X * lightAmount, tile.Position.Y * lightAmount);

            tempBG.DrawRectangle(pos, lightAmount - 1, lightAmount - 1, true, 0, Colors.Red());
        }

        // Checks every goodTile if player is on it
        foreach (var tile in GoodTiles)
        {
            if (tile.AnyActiveSensorsInTile())
            {
                playerInGoodTile = true;
            }
        }

        // When Player isnt on a goodTile the game is over
        if (!playerInGoodTile)
        {
            gameOver = true;
        }
    }
}
```

Figuur 12 Lateupdate part 2

```
// When the game is over a there will be a yellow tile that marks the reset button
else
{
    var pos = new Vector2(0 * lightAmount, 0 * lightAmount);
    tempBG.DrawRectangle(pos, lightAmount - 1, lightAmount - 1, true, 0, Colors.Yellow());
    foreach (var tile in resetTile)
    {
        if (tile.AnyActiveSensorsInTile())
        {
            Reset();
        }
    }
}

Draw(tempBG);
}
```

Figuur 13 Lateupdate part 3

```
// Draws all the goodtiles on the background
private void DrawGoodTiles(Bitmap tempBG)
{
    if (GoodTiles.Count > 0)
    {
        foreach (var tile in GoodTiles)
        {
            var pos = new Vector2(tile.Position.X * lightAmount, tile.Position.Y * lightAmount);
            var color = Colors.Green();

            // Draw them at height&width = light amount. first x&y position is automatically added.
            tempBG.DrawRectangle(pos, lightAmount - 1, lightAmount - 1, true, 0, color);
        }
    }
}
```

Figuur 14 Draw green tiles

```
// Draws the background on the floor and shows this in the program
public void Draw(Bitmap bmp)
{
    Game.Get.TileManager.SetRenderGraphic(bmp);
}

// Check collision within tile, if collision found draw outline of the tile yellow
public void TrackPlayer(Bitmap tempBG)
{
    // Gets all tiles on the floor and stores them in allTiles
    var allTiles = Game.Get.TileManager.Tiles;

    // Goes through all the tiles in allTiles
    foreach (var tile in allTiles)
    {
        // If a sensor is activated in the tile it will outline the tile yellow
        if (tile.AnyActiveSensorsInTile())
        {
            var pos = new Vector2(tile.Position.X * lightAmount, tile.Position.Y * lightAmount);

            tempBG.DrawRectangle(pos, lightAmount - 1, lightAmount - 1, false, 0, Colors.Yellow());
        }
    }
}
```

Figuur 15 Draw and Trackplayer

```
// Gets random tiles where the player can stand on
public void PickRandomTiles()
{
    GoodTiles = new List<Tile>();
    var tiles = Game.Get.TileManager.Tiles;
    var rng = new Random();

    // Picks goodTilesAmount goodTiles where the player can stand on
    for (var index = 0; index < goodTilesAmount; index++)
    {
        GoodTiles.Add(tiles[
            rng.Next(0, tiles.GetLength(0)),
            rng.Next(0, tiles.GetLength(1))
        ]);
    }

    // Every round there is one less goodTile untill it gets to 1
    if (goodTilesAmount > 1)
    {
        goodTilesAmount--;
    }
}
```

Figuur 16 Pick random tiles

```
// Gives the start signal of the game.
public void StartGame(Bitmap tempBG)
{
    var allTiles = Game.Get.TileManager.Tiles;
    var color = Colors.Red();

    // Goes through 4 different colors to make the startsignal
    switch (counterStart)
    {
        case 0:
            color = Colors.Red();
            break;
        case 1:
            color = Colors.Black();
            break;
        case 2:
            color = Colors.Orange();
            break;
        case 3:
            color = Colors.Black();
            break;
        case 4:
            color = Colors.Green();
            break;
        case 5:
            color = Colors.Black();
            break;
        default:
            break;
    }
}
```

Figuur 17 Startgame part 1

```
// Draws all tiles one color on the background
foreach (var tile in allTiles)
{
    var pos = new Vector2(tile.Position.X * lightAmount, tile.Position.Y * lightAmount);

    tempBG.DrawRectangle(pos, lightAmount - 1, lightAmount - 1, true, 0, color);
}

if (counterStart > 4)
{
    startGame = false;
}

counterStart++;
}
```

Figuur 18 Startgame part 2



```
// Gives the end signal of the game
public void GameOver(Bitmap tempBG)
{
    var allTiles = Game.Get.TileManager.Tiles;
    var color = Colors.Red();
    startGame = true;

    // Sets the color of the floor
    switch (colorNumber)
    {
        case 0:
            color = Colors.Red();
            colorNumber = 1;
            break;
        case 1:
            color = Colors.Black();
            colorNumber = 0;
            break;
    }

    //Every tile in allTiles is being drawn on de tempBG with the assigned color
    foreach (var tile in allTiles)
    {
        var pos = new Vector2(tile.Position.X * lightAmount, tile.Position.Y * lightAmount);

        tempBG.DrawRectangle(pos, lightAmount - 1, lightAmount - 1, true, 0, color);
    }

    counterEnd++;
}
```

Figuur 19 Gameover

```
// Resets the game
public void Reset()
{
    firstTime = true;
    startGame = true;
    setReset = false;
    gameOver = false;
    playerInGoodTile = false;

    counterStart = 0;
    counterEnd = 0;
    colorNumber = 0;

    goodTilesAmount = GameFieldTileSize.X * GameFieldTileSize.Y / 2;
    timeProgression = 10;
}
```

Figuur 20 Reset

## Aanpassingen/Wijzigingen

Alle aanpassingen en wijzigingen zijn gedocumenteert in de CRPR lijst.

#	type	versie	status	prioriteit	datum	omschrijving	indelaar	opmerkingen
1	1	1	Done	Must	12May2020	Tovoege van een speler tracker aan het spel.	Shen	Tongevogt aan het spel.
2	1	1	Done	Must		Feedback programma van eisen: hierbij waren er opmerkingen over mijn zin opbouw en wat spellings fouten.	Shen	Document bijgewerkt
3	1	1	Done	Must		Feedback projectplan: het grafiek om de planning aan te geven was te eenvoudig en toch verwierend. Om dit op te lossen heb het wls bestand deels toegelicht waarin detail staat beschreven wat er gaat gebeuren in welke week. Verder waren er ook opmerkingen over de zin opbouw en spelling.	Shen	Document bijgewerkt
4	1	1	Done	Must		Feedback technisch ontwerp: het klassediagram was niet echt een klassediagram. Hierover heb ik uitleg gekregen en heb ik aangepast zodat het wel een klassediagram is. Verder moest er beter uitgelegd worden hoe de programma's met elkaar communiceren.	Shen	Document bijgewerkt
5	1	1	Done	Must		Feedback functioneel ontwerp: Sommige functies waren niet duidelijk beschreven en moesten hierdoor worden aangepast. Zoals het beschrijven van wat er gebeurt als iemand van de vloer afstapt wat het spel dan doet.	Shen	Document bijgewerkt
6	1	1	Done	Must		Probleem met de resetknop: het spel kon de resetTegel niet apart als 1 tegel opstaan en crashte zodra deze regel werd uitgevoerd. Om dit op te lossen heb ik de resetTegel in een eigen tegel array gezet om zo aangesproken te worden.	Shen	Code aangepast
7	1	1	Done	Must		Het startsignaal speelde niet af bij het opstarten van het spel ging het startsignaal niet af. Dit kwam vanwege een te kleine delay tussen de signalen en daarom werd dit niet uitgevoerd. Dit is opgelost door de delay iets groter te maken.	Shen	Code aangepast
8	1	1	Done	Must		Het tracken van de speler ging niet als gewenst: bij het bijhouden van de locatie van de speler werd de gele lijn niet getekent die aangeeft waar de persoon is. Het probleem lag erbij dat de lijn die getekent moest worden overlapt werd door een volledige kleur tegel. Om dit op te lossen word het tekenen van de locatie van de speler als laatst gedaan zodat deze altijd bovenop komt te liggen.	Shen	Code aangepast
9	1	1	Done	Must		Het automatisch uitrekenen van hoeveel groene tegels een spel begint: bij het begin van het spel word er aan de hand van de hoeveelheid tegels uitgereken hoeveel groene tegels er in totaal komen. De berekening was niet correct en werd hierdoor veel te groot. uiteindelijk was heel het veld groen voor een lange tijd. Om dit op te lossen heb ik de calculatie aangepast met door de correcte nummers te gebruiken. eerst werd de uitkomst van floorwidth * richtamount gebruikt als nummer. nu word alleen floorwidth gebruikt.	Shen	Code aangepast

## Specificaties

### Kennis

Om dit project te maken en te begrijpen is er een redelijke kennis nodig van C# en hoe je moet omgaan met visual studio.

### Gebruikte hardware

- Laptop
  - MSI gpt26qf leopard
- Interactieve vloer

### Gebruikte software

- Visual studio
  - Visual studio .Net framework 4.8
  - C# XAML capable IDE(visual studio / VSCode recommended)
- Aansturings programma.
- Windows 10 pro
- Microsoft office word, excel, en powerpoint
- Trello
- Github
- Gitkraken
- Onedrive

Logboek kandidaat  
Proeve van Bekwaamheid

Persoonsinformatie	
Naam kandidaat	Sam Hendriks
Studentnummer	77199
Klas/groep	IC17AO.A
(Leer)bedrijf	Embedded fitness
Schoolbeoordelaar	
Praktijkbeoordelaar	Joël van Huijkelom



Datum	Werkzaamheden en/of examenopdracht	Resultaat	Werkproces
13-4-2020	Brainstorm sessie houden met collega's	Een aantal ideeën voor het project waar ik aan ga werken	B1-K1-W1
13-4-2020	Concept uitschrijven	Een concept geschreven over een van de ideeën die ik heb	B1-K1-W1
13-4-2020	Pitch voorbereiden	Het maken van een presentatie om een pitch te geven aan de opdrachtgever	B1-K1-W1
14-4-2020	Pitch presenteren aan de opdrachtgever	Pitch gegeven aan de opdrachtgever. Daarbij gevraagd voor eventuele toevoegingen aan het project en goedkeuring	B1-K1-W1
14-4-2020	Pitch analyseren en vastleggen	De pitch analyseren en vastleggen voor het programma van eisen	B1-K1-W1
15-4-2020	Beginnen aan het maken van het programma van eisen	Opzetten document	B1-K1-W1
15-4-2020	Brainstorm sessie en het concept samen met de pitch analyse in het programma van eisen documenteren	Het documenteren van de brainstorm sessie, concept en de pitch analyse	B1-K1-W1
16-4-2020	Verzamelen van informatie doormiddel van het zoeken van documenten of oude software. Daarnaast ook gekeken naar nog bestaande software en vragen gesteld bij de opdrachtgever	Weinig informatie kunnen vinden over het project. Wat ik kon vinden is gedocumenteert in het programma van eisen	B1-K1-W1
16-4-2020	De afspraken, eindresultaat en impact gedocumenteert in het programma van eisen	De afspraken, eindresultaat en impact gedocumenteert in het programma van eisen	B1-K1-W1
17-4-2020	Het laten goedkeuren van het programma van eisen	Een paar kleine aanpassingen maken. Daarna was het geaccepteert	B1-K1-W1

17-4-2020	Opzetten van het projectplan	Het aanmaken van het projectplan document	B1-K1-W2
17-4-2020	Het documenteren van het projectdoel, projectactiviteiten, projectgrenzen, en de uiteindelijke producten	Een deel van het projectplan is afgerond	B1-K1-W2
20-4-2020	Het maken van een WBS en een trello pagina	Wbs is gemaakt samen met een trello pagina	B1-K1-W2
20-4-2020	Het maken van een planning doormiddel van de WBS en trello	Planning gemaakt voor het gehele project	B1-K1-W2
20-4-2020	De planning gedocumenteert in het projectplan samen met de risico's in het project.	Project plan afgerond	B1-K1-W2
21-4-2020	Projectplan opleveren	De zinsopbouw en planning moeten worden bijgewerkt	B1-K1-W2
21-4-2020	Aanpassingen maken aan het projectplan	Projectplan goedgekeurd	B1-K1-W2
21-4-2020	Het aanmaken van het functioneel ontwerp	Functioneel ontwerp document aangemaakt	B1-K1-W3
21-4-2020	Project management, project detail, en project omschrijving gedocumenteert.	Een deel van het functioneel ontwerp afgerond	B1-K1-W3
22-4-2020	Requirements opzetten in het functioneel ontwerp	Requirements gedocumenteert	B1-K1-W3
22-4-2020	Het maken van een user case diagram, een dataflow diagram, en een aantal wireframes	Diagrammen en wireframes afgerond	B1-K1-W3
23-4-2020	Het laten beoordelen van het functioneel ontwerp	Ik moest meer in detail gaan over de functies en hoe de applicatie werkte	B1-K1-W3
23 -4-2020	Aanpassingen gemaakt aan het functioneel ontwerp. Ik heb hierbij bij de projectomschrijving meer in detail gegaan hoe	Functioneel ontwerp is goedgekeurd	B1-K1-W3

	de applicatie werkt en wat het uitvoert. Ook worden de functies beter beschreven.		
24-4-2020	Het aanmaken van het technisch ontwerp	Technisch ontwerp document opgezet	B1-K1-W3
27-4-2020	Functies omzetten naar technische eisen in het technisch ontwerp	Technische eisen verwerkt in het technisch ontwerp	B1-K1-W3
27-4-2020	Gekeken naar alle componenten en modules die gebruikt worden en die gedocumenteert in het technisch ontwerp	Alle componenten en modules gedocumenteert	B1-K1-W3
27-4-2020	Gekeken hoe de componenten en modules met elkaar samenwerken	Gedocumenteert hoe alle componenten en modules met elkaar samenwerken	B1-K1-W3
28-4-2020	Het maken van een klassendiagram en een activiteiten diagram	Een klassen en activiteiten diagram gedocumenteert in het technisch ontwerp	B1-K1-W3
28-4-2020	Het beschrijven van het relationele datamodel, en ter verduidelijking een dataflowdiagram toegevoegd	Een dataflow diagram en een relationeel datamodel gedocumenteert in het technisch ontwerp	B1-K1-W3
29-4-2020	De beslissingen, afspraken, grenzen, het plan van aanpak, en het test plan gedocumenteert	De beslissingen, afspraken, grenzen, het plan van aanpak, en het test plan gedocumenteert	B1-K1-W3
29-4-2020	Het opleveren van het technisch ontwerp	Het klassendiagram was niet echt een klassendiagram. De zinsopbouw was onvoldoende	B1-K1-W3
29-4-2020	Gevraagd voor uitleg over een klassendiagram	Uitleg gekregen over wat het precies inhoud en hoe ik het kan toepassen op mijn project	B1-K1-W3

30-4-2020	Klassendiagram gemaakt en laten controleren	Klassendiagram goedgekeurd	B1-K1-W3
30-4-2020	Technisch ontwerp aangepast door de verbeterde klassendiagram toe te voegen en mijn zinsopbouw te verbeteren	Technisch ontwerp bijgewerkt	B1-K1-W3
30-4-2020	Technisch ontwerp opgeleverd	Technisch ontwerp goedgekeurd	B1-K1-W3
1-5-2020	Opzetten voorbereiding realisatie	Het voorbereidings realisatie document opgezet	B1-K1-W4
1-5-2020	Benodigheden lijst maken. Hierin staan allen benodigheden in die nodig zijn voor het project	Benodigheden lijst gedocumenteert in het voorbereidings document	B1-K1-W4
1-5-2020	Werkomgeving gedocumenteert	Werkomgeving gedocumenteert in het voorbereidings document	B1-K1-W4
1-5-2020	Het voorbereidings document laten controleren	Document goedgekeurd	B1-K1-W4
4-5-2020	Opzetten werkomgeving, dit heb ik gedaan door een lijst met instellingen te volgen van het aansturings programma.	Functionele werkomgeving opgezet	B1-K1-W4
4-5-2020	Testgame folder opgezet voor het testen van de functies	Game folder opgezet	B1-K1-W4
4-5-2020	Functies uitproberen die in een ander spel staan	Functies werken in de test folder	B1-K1-W4
4-5-2020	De vloer is lava folder opgezet in het aansturings programma	De vloer is lava folder is aangemaakt	B1-K2-W1
5-5-2020	CRPR lijst word aangemaakt	Een CRPR lijst aangemaakt voor veranderingen of problemen	P1-K1-W1
5-5-2020	Hulp gevraagd bij de programmeur van het aansturings programma hoe ik het best kan beginnen	Informatie gekregen over het programma zelf	B1-K2-W1



6-5-2020	Begonnen met het koppelen van het spel aan het aansturings programma. hiervoor heb ik de game op meerdere plekken in het aansturings programma moeten definiëren.	Spel gekoppelt aan het aansturings programma	B1-K2-W1
7-5-2020	Inializatie toevoegen om het aantal tegels te berekenen en de vloer standaard op zwart te zetten	Spelgrootte word automatisch berekent bij het opstarten van het spel.	B1-K2-W1
7-5-2020	Haal alle sensoren en ledjes automatisch op voor iedere tegel	Bij het opstarten van het spel word er berekent hoeveel ledjes en sensoren er zijn per tegel	B1-K2-W1
8-5-2020	Automatische update toevoegen die constant om de zoveel tijd word uitgevoerd	Een loop functie toegevoegd die 10 ticks heeft tussen elke loop	B1-K2-W1
8-5-2020	Het kleur geven aan een tegel	Ging nog niet volgens plan, de ledjes gingen niet aan	B1-K2-W1
11-5-2020	een clone van de achtergrond maken om de tegels van kleur te kunnen laten veranderen	De clone word gemaakt de tegel veranderd niet van kleur	B1-K2-W1
11-5-2020	Een draw functie gemaakt die de clone background uitvoert op de vloer	De draw functie word uitgevoerd alleen de vloer blijft zwart	B1-K2-W1
12-5-2020	Via een functie op de clone een tekening uitvoeren die de kleur veranderd de functie heeft wel wat variabelen nodig	Zodra de functie uitgevoerd is en daarna de draw functie veranderd de kleur van de uitgekozen positie	B1-K2-W1
13-5-2020	Een tegel array aanmaken voor alle groenen tegels	Array aangemaakt	B1-K2-W1
13-5-2020	Een random positie selector voor alle groene tegels	Dit was met een random generator goed te doen	B1-K2-W1
14-5-2020	De random geselecteerde tegels laten tekenen en vertonen op de vloer dit ga ik doen aan de hand van een foreach loop	Bij elke tegel word er op de clone achtergrond een groene tegel getekent zodra deze clone op de	B1-K2-W1

		draw functie komt word alles weergeven op de vloer	
14-5-2020	Lateupdate gemaakt deze update loop komt na de standaard update	De lateupdate word uitgevoerd na de update	B1-K2-W1
15-5-2020	De functies van het tekenen verplaatst naar de lateupdate	Werkt nog hetzelfde	B1-K2-W1
15-5-2020	Een startsignaal toegevoegd aan het spel het start signaal word eerst rood daarna oranje en dan groen	Elke keer als het spel start word het startsignaal gegeven	B1-K2-W1
18-5-2020	Een eindsignaal toegevoegd aan het spel, zodra de speler af is knippert de vloer rood	Zodra er een verliesconditie is gegeven knippert de vloer rood	B1-K2-W1
18-5-2020	De overige tegels rood maken een seconden voordat de goede tegels weer veranderen	De tegels die nog geen kleur hadden gekregen worden rood voordat de groene tegels weer van positie verplaatsen	B1-K2-W1
19-5-2020	Verlies condities gemaakt aan het spel	Als een speler niet op een groene tegel staat als de vloer rood word is het spel afgelopen	B1-K2-W1
20-5-2020	Speler zijn positie uitlezen en deze plaats markeren	Alle tegels worden gecontroleert of hier een sensor actief is indien dat zo is word de rand van deze tegel geel	B1-K2-W1
21-5-2020	Een reset tegel maken voor wanneer de speler heeft verloren	Het selecteren van een tegel leid het spel te laten crashen	B1-K2-W1
22-5-2020	De reset tegel verwerken door een tegel in een array te zetten en deze te controleren	Zodra iemand op de reset tegel staat wanneer het spel is afgelopen begint het opnieuw	B1-K2-W1
22-5-2020	Testen van de applicatie op fouten	Tijdens de eerste test kwamen er wat problemen met het	B1-K2-W2

		startsignaal en de reset tegel	
25-5-2020	Het oplossen van de problemen uit de eerste test de reset tegel kwam soms niet tevoorschijn dit probeer ik op te lossen door de functie op een andere locatie in de code te zetten en het startsignaal heb ik de tijd tussen kleuren iets langer gemaakt	De oplossingen hebben zover gewerkt	B1-K2-W2
25-5-2020	Testen van de applicatie op fouten	Alle onderdelen die getest zijn werken zoals behoren	B1-K2-W2
26-5-2020	Acceptatie test geschreven hierin heb ik gezet waaraan de applicatie moet voldoen om te slagen hierbij kunnen testers schrijven wat zij hier liever hebben of andere opmerkingen	Acceptatie test geschreven	B1-K3-W1
26-5-2020	Met 1 collega door de acceptatie test heen gegaan	Alle functies waren voldoende verder waren er kleine opmerkingen	B1-K3-W1
27-5-2020	Met 2 collega's de acceptatie test doorgenomen	Alle functies waren voldoende verder waren er kleine opmerkingen	B1-K3-W1
28-5-2020	Alle documenten bekeken op juistheid en volledigheid, ik heb wat aanpassingen gemaakt aan het programma van eisen en het technisch ontwerp	Documentatie bijgewerkt	P1-K1-W1
29-5-2020	Product opgeleverd bij de opdrachtgever	Product is opgeleverd	B1-K3-W2
29-5-2020	Het evalueren van het product en het werkproces, hierbij hebben ik en de opdrachtgever gekeken hoe het product is afgerond en hoe het werkproces is verlopen. Dit heb ik daarna gedocumenteert in het evaluatie verslag	Evaluatieverslag gemaakt en in gedocumenteerd	B1-K3-W2

Handtekening voor gezien			
Praktijkbeoordelaar		Datum	
Kandidaat		Datum	

## Bijlagen

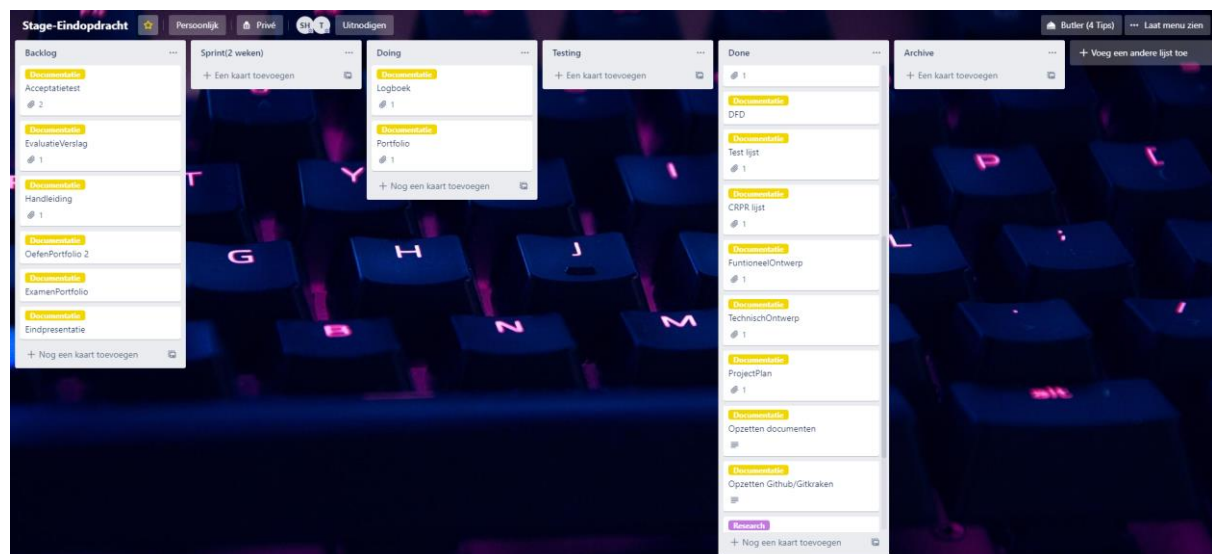
### CRPR lijst

nr	type	versie	status	prioriteit	datum	omschrijving	indelen	opmerkingen
1	1	1	Done	Must	12May2020	Toevoegen van een speler tracker aan het spel.	Shen	Tegenvoert aan het spel.
2	1	1	Done	Must		Feedback programma van eisen; hierbij waren er opmerkingen over mijn zin opbouw en wat spellings fouten.	Shen	Document bijgewerkt
3	1	1	Done	Must		Feedback projectplan; het grafiek om de planning aan te geven was te eenvoudig en toch verwarrend. Om dit op te lossen heb het wls bestand iets toegevoegd waarin detail staat beschreven wat er gaat gebeuren in welke week. Verder waren er ook opmerkingen over de zin opbouw en spelling.	Shen	Document bijgewerkt
4	1	1	Done	Must		Feedback technisch ontwerp; het klassediagram was niet echt een klassediagram. Hierover heb ik uitleg gegeven en heb ik aangepast zodat het wel een klassediagram is. Verder moest er beter uitgelegd worden hoe de programma's met elkaar communiceren.	Shen	Document bijgewerkt
5	1	1	Done	Must		Feedback functioneel ontwerp. Sommige functies waren niet duidelijk beschreven en moesten hierdoor worden aangepast. Zoals het beschrijven van wat er gebeurt als iemand van de vloer afstapt wat het spel dan doet.	Shen	Document bijgewerkt
6	1	1	Done	Must		Probleem met de resetknop; het spel kan de resetTegel niet apart als 1 tegel opstaan en crashde zodra deze regel werd uitgevoerd. Om dit op te lossen heb ik de resetTegel in een eigen tegel array gezet om zo aangesproken te worden.	Shen	Code aangepast
7	1	1	Done	Must		Het startsignaal speelde niet af; bij het opstarten van het spel ging het startsignaal niet af. Dit kwam vanwege een te kleine delay tussen de signalen en daarom werd dit niet uitgevoerd. Dit is opgelost door de delay iets groter te maken.	Shen	Code aangepast
8	1	1	Done	Must		Het tracken van de speler ging niet als gewenst; bij het bijhouden van de locatie van de speler werd de gele lijn niet getekend die aangeeft waar de persoon is. Het probleem lag erbij dat de lijn die getekent moest worden overrupt werd door een volledige kleur tegen. Om dit op te lossen word het tekenen van de locatie van de speler als laatste gedaan zodat deze altijd bovenop komt te liggen.	Shen	Code aangepast
9	1	1	Done	Must		Het automatisch uittrekken van hoeveel groene tegels een spel begint; bij het begin van het spel word er aan de hand van de hoeveelheid tegels uitgetreken hoeveel groene tegels er in totaal komen. De berekening was niet correct en werd hierdoor veel te groot. Uiteindelijk was het wel goed groen voor een lange tijd. Om dit op te lossen heb ik de calculatie aangepast met door de correcte nummers te gebruiken, eerst werd de uitkomst van floor(n/10) * lichtamount gebruikt als nummer, nu word alleen floor(n/10) gebruikt.	Shen	Code aangepast

### WBS

1	Projectplan	M	Shen	3	3	Onedrive/Github	
2	Pitch presentatie	M	Shen	1	2		
3	WBS	M	Shen	2	2		
4	Technisch ontwerp	M	Shen	3	3		
5	Functioneel ontwerp	M	Shen	3	3		
6	Programma van eisen	M	Shen	3	3		
7	CRPR lijst	M	Shen	1	1		
8	Test lijst	M	Shen	2	2		
9	DFD	M	Shen	2	2		
10	Architectuur document	M	Shen	3	3		
11	Design document	M	Shen	3	3		
12	Acceptatietest	M	Shen	2	2		
13	EvaluatieVerslag	M	Shen	1	1		
14	Handleiding	M	Shen	2	2		
15	Kerntaak 1	M	Shen	5	5		
16	Kerntaak 2	M	Shen	5	5		
17	Kerntaak 3	M	Shen	5	5		
18	Kerntaak 4	M	Shen	5	5		
19	OefenPortfolio 1	M	Shen	20	20		
20	OefenPortfolio 2	M	Shen	25	25		
21	ExamenPortfolio	M	Shen	40	40		
22	Eindpresentatie	M	Shen	2	2		
23	Sensoren uitlezen	M	Shen	10	10	Als de sensoren niet werken kan je de positie niet uitlezen waar iemand staat.	
24	Ledjes aansturen	M	Shen	6	6	Als de ledjes niet werken kan er niet worden aangegeven waar iemand kan staan en waar niet.	
25	Win/Lose condities	M	Shen	10	10	Zonder dit zou er geen competitie zijn.	
26	LoseEffect	M	Shen	20	20	Zonder dit zou je niet weten wanneer je verloren hebt.	
27	Randomizer	M	Shen	20	20	Zonder een randomizer kan je blijven staan of kan je een patroon lezen.	
28	Versneller	M	Shen	14	14	Anders kan je voor eeuwig doorgaan.	
29	Communicatie aansturing programma	M	Shen	10	10	Als dit er niet is kan de vloer niet worden aangestuurd.	
30	Start effect	M	Shen	20	20	Zonder weet de gebruiker niet dat het spel is begonnen.	
30	Moelijkheids keuze	C	Shen	40	40		
	Totaal			288	288		

### Trello



## Dankwoord

Ik wil als laatste nog het stagebedrijf bedanken voor deze kans die ze mij hebben gegeven om aan dit project te kunnen werken. Daarnaast wil ik mijn praktijkbegeleider Joël bedanken voor de tijd en moeite die in mijn stage project is gestoken. Verder wil ik Thomas bedanken voor het verduidelijken van bepaalde zaken rondom het stage en de snelle reactie die ik kreeg op mijn mailtjes. Ik heb veel geleerd tijdens deze opdracht zelf en nog meer tijdens heel mijn stage bij embedded fitness.