

# Game Development

## Kermodule 3 AI

Kevin Absen

### Opdracht 1

Robo Recall analyse

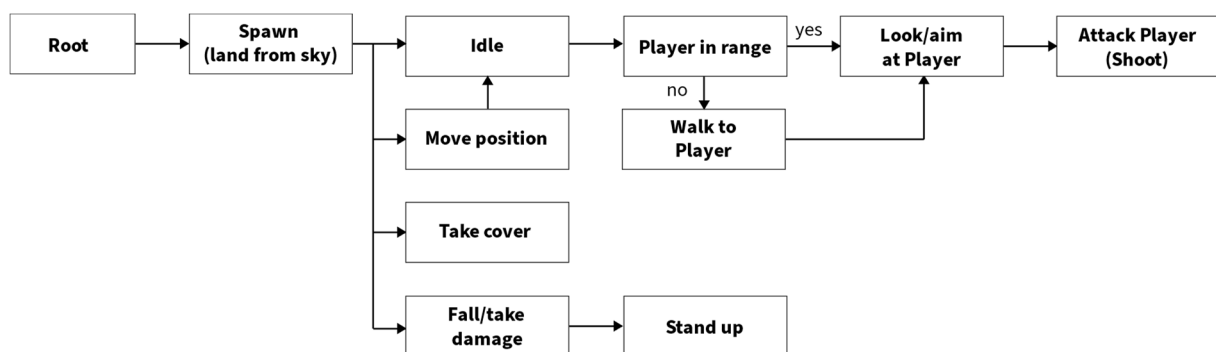


### Functie van AI

In de VR game Robo Recall hebben robots de wereld overgenomen en ze hebben maar één missie, namelijk de speler vernietigen. Als speler is je doel om verschillende levels te clearen door de stad, maar het wordt steeds moeilijker omdat de robots sterker en slimmer worden.

Er kan onderscheidt worden gemaakt tussen verschillende type robots/AI, beginnend bij de meest voorkomende, de human robots. Deze AIs vallen alleen aan vanaf de grond en hebben zwakke wapens. Er zijn ook spiderbots, deze AIs lopen over de grond en 'chargen up' waarna ze vervolgens op je gezicht gaan zitten. Er zijn ook vliegende robots die raketten, kogels of lasers kunnen schieten. Tenslotte is er ook een boss robot, Odin, en hij heeft verschillende states die telkens afwisselen. Voor de analyse heb gekozen om de human robots te analyseren.

### States (human robot):



<i>State</i>	<i>Trigger</i>
<i>Spawn</i>	Wanneer speler checkpoint nadert
<i>Idle</i>	Nadat de enemy spawned is
<i>Move position</i>	Random, nadat de enemy spawned is
<i>Take cover</i>	Als de enemy schade heeft gekregen en weer is opgestaan
<i>Fall over</i>	Als de enemy schade heeft gekregen
<i>Stand up</i>	Nadat de enemy schade is gevallen
<i>Walk to player</i>	Als de speler niet in range is
<i>Look/aim at player</i>	Als de speler in range is
<i>Attack player (shoot)</i>	Als de enemy gericht heeft op de speler

#### **Wat werkt goed:**

Ik vind het interessant dat de AI verschillende patronen heeft. Ze blijven dus niet constant op dezelfde positie staan en elke keer dat je speelt spawnen de AIs steeds op een andere plek. Daarnaast vind ik het ook goed dat er bepaalde tussenstates zijn wanneer de robot bijvoorbeeld schade krijgt. De robot valt om, gaat naar idle voor een paar seconden, en staat vervolgens weer op. Dit maakt de AI naar mijn mening realistischer.

#### **Wat werkt niet goed:**

De AI is soms gefocust op hele andere dingen dan de speler. Dit kan je merken wanneer de robot zich verplaatst, hij kijkt dan voor zich uit in plaats van naar de speler en blijft dan een paar seconden stilstaan voordat hij de speler weer focust. Dit maakt het voor de speler wel extra makkelijk om de AI te doden, maar het geeft ook het gevoel dat de AIs een beetje dom zijn.

#### **Bronnen:**

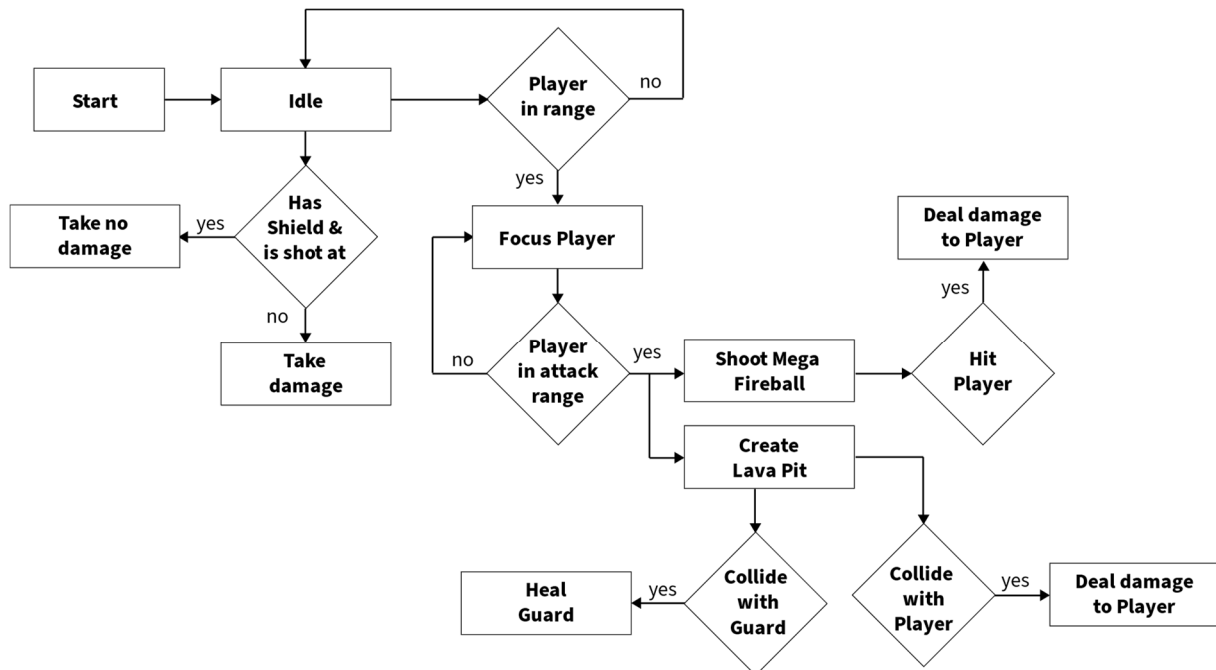
<https://www.youtube.com/watch?v=Tjzcuz7PN5k>  
[https://www.youtube.com/watch?v=t\\_rpcTo1SW8](https://www.youtube.com/watch?v=t_rpcTo1SW8)

## Opdracht 2

AI states/behaviours

### Boss

Flowchart:

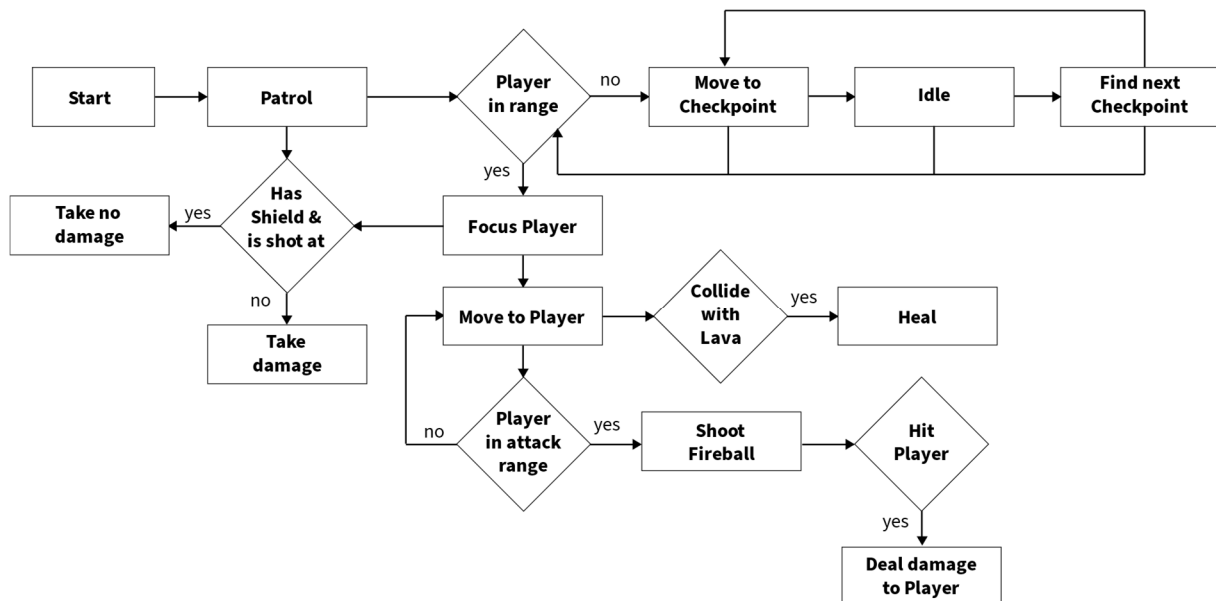


States:

State	Purpose	Interaction Type
<b>Focus Player</b>	The boss will focus the player when he is within the look radius. Rotates towards the players' position.	Player and AI
<b>Attack (Shoot Mega Fireball)</b>	When the player is within the attack radius, the boss will shoot a big fireball towards the players' last position. When hit, the mega fireball deals massive damage to the player.	Player and AI
<b>Create Lava Pit</b>	When the player is within the attack radius, the boss creates a lava pit on a random position within the spawn radius. The lava pit becomes bigger over time and damages the player if he collides with it.	World and AI
<b>Heal Guard</b>	If a guard collides with the lava pit, the boss will heal the guard until it is back to full HP.	AI and AI

## Guard

Flowchart:



States:

State	Purpose	Interaction Type
<b>Patrol</b>	The guard patrols around a certain area with an X amount of checkpoints. The patrol state has 3 sub-states: Move to Checkpoint, Idle and Find Next Checkpoint.	World and AI
<b>Chase Player</b>	When the player is within the look radius, the guard will chase the player until it is out of range. The guard will automatically rotate and move towards the players' position.	Player and AI
<b>Attack (Shoot Fireball)</b>	When the player is within the attack radius, the guard will shoot a fireball towards the players' last position. When hit, the fireball deals significant damage to the player.	Player and AI
<b>Heal</b>	If the guard collides with a lava pit, the lava will heal the guard until it is back to full HP.	AI and AI

## Opdracht 3

### Documentatie en game architecture

#### Gameplay

Deze game is een top-down shooter waarbij de speler vijanden moet verslaan om de overwinning te bereiken. Deze vijanden zijn Guards en de allermachtige Boss AI die het hoogtepunt is van de game. In de game zitten ook magische tempels en deze worden beschermd door de Guards. De tempels bieden zelf ook bescherming voor de vijanden door middel van schilden, die pas verdwijnen wanneer de tempel is vernietigd.

Het is de taak van de speler om de vijanden uit te schakelen. Dit kan hij doen door ijsballen te schieten. Hoe langer de speler de ijsbal oplaad, hoe meer schade deze aanricht. De vijanden bekogelen de speler met vuurballen en de speler moet proberen om zo lang mogelijk te overleven en de Boss te verslaan.

De Boss heeft een krachtig schild dat pas verdwijnt wanneer alle tempels zijn vernietigd. Daarnaast kan de Boss lava pits spawnen die langzaam schade aanrichten aan de speler, maar ook levens toevoegen aan de Guards wanneer ze hiermee in aanraking komen. De speler wordt dus uitgedaagd om met een tactische oplossing te komen om de boss te verslaan.

#### Architecture

De game maakt gebruik van 2 soorten AIs: de *Guard* en de *Boss*. Voor het managen van de verschillende states/behaviours van de AIs heb ik gebruik gemaakt van de Panda Behaviour Tree. Ik heb voor deze behaviour tree gekozen omdat er veel documentatie online te vinden is en het snel en makkelijk te begrijpen is.

Ik wilde een poging wagen om mijn eigen behaviour tree te ontwerpen, maar dit ging nogal moeizaam omdat ik niet precies wist waar ik moest beginnen. Voor het gemak heb ik daarom gezocht naar een bestaande behaviour tree waarbij ik mijn states kan managen door middel van nodes. In opdracht 2 is te zien dat ik bepaalde states heb ontworpen die met de speler, met de AI en met de wereld kunnen interacteren. Ik wilde ook nog een state toevoegen waarbij de Boss minions kan spawnen, maar om een of andere reden kon ik de AIs niet instantiëren omdat de behaviour tree dan niet meer zou werken.

Het idee van de boss battle is een beetje geïnspireerd uit de boss battles van Diablo, waarbij de AIs verschillende states hebben zoals het achtervolgen van de speler, patrouilleren van de wereld, schieten van spreken, en het creëren van schadelijke ondergronden. Deze states vond ik heel interessant en heb ik daarom ook zelf geïmplementeerd in mijn eigen game.

De Guards en de Boss hebben allebei een eigen Unit script waarin de states en behaviour staan beschreven. De reden dat ze elk een apart script hebben is omdat ik eerst alles in één script had, maar dat bleek al gauw onoverzichtelijk te worden. Daarnaast staan in de Unit script alle stats van de vijand, zoals movement, health en attack stats beschreven. Voor de visualisatie heb ik de speler en vijanden ook health bars gegeven. De waardes hiervan kunnen gemanipuleerd worden door bijv. de spreken en de lava pits. De vijanden hebben standaard een schild en als er een tempel vernietigd is, dan wordt er een boolean op false gezet waarna de vijand pas schade kan krijgen.

Als pathplanning algoritme voor de movement van de AI heb ik A\* geïmplementeerd. Dit algoritme had precies wat ik wilde bereiken, namelijk snel en effectief vanuit een startpunt de burens definiëren en het snelste pad naar het doelwit berekenen. Om mij een beetje op weg te helpen heb ik de A\* tutorial series van Sebastian Lague gevolgd en daarna de code uitgebreid door mijn eigen unit behaviours toe te voegen.

Tenslotte heb ik nog een simpel main menu, win en lose screen toegevoegd zodat er een loop ontstaat en het meer als een game aanvoelt.

## UML diagram:

