

# Kernmodule 2 - AI - Opdracht 3

## Behavior tree

Om het gedrag van de AI te beslissen gaat de agent stap voor stap zijn behavior tree branches af en kijkt bij elke branch of de node TaskStatus.Success returned.

Aan het begin van elk gedrag worden condities gecontroleerd om zo het gewenste gedrag te vinden bijvoorbeeld een afstand tussen de agent en een ander GameObject of dat een booleans state op true staat.

De root node van de behavior tree is altijd een PrioritySelector node die TaskStatus.Failed returned als alle meegegeven nodes dat ook doen.

Er moet dus een behaviour zijn die altijd TaskStatus.Success returned anders weet de AI niet wat hij moet doen, bijvoorbeeld een Idle behavior.

### **Gedrag 1 - Enemy - Guard**

Tree: BTPrioritySelector

Nodes:

- Stunned
- Patrol
- Engage

#### **- Stunned**

De Stunned behavior zorgt ervoor dat de guard stopt met alles wat hij doet voor een paar seconden en ook zijn state (boolean) reset. Dit zorgt ervoor dat de speler een kans heeft om uit het zicht te komen van de guard.

Stunned heeft de hoogste prioriteit en door stunned bovenaan op de lijst te zetten kunnen de andere branches niet worden uitgevoerd.

#### **- Patrol**

De Patrol behavior zorgt ervoor dat de guard van waypoint naar waypoint loopt en wanneer de guard bij zijn bestemming aankomt stopt hij en wacht daar dan voor een paar seconden.

Wanneer hij klaar is met wachten pakt hij de volgende waypoint in de lijst en als er geen waypoint meer is pakt hij weer de eerste waypoint uit de lijst en gaat vanaf daar weer verder.

Terwijl de guard rondloopt kijkt hij of de speler in zijn zicht is.

#### - Engage

De Engage behavior zorgt ervoor dat de guard naar zijn target (de speler) loopt en hem aanvalt wanneer hij in de buurt is. Als de guard zijn wapen niet heeft gaat hij eerst richting zijn wapen om die op te pakken.

Wanneer de guard dichtbij genoeg is om zijn target te raken speelt de guard een aanval animatie af.

Als de afstand te groot is tussen de guard en target gaat de guard uit zijn state en weer terug naar patrol.

### **Gedrag 2 - Ally - Ninja**

Tree: BTPrioritySelector

Nodes:

- Hide
- Follow

#### - Follow

De Follow behavior zorgt ervoor dat de ninja de speler blijft volgen.

De ninja sluipt achter de speler aan als de speler niet is ontdekt door de guard.

Wanneer de afstand te groot wordt krijgt de ninja een hogere move speed en een Run animatie om de afstand tussen de speler en de ninja te verkleinen, en als de afstand te klein is stopt de ninja met lopen en verandert de animatie naar een Idle animatie.

#### - Hide

De Hide behavior zorgt ervoor dat de ninja de dichtstbijzijnde schuilplaats opzoekt die niet in het zicht is van de guard. Wanneer de schuilplaats van de ninja in het zicht is van de guard terwijl de ninja op dezelfde positie staat gaat de ninja weer op zoek naar de dichtstbijzijnde schuilplaats die niet in het zicht is van de guard.

Nadat de ninja een aantal seconde op haar schuilplaats staat stunned de ninja de guard waardoor de guard zijn state reset. Hierdoor gaat de ninja weer in Follow behavior totdat de speler weer wordt gezien.

# Nodes

## Composite

### - Sequence

De Sequence node gaat stuk voor stuk de meegegeven nodes af.

Wanneer een node TaskStatus.Running returned dan wordt de task opgeslagen in een List met nodes als de List de node nog niet bevat.

Elke Node uit deze List wordt daarna afgespeeld totdat de node

TaskStatus.Success returned, waarna de node uit de List wordt gehaald.

Wanneer de List leeg is dan returned de Sequence node TaskStatus.Success en anders TaskStatus.Running.

### - PrioritySelector

De PrioritySelector node gaat stuk voor stuk de meegegeven nodes af.

Wanneer een node TaskStatus.Success returned herhaald de priority selector de node tot dat de node TaskStatus.Failed returned.

Als als alle meegegeven nodes zijn geweest returned de priority selector TaskStatus.failed.

Note: Als de root node een PrioritySelector is dan **moet** er een node zijn die altijd TaskStatus.Success returned .

## Decorator

### - BTCheckBool(VariableBool, Bool)

Vergelijkt de VariableBool.Value met de meegegeven Bool

### - BTSetBool(VariableBool, Bool)

Set de VariableBool.Value met de meegegeven Bool

### - BTCheckTaskStatus(BTBaseNode, params BTBaseNode[])

Als de return value van de BTBaseNode TaskStatus.Success is worden de meegegeven BTBaseNode's uitgevoerd. BTBaseNode[] kan één of meer BTBaseNode's zijn.

### - BTInverter

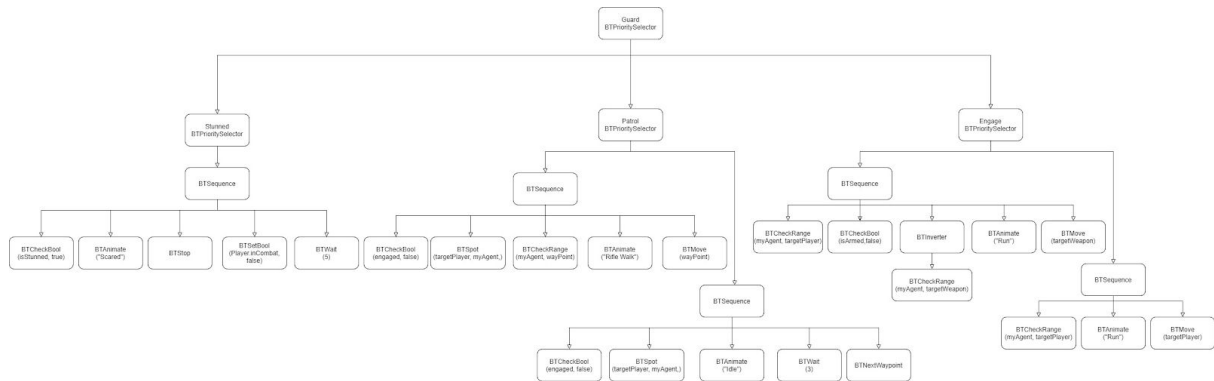
Geeft het omgekeerde return value mee.

TaskStatus.Success = TaskStatus.Failed.

TaskStatus.Failed = TaskStatus.Success

UML: [Kesaulija\\_Andi\\_Kermodule\\_2\\_AI.png](#)

Guard:



Rogue:

