Student: Pim Horeman - 3032469
Vak: Kernmodule 3 - AI
Docenten: Valentijn Muijrers & Vincent Booman

Kernmodule 3 - AI

Welk systeem is er toegepast

Uit de gegeven systemen die je kon gebruiken (Behaviour Tree, UtilitySystem of Goal Oriented Action Planning), heb ik gekozen om de behaviour tree te implementeren.

Wat is een behaviour tree?
een behaviour tree (ook wel “BT” afgekort) is een manier om het schakelen tussen verschillende taken te structureren in een zelfstandig object, zoals een robot of een virtuele entiteit (AI) in een computerspel.

Ik heb voor een behaviour tree gekozen omdat het mij het meest toepasselijk lijkt om een behaviour tree te gebruiken met virtuele entiteiten. Daarnaast vind ik het interessant om te bekijken hoe een behaviour tree werkt en waarom het een veel gebruikt systeem is. (Ik heb ook project zomboid gespeeld en toen ik in het gamasutra artikel las dat project zomboid gebruikt maakt van BT, wekte dat nog extra mijn interesse).

OntwerpKeuzes

UnityEngine:
Ik maak gebruik van de UnityEngine (versie 2019.4.9f1) omdat ik bekend ben met de UnityEngine en omdat er een voorbeeldproject te gebruiken was met unity. Het voorbeeldproject stond op versie 2019.4.9f1 en heb ik niet verder aangepast. Het veranderen van UnityEngine versies kan vaak niet veel kwaad maar het risico heb ik niet genomen voor de zekerheid.

Animaties:
Om het visueel representatiever te maken, heb ik gezorgd dat zoveel mogelijk animaties werken. Omdat het project al een aantal mixamo animaties bevat, heb ik nog een aantal mixamo animaties gedownload die ik toepasselijk/toepasselijker vindt in het project.
Het gebruik van animaties in het project zorgt ervoor dat de behaviour die de ai moet voltooien visueel fijn overkomt. Daarnaast is een het leuke leer stap om de animaties werkend te krijgen met een behaviour tree.

Guard:
Ik heb ervoor gekozen om de ‘guard’ het behaviour systeem te geven, omdat de ‘guard’ vrij algemene ai opties gebruikt. Dit geeft mij een duidelijke taken zoals waypoint patrol, enemy/speler gespot, jagen en schieten om te oefenen en uit te proberen met een behaviour tree.

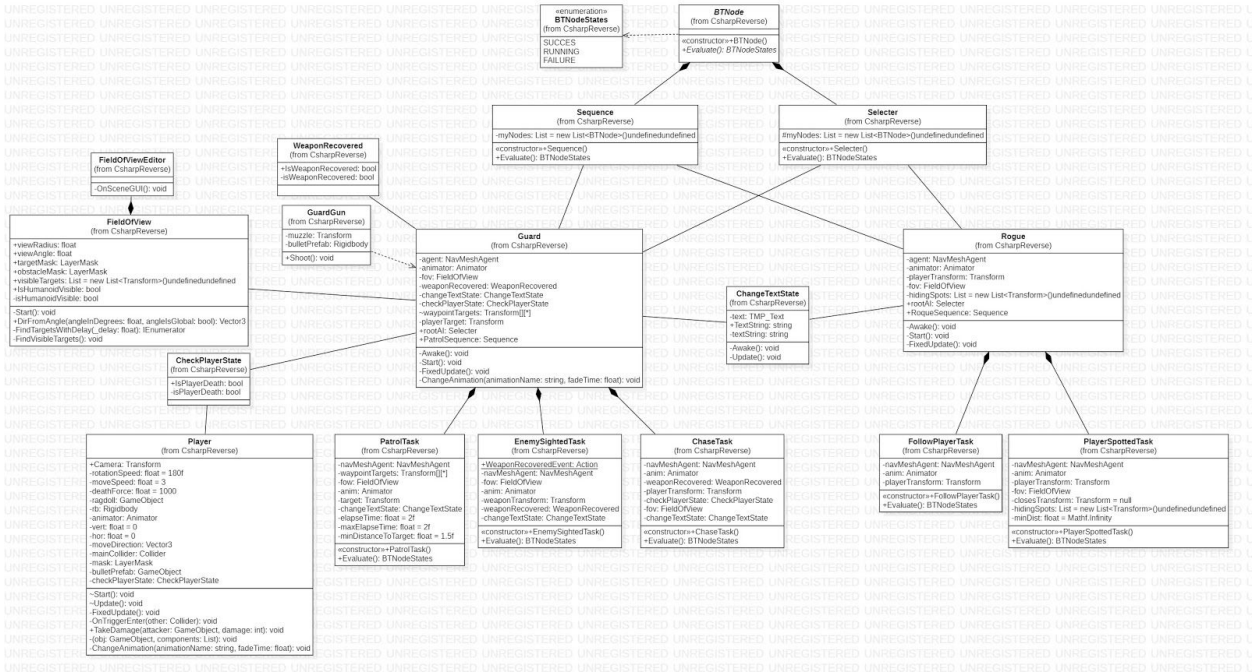
Student: Pim Horeman - 3032469
Vak: Kernmodule 3 - AI
Docenten: Valentijn Muijrers & Vincent Booman

Ik heb bij de guard een extra taak binnen de patrolling taak gestopt. Als de guard bij een waypoint aankomt dan stopt die even voor een x set seconden en gaat daarna pas door.
De animaties passen zich visueel aan, waardoor je duidelijker ziet dat de guard stil staat of weer gaat lopen. Een guard zal nooit constant rondjes lopen en zal af en toe alijd even stil staan, naast het experimenteren met de behaviour tree is het naar mijn mening een leuke toevoeging.

Ninja:
Ik wederom hier ook weer gekozen om een behaviour tree te gebruiken. Voornamelijk omdat er andere taken worden uitgevoerd in de ninja vergeleken met de guard. Helaas kwam ik in tijdnood tegen de tijd dat ik bij de ninja aankwam en heb ik een begin kunnen maken maar niet de volledige eigenschappen kunnen toevoegen.

Enemy/Speler Detectie:
Ik wist in het begin niet goed hoe ik dit wilden aanpakken. Ik begon met een Raycast maar dat werkt niet geheel naar mijn verwachtingen omdat ik het geprogrammeerd met line in plaats van cone. Daarna stapte ik over naar een Physics.overlapSphere maar hier kon ik iedereen binnen mijn 360 radius vinden wat een leuk idee is maar dat moest wederom aangepast worden naar een cone. Dus ik ging voornamelijk onderzoek plegen naar hoe je een cone of een custom radius kon geven aan je line of side. Sebastian Lague wist mij te helpen met mijn probleem en dat zit huidig in de guard als detectie (FieldOfView.cs).

UML Diagram



UML Diagram zal nogmaals los meegeleverd worden.

Student: Pim Horeman - 3032469
Vak: Kernmodule 3 - AI
Docenten: Valentijn Muijrers & Vincent Booman

PMI

Plus	Min	Interessant
Het leren van verschillende soorten systemen.	Voornamelijk moeilijk om te durven vragen omdat je geen fysiek contact hebt.	Het voornamelijk zelf laten uitzoeken.
Het implementeren van 1 of meerdere van de systemen.	Uiteraard is corona een spijtige gebeurtenis	Ik had erg veel moeite met het implementeren van een Behaviour Tree
leuk idee, om te beginnen met boids en A*	Persoonlijke min: omdat ik zelf heel erg bezig was met het begrijpen en onderzoeken van behaviour trees heb ik de rest van de code niet super mooi neergezet.	
Duidelijke uitleg en slides over de onderwerpen		

Bronnen

De lessen/slides hielpen om de stof te begrijpen met een eerste stap.

De onderstaande tutorial gaf mij inzicht in het gebruiken van een behaviour tree:
Build a Behavior Tree AI in Unity Part 1 - Making a Space Game Tutorial - YouTube

Het onderstaande artikel gaf mij informatie en inzicht in het toepassen van een behaviour tree:
Gamasutra: Chris Simpson's Blog - Behavior trees for AI: How they work

De onderstaande tutorial hielp mij om vijanden/spelers te detecteren:
Field of view visualisation (E01) - YouTube

UML Diagram Informatie:
UML Class Diagram Tutorial - YouTube
UML Class Diagram Tutorial with Examples (guru99.com)

Github:
Pimmez/Kernmodule3_BehaviourTree: Kernmodule 3 - AI (BehaviourTree) (github.com)
Gameplay Video Kernmodule 3:
https://youtu.be/nbYB6GzYee4