Inleiding

In de les zijn een aantal topics aan bod gekomen:

* Character
* Animations
* Navigation
* AI

Deze taak heeft als doel na te gaan in hoeverre je deze topics beheerst. Om dit te doen wordt verwacht dat je in unity een kleine spel uitwerkt die deze thema’s toepast. De grafische look n feel van dit spel is ondergeschikt aan de technische implementatie zodoende wordt ook verwacht dat je gebruik maat van ProBuilder om je level op te bouwen en dat je character en NPC’s simpele Y-Bots zijn, die gebruik maken van de ip mixamo aanwezige animaties. Ook de fun factor van dit spel is ondergeschikt aan die technische kant.

Voor je echter begint met programmeren wordt er echter verwacht dat je eerst een voorstel maakt van het “prototype” die je maakt, dit om eerst en vooral te zorgen dat er duidelijke afspraken zijn, en zodat uw docent kan nagaan of uw voorstel voldoende content heeft om u kennis na te gaan.

Technische verwachtingen

Character controller

In je spel zorg je voor ervoor dat je een eigen avatar kunt aansturen met een controller. Het spel zal getest worden met een Xbox One controller.

De keuze is aan u om te opteren voor een implementatie die gebruik maakt van een Character Controller Component of een RigidBody. Het is wel zo dat er verwacht wordt dat u voor het aansturen van uw character zich baseert op de weten van de fysica. Dus random getallen die vanuit het niets komen en die niet gestaafd zijn zijn niet aanvaarbaar. Van ieder getal die je character manipuleert wordt verwacht dat je ook de gebruikte SI-eenheid aangeeft. Wil je dus je character achteruit laten vliegen door een explosie datn verwacht ik dit uitgedrukt te zien als een kracht die resulteert in een versnelling van je avatar.

Je avatar moet minstens op een klassieke manier kunnen rond bewegen, tenzij voor jou prototype beperkingen nodig zijn om het werkende te krijgen.

Je avatar moet ook naast de klassieke locomotion ook 2 extra acties kunnen ondernemen. ( bijvoorbeeld: Klimmen op een muur, hangen aan een touw, klimmen op een ladder, springen naar een verhoog, springen tussen platformen)

Je avatar moet ook in interactie kunnen treden met minstens drie spel elementen, waarvan hij er minstens twee moet kunnen manipuleren volgens de regels van de fysica. Zorg dat beide elementen genoeg van elkaar verschillen. (Het rond duwen van een blok , het draaien aan een rad, het trekken aan een koord zijn maar enkele mogelijkheden) Het is aan u om desnoods even op bijvoorbeeld wikipedia na te gaan welke wetmatigheid daarvoor van belang is en deze op een correcte manier te implementeren. Zorg dat bij minstens 1 van de actief ook gebruik gemaakt wordt van inverse kinematics. (Bijvoorbeeld: De hand op de juiste plaats zetten bij het oppikken van een element. De voet op de juiste tegel zetten die ingedrukt wordt. Een hand plaatsen op een schakkelaar, )

Animaties

Zorg dat alle acties van je avatar en je npc’s gepaard gaan met de juiste animaties. Zorg dat de timing van de animaties correct zijn en de overgangen correct gebeuren. Splits je animaties proper uit in verschillende lagen zodat bijvoorbeeld je locomotion animaties niet gemengd zijn met je interactie animaties of je IK laag.

Navigation an AI

Zorg dat er in je spel ook NPC’s zijn. De rol die ze hebben binnen je prototype is vrij te kiezen. Deze mogen vijanden zijn of NPC’s waarmee je in interactie kunt gaan of die je moet manipuleren.  (Het kunnen vijanden zijn die je moet schieten, of die je moet vangen, of die zich verbergen, of die je moet begeleiden, of die je … de mogelijkheden zijn oneindig)

Wel wordt van deze NPC’s verwacht dat ze een eigen gedrag hebben die je aanstuurt door middel van een behaviour. Deze moet complex genoeg zijn zodat het lijkt alsof je NPC intelligente beslissingen neemts (bvb: hij kist een pad waarbij hij zich geregeld verbergt achter een obstacle, hij gaat actief op zoek naar jou avatar die hij probeer neer te schieten, hij verzamelt elementen uit de scene die hij naar een bepaalde plaats breng, … .)

Verder moet jou NPC ook in staat zijn zich doorheen je level te bewegen door middel van een navigation mesh. Deze navigation mesh moet minstens 2 locatie bevatten waar je de animatie veranderd van de standaard animatie naar een andere animatie (bvb: klimmen op een touw, springen van een hoogte, … )

Documentatie

Game Design

Voor je aan je project mag beginnen moet je eerst toestemming krijgen van je docent. Deze geeft toestemming op basis van een Game Design Document en een level blockout in probuilder

In dit beperkte Game Design Document beschrijft eerst en vooral je core mechanics.  Deze zullen duidelijk weergeven wat je avatar kan en hoe hij in interactie treed met de level en de NPC.

Acties zoals klimmen op touwen, schieten naar een NPC, beschoten worden door een NPC,  Praten met een NPC, een NPC jou laten volgen enz. moet dus uitvoerig genoeg beschreven zijn zodat anderen duidelijk weten wat de bedoeling is en hoe het spel zal werken.

Verder bevat jou document ook een flowboard. Dit is een  combinatie van een storyboard en een flowchart die aangeeft hoe de verschillende gameplay modes tussen elkaar overgaan.   Een gameplay mode is een situatie in je spel waarbij een bepaalde “interface” en gameplay hoort. Zo zou je bijvoorbeeld een locomotion gameplay mode hebben waarbij je met je joystick in de 4 richtingen kunt bewegen en op de a kunt drukken om in interactie te treden met iets. Bij een druk op A kun je bijvoorbeeld op een touw klimmen en kom je dus in een “klimmen “gameplay mode waarbij je de joystick kunt gebruiken om op en neer te bewegen op het touw, A kunt gebruiken om je op te trekken of B om het touw te lossen.

Een gameplay mode beschrijft dus bepaalde acties die je kunt ondernemen en de manier waarop je deze moet uitvoeren. Je flowboard geeft aan hoe je tussen de verschillende gameplay modes overgaat.

Als laatste wordt verwacht dat je ook reeds een level design maakt in ProBuilder. Je mag gerust je level blockout gebruiken om je game playmodes te documenteren.

Eenmaal je Game design document uitgeschreven is, moet je deze opsplitsen in verschillende taken die je daarna in hack’n plane bijhoudt. Probeer op voorhand voor iedere taak in te schatten hoeveel tijd je ervoor nodig zult hebben. Je zou ongeveer aan een 100tal uren werk moeten uitkomen. Indien je merkt dat je te weinig of te veel tijd hebt neem je contact op met de docent en in overleg kan de opdracht aangepast worden.

Time Tracking

Er wordt verwacht dat je via toggle in detail bijhoudt waaraan je je tijd besteed. Alle tijd moet bijgehouden worden. Ook voor alle randtaken zoals opzoek werk die je doet om de opdracht tot een goed einde te brengen. Zorg dat je duidelijk aangeeft wat je gedaan hebt. Deze documentatie dient vooral om de werklast van de opdracht in te schatten, in te schatten hoe vlot en efficiënt jij kunt werken en hoeveel werk je aan deze opdracht hebt besteed. Finaal zou je gemiddeld gezien een 100tal (efficiënte) uren aan deze opdracht moeten besteden.