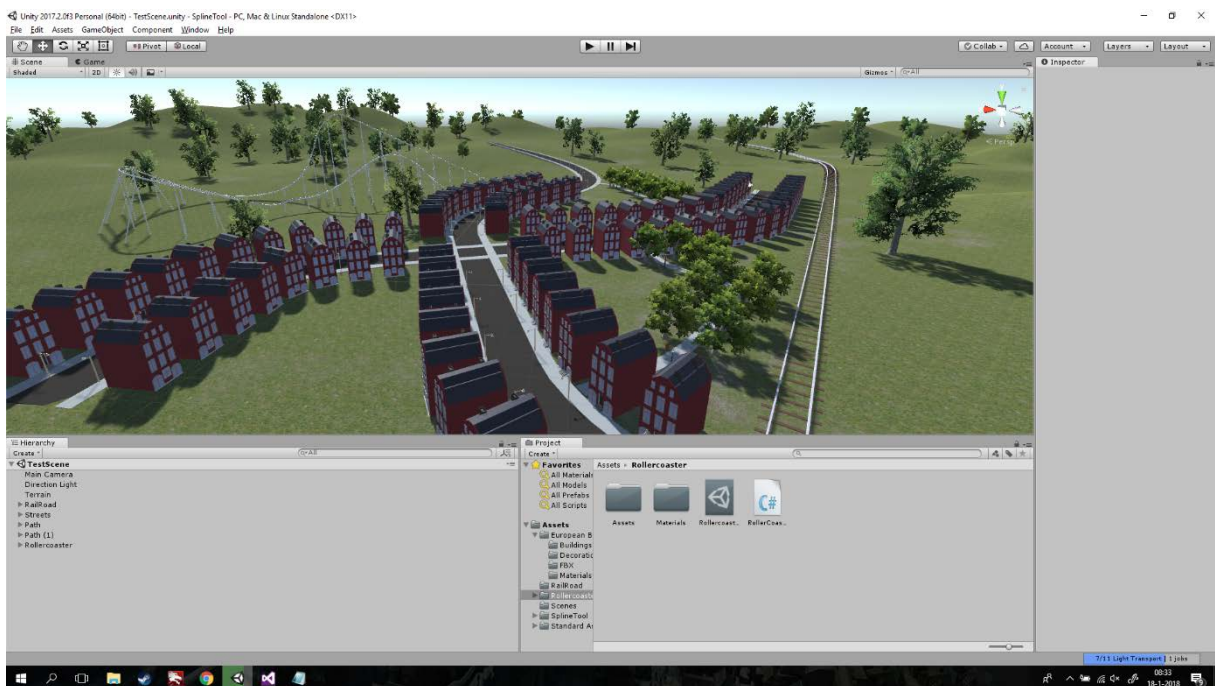


Kernmodule Game Development 2



Ruben Bimmel
GDV2

Kernmodule GDEV 2
Aaron Oostdijk
Valentijn Muijers

19-1-2018
Versie 1

Inhoud

Project overzicht.....	3
Mijn eerste editor in unity.....	3
Toelichting knoppen.....	3
Class Diagram	4
Activity Diagram	5
Van tutorial naar eigen tool	6
Proceduraly generated content	6
OOP vs unity editor	6
Deadline voor nieuwe features	7
Buddysysteem	7

Project overzicht

Voor dit vak heb ik een tool gemaakt voor het maken, bewerken en gebruiken van splines in Unity. De tool bestaat uit 2 onderdelen. De eerste is het maken van de splines zelf. Dit werkt door gebruik te maken van de SplineComponent class. SplineComponent is een soort facade class voor alle functionaliteit van de splines. Een spline component kan meerdere splines bevatten, maar het is wel de bedoeling dat deze bij elkaar horen. Het bewerken van de handles van een spline component werkt met de standaard unity gizmos en ondersteund schalen, roteren en werken in global of local space. Daarnaast kan je handles ook via de inspector bewerken doormiddel van een positie vector, een euler angle vector en één of twee floats voor de schaal. Dit lijkt dus ook erg veel op de manier waarop je een transform kan bewerken.

Het tweede grote onderdeel van de tool is het maken van slimme objecten die gegenereerd kunnen worden langs de splines. Dit werkt met een SplineSettings asset. Hierin kan je objecten toevoegen en precies instellen hoe ze langs de spline geplaatst moeten worden. De objecten worden realtime door de tool in de scene gezet. Hierdoor is het erg makkelijk toe te passen in je normale workflow.

Mijn eerste editor in unity

Voor dit project had ik nog geen ervaring met het maken van custom editors en inspectors in Unity. Het schrijven van GUI code was ook nieuw voor mij. Het is niet de moeilijkste code om te begrijpen en om te kunnen schrijven, maar ik merkte wel dat het hierdoor lastig was voor mij om van te voren te bedenken hoe de classes geschreven moesten worden. Ik heb hierdoor tijdens het project mijn plannen een aantal keer aan moeten passen.

Toelichting knoppen

Edit

In deze modus kan je de handles bewerken

Junction

In deze modus kan je punten bewerken waar meerder splines samenkomen

Spline

In deze modus kan je de SplineSettings instellen voor een spline en instellen welke assets er wel en niet actief zijn op de spline



Voeg punt toe **na** het huidige punt



Voeg punt toe **voor** het huidige punt



Verwijder het huidige punt

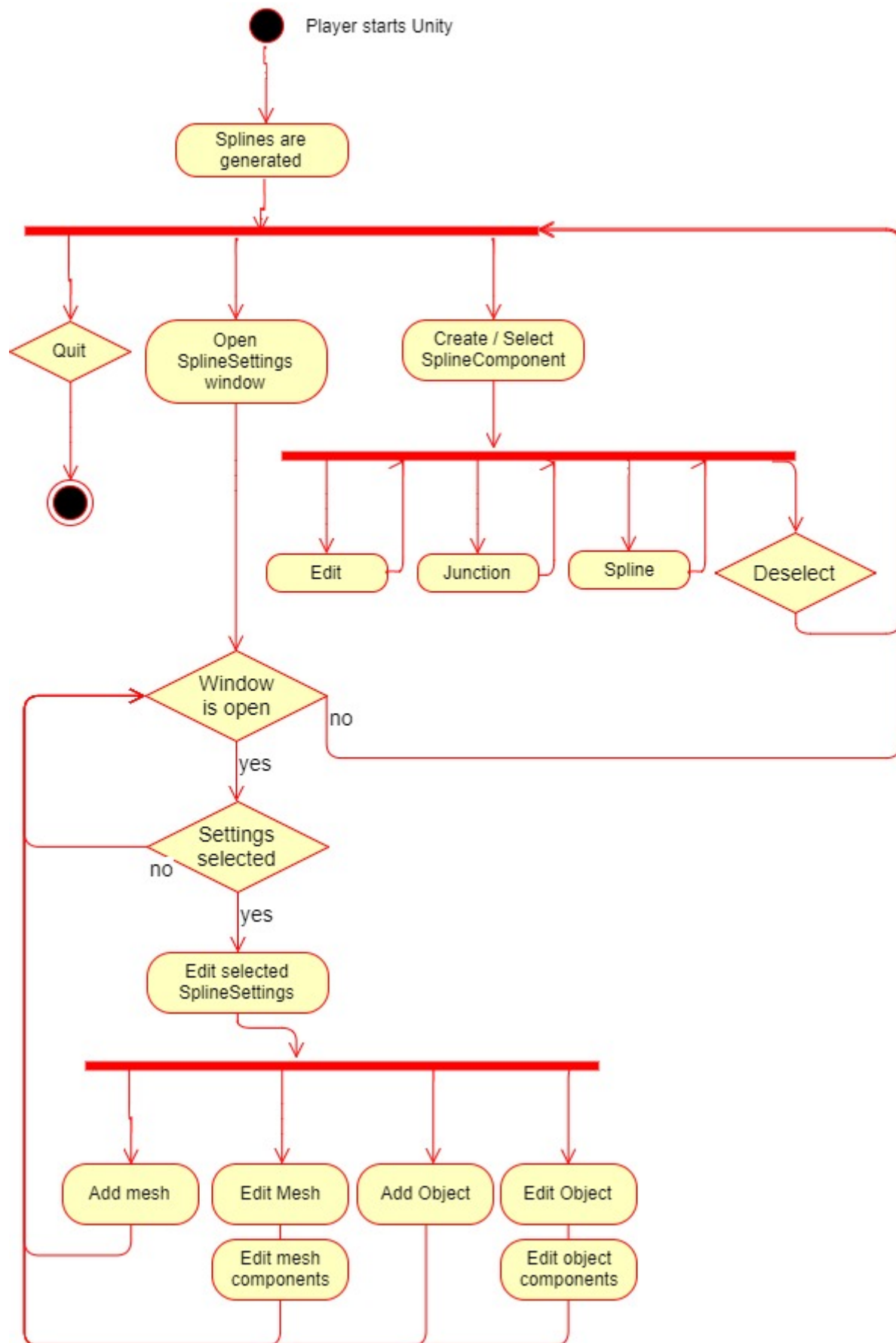


Start een nieuw spline vanaf dit punt

Class Diagram



Activity Diagram



Van tutorial naar eigen tool

Omdat ik zelf nog geen ervaring had met splines en ook niet met het schrijven van een custom editor ben ik begonnen met het volgen van de tutorial van *CatlikeCoding.com*. Daarna ben ik deze tool om gaan bouwen naar mijn eigen tool. De grootste verandering is dat de tutorial alle punten op slaat in één lijst. Ik heb er voor gekozen om een handle class te maken waar de 3 punten van een handle in zitten. Door van de handles een slim object te maken is het makkelijker om deze te bewerken.

Een van de dingen die wel zijn overgebleven uit de tutorial is dat alle referenties naar punten en splines worden doorgegeven met een index integer. Dit had ik achteraf beter anders kunnen doen want door alle dingen die ik tijdens dit project ben toe gaan voegen worden al deze indexen erg onduidelijk om mee te werken.

Procedurally generated content

Ik wilde voor dit project heel graag iets doen met procedural generation. De splines leken mij hier een heel leuk project voor. Ik had zelf nog geen ervaring hiermee en ik kon niet heel veel goede tutorials terug vinden over hoe je dit moet aanpakken.

Ik heb er voor gekozen dat de tool alle objecten in de scene hierarchy zet. Unity gaat hier niet heel goed mee om en daarom heb ik het zo moeten schrijven dat deze objecten niet worden opgeslagen wanneer je de scene afsluit. Daarnaast is een groot nadeel hier van dat je heel veel losse objecten krijgt in de scene, wat kan leiden tot wat problemen met optimalisatie.

Een voordeel van deze methode is dat je in principe de objecten later kan loskoppelen van de splines en zo een normale scene met unity objecten krijgt. Doordat de objecten in de editor geplaatst worden, en niet in de build, kan je ook gebruik maken van dingen als static lightmaps voor deze objecten. Hierdoor denk ik dat deze tool zeker handig kan zijn om snel een environment te bouwen.

OOP vs unity editor

Ik had voor dit project nog geen ervaring met het schrijven van GUI code. Hierdoor zijn de classes van dit project niet zo netjes geworden als ik gehoopt had. Sommige classes zijn veel te lang geworden en doen veel meer dan één ding. Eigenlijk had ik het liefst een groot deel van de classes opnieuw willen schrijven zodat ze wat meer aan de regels van SOLID object georiënteerd programmeren voldoen maar hier kwam ik pas te laat in het project achter.

Daarnaast merkte ik dat het schrijven van Unity editors het ook moeilijk maakt om netjes object georiënteerd te programmeren. Zo heb ik geprobeerd om de classes *GeneratedMesh* en *ObjectPlacer* allebei child classes te maken van een class genaamd *SplineAsset*. Dit maakt het een stuk netter om deze classes op te slaan in een lijst. Dit heb ik na twee weken weer teruggedraaid omdat de inheritance verloren ging tijdens het serialiseren.

Deadline voor nieuwe features

Van het vorige project had ik geleerd om niet op het laatste moment nog nieuwe features toe te gaan voegen. Ik heb er daarom een deadline voor mezelf gezet om vanaf de kerstvakantie geen nieuwe dingen meer toe te voegen. Zo had ik onder andere de mogelijkheid willen toevoegen om meerdere punten te kunnen selecteren en te bewerken en om loops te kunnen maken door punten met elkaar te kunnen verbinden. Zonder deze functies is de kracht van meerdere splines in een component niet heel groot meer. Achteraf had ik hier misschien beter niet op kunnen focussen.

Buddysysteem

Ik heb niet vanaf het begin van dit nagedacht over wat het buddysysteem zou worden. Omdat mijn tool erg veelzijdig is ging ik er vanuit dat het met heel veel tools wel te combineren zou zijn. Omdat we echt data moesten uitlezen van iemand anders (en niet alleen output gebruiken) werd dit wel een stuk lastiger. Uiteindelijk heb ik samengewerkt met Daan van Westerlaak. Zijn tool is een soort vertex paint systeem voor faces van een object. Omdat onze data allebei bestaan uit een lijst met vectoren leek het ons handig om daar iets mee te doen.

Daan had in zijn tool een sphere gebruikt en rondom die sphere allemaal vlakjes ingekleurd. Doordat hij een pad volgde rond de sphere kon ik dat pad omzetten in een spline. De methode berekend het middelpunt van alle driehoeken en verbindt ze met elkaar in de volgorde waarop Daan ze had aangeklikt. Ondanks dat dit buddysysteem een redelijk geforceerd idee was is het resultaat best leuk geworden. Als je met mijn tool een spline had willen maken die om een sphere draait dan had dat veel meer tijd gekost.