

## Project 47 Flyer

### Autodesk Maya – Xsens MVN Studio PLUG-IN

Precieze werking van de plug-in is te vinden op de documentatie pagina. In deze flyer alleen kort de

workflow van de plug-in team, om het zo kort mogelijk te houden.

Deze flyer is meer voor evaluatie doeleinden voor onszelf en voor de beoordelaar zodat die een idee krijgt van wat voor technieken er allemaal gebruikt zijn om het geleverde product te omschrijven.

#### **Intro:**

Om te beginnen stond het eind doel van het project niet geheel vast bij de Opdrachtgever.

Uiteindelijk kwam het er een beetje op neer dat er verschillende modellen gemaakt moesten worden en het liefst een soort plug-in voor MVN Studio of ZignTrack om real-time motion capture data uit te kunnen oefenen in Maya.

Verschillende teams wilden beginnen met MVN Studio en hadden daartoe ook een poging gedaan, maar het later laten vallen. Wij zijn het enige team die ermee door gegaan zijn, en het is ons gelukt.

#### **Hoe hebben we het aangepakt?:**

1. Om te beginnen, we wisten allen helemaal niets van MVN Studio of Maya dan ook.

Het is altijd uitdagend om software te schrijven op al bestaande programma's, omdat je op ontzettend veel manieren beperkt kan worden.

We hebben dus eerst gekeken naar wat beide programma's ons aan te bieden hadden.

Maya komt met een heel python console ingebouwd (Python 2.6) en een eigen API met alle benodigde modules om Maya commando's uit te kunnen voeren met Python.

Voor mij was dit erg mooi, ik zelf begon een paar maanden daarvoor met Python en ik heb het dus snel aangeraden aan de andere teamleden, omdat Python in zoveel 3d programma's gebruikt wordt, ik er zelf beter mee wilde worden en de andere optie (MEL) is een programmeer taal die alleen in Maya zelf gebruikt wordt. Dus niet super interessant.

Mvn Studio kwam met de mogelijkheid om over UDP naar [IP][POORT] data te versturen die een gehele positie omschreef. Dit process kon uitgevoerd worden tijdens een capture sessie. Dus het was gewoon mogelijk om real-time data te versturen naar iets.

Hierop hadden we al gauw een code in elkaar geknutseld die de data opvangt en uitprint. Wat we eruit kregen, zie bijlage.

Verder raadde ik het aan om een documentatie pagina bij te houden zodat het uiteindelijk voor onszelf en voor toekomstige gebruikers makkelijk over te nemen is.

2. Na dat gezien te hebben dachten we OK, waar te beginnen, waar te beginnen?

Gedurende het project had ik in het begin uit eigen initiatief contact opgenomen met Xsens om te kijken of we hun kennis niet als consultancy konden gebruiken en dergelijke.

Xsens antwoorde mij hierop en zei dat ze zelf toevallig ook bezig waren met een plug-in voor Maya.

Toen dachten we, Misschien is dat wel mooi voor ons! Kunnen we geen kennis uit

wisselen? Maar daarop antwoorde Xsens ons en deed een aanbod om één of ander SDK te kopen voor 1500 euro. Nou, we weten hoe dat gaat .. dus dat werd hem niet voor ons.

We zaten even vast, we wisten helemaal niets van het abracadabra bericht dat we binnen kregen en zochten allerlei manieren, en protocollen om het proberen te vertalen. Maar we bleven door gaan en waren gedurende het hele project eigenlijk vast beraden om het af te maken.

Wel gingen we verder met de achterliggende werking van de plug-in. Ik had een hele state machine uitgedacht waarin de plug-in zich eigenlijk afspeelt en eigenlijk altijd voor zekerheid zorgt. We wisten dus wat er in elke staat gebeuren kon en in welk deel van de code we het vertalen van de abracadabra code zouden doen.

**3.** Het ziet er intimiderend uit, maar er blijkt wel een structuur in te zitten. Hier kwamen we plots achter toen de opdrachtgever zelf ons een .pdf overhandigde waarin het protocol tot in detail beschreven stond.

**4.** Daarop hebben we vertaal code geschreven die elke keer uit die hele lange reeks de juiste bytes pakt, omzet naar 1 float waarde (4bytes) en toekend aan het juiste lichaams onderdeel en de juiste as (XYZ). Er moest veel gedebugd worden en geoptimaliseerd om een zo real-time mogelijk effect te krijgen. We hebben eigenlijk constant 1 bericht uit de 'sample' genomen om alles te testen. En het geheel werkte allemaal, missie geslaagd.

*We zijn ontzettend blij, en we hopen dat men er nog veel plezier aan beleefd. En vooral dat het project wordt voortgezet door komende jaren of zelfs mensen buiten de school.*

### **Afsluiting:**

Naast de oneindige uren van testen, debuggen en research,

Uiteindelijke onderdelen van de Plug-in waar aan is gewerkt.

- State machine-like code
- Data receive code
- Data process code
- Threading
- Template 23 locators
- En zoveel om op te noemen verder ..

Uiteindelijke andere producten:

- Documentatie pagina
- GitHub repository (Opensource, iedereen kan er mee verder gaan)

**Joey en geheel subteam 47**