

INFRAROOD CAMERA DECODING TOOL

ROEL ZWAKMAN - KERNMODULE GDV 2

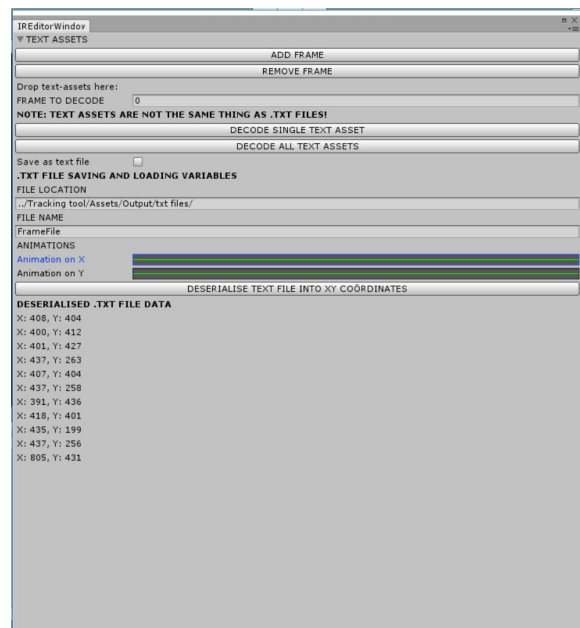
GEBRUIKSAANWIJZING

WAT DOET HET?

Deze tool maakt het mogelijk om met een infraroodcamera of een webcam in de juiste omstandigheden XY-bewegingscoördinaten op te slaan naar een .txt-bestand, en/of om te zetten naar een AnimationCurve in Unity3D.

HOE GEBRUIK JE HET?

1. Deze tool werkt het beste als losstaand project, dus download het project en open hem in Unity3D.
2. Ga naar de folder "Scene", en open de Capture scene. Als het goed is, zie je nu in de Game View twee witte vlakken met daartussenin een knop met de text "**START RECORDING**". Druk op de Play-knop bovenaan en in het midden van de editor.
3. Je ziet nu links een livefeed van de camera, en rechts een deels verwerkte livefeed van deze camera. In de rechterfeed zie je rode pixels waar het beeld wit genoeg is / de infraroodlampjes in beeld zijn. Als je tevreden bent, klik je op de middelste knop, en kan je beginnen met het opslaan van de data. Als je met een webcam werkt kan het zijn dat er veel van de linkerfeed wit is, en daarom op de rechterfeed rood. In dit geval raad ik aan om een zwart kleed achter je op te hangen of om een infraroodcamera aan te schaffen en te werken met infraroodlampjes.
4. Als je hebt opgenomen wat je wilde, moet je in de Project-view de folder "Text Assets" openen en vervolgens *Rechtermuisknop* → *Refresh* aanklikken. Er komen nu .asset bestanden in de folder te staan. *Let op: als je opnieuw gaat recorden, dan worden deze bestanden overschreven! Ik raad aan om meteen de volgende stap te nemen!*
5. Om deze data beter te processen, klik je bovenaan in de editor op *Window* → *Infrared mo-cap editor*. Als het goed is komt het volgende menu dan in beeld:



6. Klik onder "TEXT ASSETS" op de knop "ADD FRAME", en sleep vervolgens op de volgorde waarin je de data terug wilt krijgen de Text Assets (.asset) bestanden vanuit de Project/Text Assets naar de velden die erbij zijn gekomen.
7. Als je in een specifieke volgorde de data wilt decoderen, typ je in het veld "FRAME TO DECODE" het getal van de positie in van de frame die je wilt decoderen. Druk vervolgens op de knop met "DECODE SINGLE TEXT ASSET". *Let op: de lijst telt vanaf 0, dus als je de 6e in de lijst van boven naar onder wilt decoderen, typ dan 5!*
8. Als je elke frame tegelijk wil decoderen, druk dan op de knop "DECODE ALL TEXT ASSETS".
9. Als het goed is, zul je de groene lijnen onder het kopje "ANIMATIONS" na het drukken van een van deze knoppen hebben zien veranderen. Deze lijnen zijn het resultaat van je opgenomen data!
10. Als je deze data wilt overzetten naar een andere computer, kun je het hokje met "Save as text file" aanvinken. De volgende keer dat je op een van de "DECODE" knoppen drukt, zal dit worden opgeslagen in de locatie die is aangegeven onder het kopje ".TXT FILE SAVING AND LOADING VARIABLES". Als je deze data vervolgens wilt uitlezen, druk je op de knop "DESERIALISE TEXT FILE INTO XY COÖRDINATES".

DOCUMENTATIE

WAT HEB IK GELEERD TIJDENS DIT PROJECT / WAT KAN ER BETER?

Ik heb in dit project een aantal slechte dingen over de structuur van mijn code ontdekt, die ik deels achteraf heb kunnen oplossen maar deels ook niet. Een deel van mijn code zit met references in de inspector in elkaar, wat het toevoegen van functies in sommige gevallen in de weg zou hebben kunnen zitten.

Gelukkig is dat vooral het deel wat zodra het klaar is niet veel meer aan hoeft te gebeuren, en zijn andere delen wel los van elkaar te gebruiken. Een voorbeeld daarvan is de (de)serialisation-code, die in een losse class die niet van MonoBehaviour erft en static functies gebruikt. Daarnaast is de code in van de EditorWindow vrij losstaand, en is deze alleen een composition met InfraredImageDecoder, en wordt een Object die de InfraredImageDecoder-class gebruikt altijd in de EditorWindow zelf tijdelijk aangemaakt, zodat de reference er altijd is.

Het grootste ding dat ik geleerd heb tijdens dit project is de basis van het schrijven van Compute Shaders, en hier ben ik dan ook het meest blij om. Hoewel ik het decoderen van de camerafeed niet volledig real-time heb kunnen doen, wordt er real-time wel al een hoop van het voorwerk gedaan, wat ervoor zorgt dat het uiteindelijke processen van de data veel minder lang duurt dan het anders had gedaan. Op de GPU wordt nu namelijk al de brightness-check gedaan, en een deel van het omzetten naar losse punten om te gebruiken als XY-coördinaten. Het enige wat niet real-time gebeurt is de laatste pass van het proces waarin ik van gebieden met licht naar XY-coördinaten die een enkele positie per infrarood-lampje representeren ga.

UML CLASS DIAGRAM

