

Informe inicial: Entorn de simulació per la captura d'imatges des de dron

Kevin Martín Fernández

1 INTRODUCCIÓ

LA simulació de espais per a la generació de imatges artificials és un àmbit en el que es busca representar el món real de la forma més realista possible per tal de poder generar informació que en entorns reals ens representaria un perill o un cost econòmic elevat, d'aquesta forma es pot arribar a ensenyar als algorismes d'aprenentatge reduint la quantitat d'informació real que es necessitaria.

En aquest treball és vol aconseguir que mitjançant dades obtingudes a través de càmeres de diferents mitjans aeris com poden ser satèl·lits, avions o drons podem representar terreny real en entorns simulats.

2 OBJECTIUS

En aquest apartat determinarem els diferents objectius del projecte en format de jerarquia per tal de veure la dependència entre els diferents objectius:

1. Analitzar
 - 1.1. Investigar l'estat de l'art
 - 1.1.1. Estudiar AirSim
 - 1.1.2. Estudiar Carla SIMULATOR
 - 1.1.3. Estudiar proposta pròpia
 - 1.2. Definir els objectius
 - 1.3. Investigar motors gràfics
 - 1.3.1. Investigar Unreal Engine
 - 1.3.2. Investigar altres motors gràfics
 - 1.4. Investigar els mapes d'altura
 - 1.5. Investigar les imatges tant RGB com multi-espectrals
 - 1.6. Investigar webservices WMS
 - 1.7. Investigar llibreries RCP
2. Definir
 - 2.1. Definir mòduls interessants pel projecte
 - 2.2. Definir l'estructura del software
 - 2.3. Definir plataformes utilitzades
 - 2.3.1. Definir els mòduls a desenvolupar
 - 2.3.2. Definir la comunicació entre els mòduls
 - 2.3.3. Definir estructura de les dades que rebrà Unreal Engine
3. Desenvolupar
 - 3.1. Desenvolupar modul de transformació i obtenció de dades
 - 3.1.1. Desenvolupar el modul de transformació de dades d'altura i textures
 - 3.1.2. Desenvolupar l'importador de dades en Unreal Engine
 - 3.2. Desenvolupar modul gràfic (Unreal Engine)
 - 3.2.1. Desenvolupar la interfície del menú
 - 3.2.2. Desenvolupar el mòdul per importar fitxers personalitzats en Unreal Engine
 - 3.2.3. Desenvolupar codi per la generació del terreny
 - 3.2.4. Desenvolupar codi per la generació dels materials
 - 3.2.4.1. Visualitzar diferents materials (RGB, Multiespectre, etc)
 - 3.2.5. Desenvolupar codi de servidor per control del vehicle
 - 3.3. Desenvolupar modul de scripting
 - 3.3.1. Desenvolupament client que controlara el vehicle
 - 3.4. Integrar els mòduls de AirSim en Unreal Engine
 - 3.4.1. Integrar modul de Drons en el projecte
 - 3.4.2. Integrar modul de Segmentació en el projecte

- 3.5. Desenvolupar altres capes d'informació
- 4. Testejar
 - 4.1. Fer provàs del modul de transformació de dades
 - 4.1.1. Provar amb dades locals
 - 4.1.2. Provar amb dades externes
 - 4.2. Fer provàs del modul gràfic
 - 4.3. Fer provàs del modul de control per scripting
 - 4.3.1. Elaborar script d'exemple
 - 4.3.2. Provar script d'exemple
- 5. Documentar
 - 5.1. Redactar informe inicial
 - 5.2. Redactar informe de seguiment I
 - 5.3. Redactar informe de seguiment II
 - 5.4. Redactar el informe final
 - 5.5. Elaborar proposta de presentació
 - 5.6. Elaborar pòster
 - 5.7. Gestionar la documentació del dossier

3 ESTAT DE L'ART

Actualment existeixen diverses aplicacions per la generació d'imatges en entorns que simulen la realitat amb la finalitat de generar dades per algoritmes d'aprenentatge. En aquest àmbit unes de les més importants és AirSim [1] desenvolupat per Microsoft sobre el motor gràfic Unreal Engine aquest entorn permet la simulació de cotxes i drons, control per scripts, simulació climatològica, simulació del sol, generació d'imatges en visualització com son rgb, segmentació o profunditat, etc... Per altra banda un altre entorn similar al anterior és Carla SIMULATOR [2] aquest software esta enfocat en cotxes ja que el seu principal objectiu es obtenir dades per l'aprenentatge de cotxes autònoms.



Fig. 1: Simulador Airsim

4 METODOLOGIA

En aquest projecte hem decidit utilitzar una metodologies de tipus Agile [3] ja que això ens permetrà identificar de una forma millor les petites parts de les que es compona aquest projecte, a més a més d'adaptar-se als canvis imprevistos. En concret s'ha escollit la tècnica Kanban [4] que consisteix en organitza el nostre backlog (tasques de curta duració) en targetes que ficarem en un taulell segons en quin punt del cicle de vida de la tasca es trobi (pendent, començada, etc) per això s'ha decidit utilitzar l'eina Trello [5] que permet de forma visual crear targetes i moure-les entre les diferents llistes.

4.1 Diagrama de Gant

Per tal de gestionar el projecte també s'ha empleat un diagrama de Gant elaborat amb excel com podem veure a la figura 2. En aquest diagrama contemplarem les diverses tasques i subtasques per tal de fer una previsió del treball a realitzar, el temps aproximat que trigarem per cada tasca, això ens permetrà fer una aproximació del temps que ens comportara el projecte d'aquesta forma podrem determinar la viabilitat.

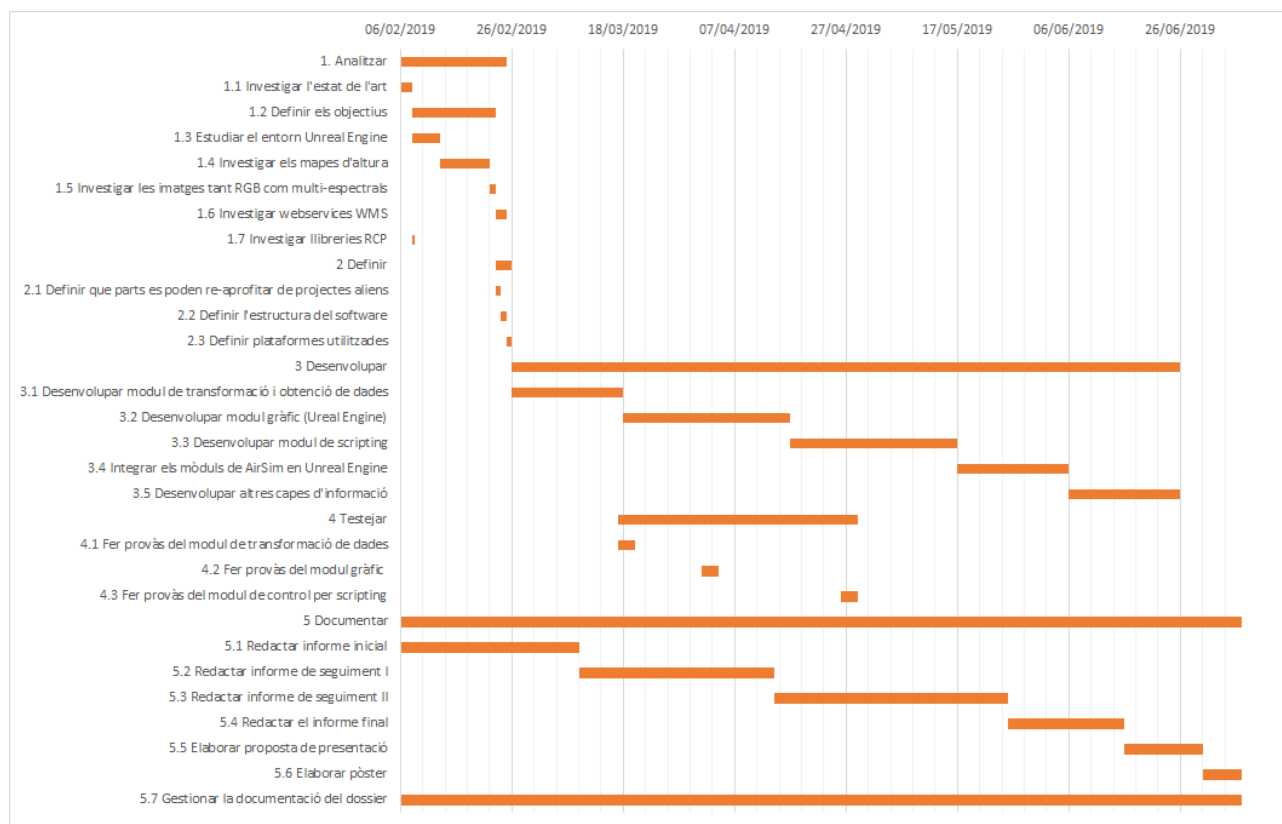


Fig. 2: Diagrama de Gant

5 ESTRUCTURA D'ALTRES PROJECTES

Per tal de determinar la estructura del nostre projecte s'han estudiat varies alternatives vistes al apartat 3 en les quals es pot veure com estan organitzats altres projectes similars. En aquest apartat veurem l'estructura de diferents projectes amb l'objectiu de decidir l'estructura del projecte y diferents llibreries.

5.1 AirSim

AirSim esta compostat per múltiples mòduls escrits en diversos llenguatges com es pot veure a continuació:

- AirLib (C++): Modul per a Unreal Engine que proporciona les classes bàsiques per comunicar-se mitjançant el protocol RCP i control dels vehicles simulats.
- DroneServer (C++): Servidor per rebre ordres de Dron mitjançant RCP
- DroneShell (C++): Client de consola per enviar ordres al modul de dron.
- PythonClient (Python): Client que envia ordres mitjançant RCP, també incorpora codi per el tractament, segmentació d'imatges, etc.
- SGM (C++): Codi per tractar imatges i generar les les vistes segmentada i stereo.
- Unity (C# i C++): Demostració en el motor gràfic Unity, incorpora una serie de mòduls per Unity per visualitzar la informació de AirSim.
- Unreal Engine (C++): Demostració en el motor gràfic Unreal Engine, incorpora una serie de mòduls per Unreal Engine per visualitzar la informació de AirSim.

5.2 Carla SIMULATOR

Carla SIMULATOR esta compostat per múltiples mòduls com podem veure en la figura 3 escrits en diversos llenguatges com es pot veure a continuació:

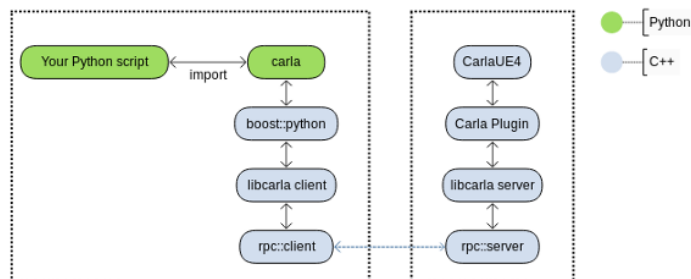


Fig. 3: Relació entre els mòduls de Carla

- LibCarla (C++): Llibreria principal de Carla
- Unreal (C++): Motor gràfic amb el plugin de carla, que incorpora totes les funcionalitats afegides a Unreal.
- PythonAPI (Python): API que ens permetrà enviar comandes al module de Carla que fa de servidor, aquesta api serveix per crear scripts propis.

5.3 Estructura escollida

Analitzant varis projectes de característiques similar s'ha decidit per una estructura pròpia com podem veure en la figura 4 podent reutilitzar alguns dels petits mòduls open-source de altres projectes. S'ha pres aquesta decisió degut a que els altres projectes es basen en crear terrenys predefinitos mitjançant l'entorn de Unreal, contrari a la finalitat d'aquest projecte en el qual es volen elaborar de forma automàtica terrenys a partir de dades reals.

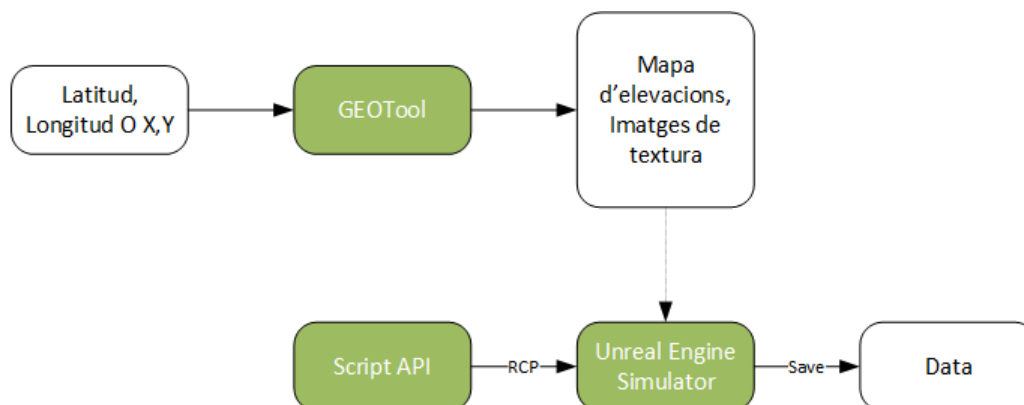


Fig. 4: Estructura DronSimulator

En el nostre projecte constara d'aquests mòduls:

- GEOTool (Python): En aquest modul es centrara en donar les eines necessàries per la obtenció y adaptació de les dades amb l'objectiu de poder importar-les en el entorn de Unreal Engine, també pot incorporarà mòduls de processament de imatge per tal d'afegir informació addicional a les textures o fer tractaments de millora d'imatges a partir de xarxes neurals.
- ScriptAPI (Python): API que ens permetrà fer scripts en Python per tal de controlar la càmera en el entorn gràfic. D'aquesta forma també ens permetrà l'obtenció d'imatges en l'entorn Unreal Engine.
- Unreal Engine Simulator (C++): Incorporara tot el motor gràfic i plugins necessaris per la comunicació amb els fitxers generats amb el mòdul GEOTool, visualització de dades, etc.

REFERÈNCIES

- [1] AirSim - <https://github.com/Microsoft/AirSim> [19/02/2019]
- [2] Carla SIMULATOR - <http://carla.org> [19/02/2019]
- [3] Agile software development
https://en.wikipedia.org/wiki/Agile_software_development [19/02/2019]

[4] Kanban

<https://www.iebschool.com/blog/metodologia-kanban-agile-scrum/> [19/02/2019]

[5] Trello - <https://trello.com/> [19/02/2019]