

TFG – Arquitectura de computadors i sistemes operatius

FITA#01: Preparació de l'entorn de desenvolupament

Gestió del projecte a Github:

Branca del repositori:

<https://github.com/UOC-Assignments/uoc.tfg.jbericat/tree/FITA%2301>

Dashboard de seguiment de les tasques associades a la fita:

<https://github.com/UOC-Assignments/uoc.tfg.jbericat/projects/3>

Estudiant: Jordi Bericat Ruz

Professor col·laborador: Daniel Rivas Barragan

Semestre: Tardor 2021/22 (Aula 1)

Versió: ESBORRANY v6

Índex

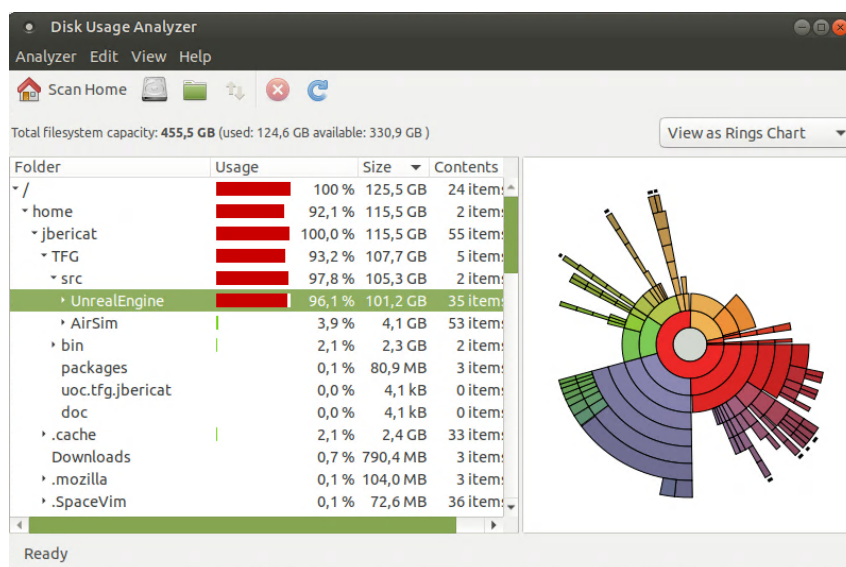
1 - Preparació de l'entorn de desenvolupament	1
1.1 – Anàlisi de pre-requisits	1
1.2 – Característiques del maquinari	3
1.3 – Preparació del programari	5
1.3.1 - Selecció i instal·lació del sistema operatiu (Linux Workstation).....	5
1.3.2 – Estructura de directoris del projecte	5
1.3.3 - Instal·lació i configuració dels paquets de software	7
1.3.3.1 - Paquets base i dependències	7
1.3.3.2 - Controladors de dispositiu	9
1.3.3.3 – Instal·lació i configuració de l'entorn Python	10
1.3.3.4 – IDE: "Visual Studio Code"	11
1.3.3.4 – Motor gràfic: Unreal Engine	11
1.3.3.5 – Plataforma de simulació: AirSim (A erial I nformatics and R obotics S imulation)	12

1 - Preparació de l'entorn de desenvolupament

1.1 – Anàlisi de pre-requisits

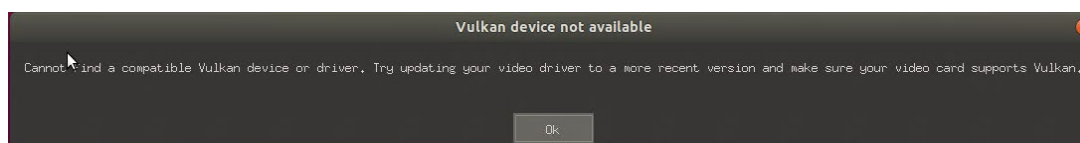
a) Capacitat i velocitat d'accés dels dispositius d'emmagatzematge

- Per a descarregar totes les fonts del motor gràfic “*Unreal Engine*” calen 110Gb d'espai lliure, tot i que serà necessari reservar espai per a compilar els arxius binaris.
- Per a descarregar les fonts del simulador “*AirSim*” caldran 5gb d'espai lliure.
- Per a la resta de paquets seran necessaris gairebé 10Gb.
- Serà requerit disposar d'espai lliure addicional per a generar arxius binaris i obtenir dades del simulador.
- Es recomana un disc SSD dedicat amb un alt rati de transferència de dades.



b) Co-processor gràfic

- El co-processor gràfic o GPU ha de ser compatible amb la versió 1.2 del driver *Vulkan* de codi obert:



c) Sistema Operatiu

- Segons requeriments indicats a la documentació de la plataforma de simulació *AirSim* els sistemes operatius en els quals s'ha verificat el seu correcte funcionament són els següents:
 - Windows 10
 - MAC OSX Catalina (10.5)
 - Ubuntu 18.04 LTS

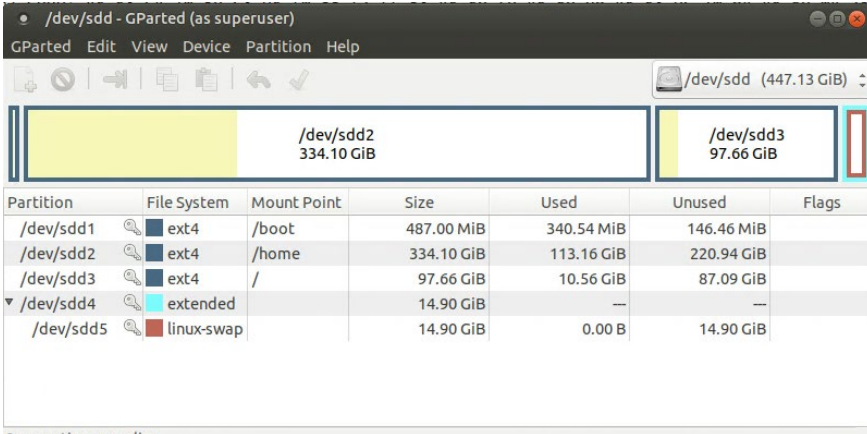
1.2 – Característiques del maquinari

a) Sistema

- Processador: Intel Core i5 CPU 760 @260Ghz (x 4)
- Memòria Ram RAM: 16Gb - 1333Mhz

b) Emmagatzematge

- Dispositiu SDD dedicat: 480Gb (Kingston A400)
- Taula de particions personalitzada:
 - /boot → 512 Mb (Partició d'arrencada del Sistema)
 - /home → 380 Gb (Partició dedicada per als axius del projecte)
 - / → 100 Gb (Arxiu del Sistema operatiu)
 - swap → 16 Gb (Partició d'intercanvi)



Partition	File System	Mount Point	Size	Used	Unused	Flags
/dev/sdd1	ext4	/boot	487.00 MiB	340.54 MiB	146.46 MiB	
/dev/sdd2	ext4	/home	334.10 GiB	113.16 GiB	220.94 GiB	
/dev/sdd3	ext4	/	97.66 GiB	10.56 GiB	87.09 GiB	
▼ /dev/sdd4	extended		14.90 GiB	—	—	
/dev/sdd5	linux-swap		14.90 GiB	0.00 B	14.90 GiB	

0 operations pending

c) Unitat GPU

- GPU NVIDIA Geforce GTX 1650 (4Gb GDDR6, 896 Nuclis, Arquitectura “Turing”)

S'ha procedit a obtenir un dispositiu gràfic que compleixi amb els requeriments especificats a l'apartat 1 (això és; compatible amb el controlador gràfic de codi obert *Vulkan*, així com pels diferents sistemes operatius suportats per *AirSim*):

Vulkan

[Devices](#)
[Reports](#)
[Properties](#)
[Features](#)
[Extensions](#)
[Formats](#)
[Memory](#)
[Surface](#)
[Instance](#)
[Download](#)
[About](#)

gpuinfo.org

Listing reports for GeForce GTX 1650 on Linux

[All platforms](#)
[Windows](#)
[Linux](#)
[Android](#)
[Macos](#)
[Ios](#)

Type to filter

Type to filter

Type to filter

Type to filter

Type to filter

Type to filter

Type to filter

Type to filter

id	Device	Driver	Api	Vendor	Type	OS	Version	Platform	compare
10918	GeForce GTX 1650	460.56.0.0	1.2.155	NVIDIA	discrete	ubuntu	20.10	x86_64	<input type="checkbox"/>
9409	GeForce GTX 1650	450.66.0.0	1.2.133	NVIDIA	discrete	arch	unknown	x86_64	<input type="checkbox"/>
8658	GeForce GTX 1650	440.82.0.0	1.1.119	NVIDIA	discrete	manjaro	unknown	x86_64	<input type="checkbox"/>
8084	GeForce GTX 1650	435.21.0.0	1.1.109	NVIDIA	discrete	ubuntu	18.04	x86_64	<input type="checkbox"/>
7889	GeForce GTX 1650	440.44.0.0	1.1.119	NVIDIA	discrete	arch	unknown	x86_64	<input type="checkbox"/>
7129	GeForce GTX 1650	435.21.0.0	1.1.109	NVIDIA	discrete	manjaro	unknown	x86_64	<input type="checkbox"/>

Showing 1 to 6 of 6 entries (filtered from 12,302 total entries)

Previous

1

Next

Vulkan Hardware Database - © 2016-2021 by Sascha Willems

Vulkan

and the Vulkan logo are trademarks of the Khronos Group Inc.

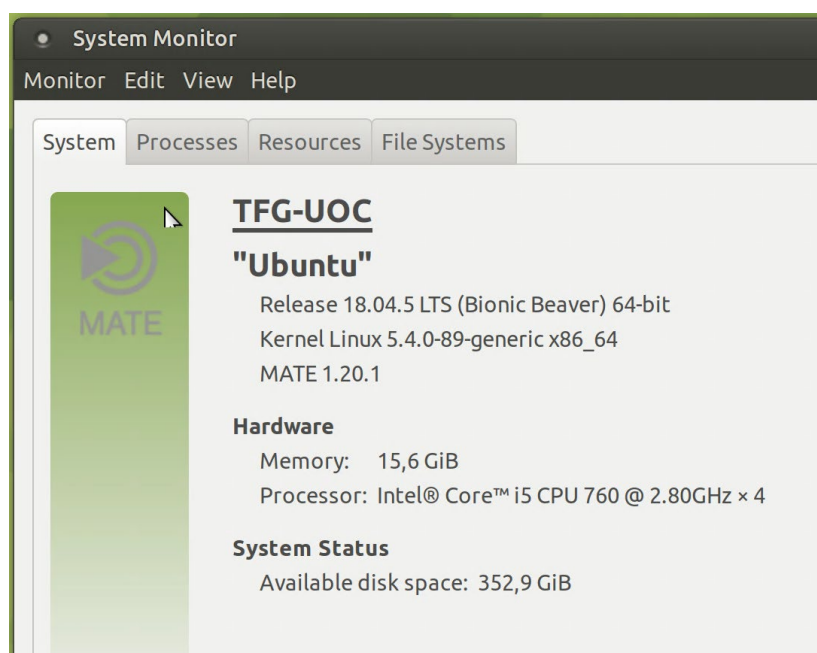
Privacy policy

The data presented is licensed under the [Creative Commons Attribution 4.0 International License](#).

1.3 – Preparació del programari

1.3.1 - Selecció i instal·lació del sistema operatiu (Linux Workstation)

Ubuntu-MATE 18.04.5 LTS¹



1.3.2 – Estructura de directoris del projecte

- **Project root folder:**
 - /home/jbericat/Workspaces/uoc.tfg.jbericat/
- **root sub-folders:**
 - bin/ → .gitignore
 - doc/
 - src/
 - AirSim/ → .gitignore
 - UnrealEngine/ → .gitignore
 - UnrealProjects/ → .gitignore

¹ <https://cdimages.ubuntu.com/ubuntu-mate/releases/18.04.5/release/ubuntu-mate-18.04.5-desktop-amd64.iso>

- `usr/` → `.gitignore`
- `usr/archive/`
- `usr/lib/`
- `usr/packages/`

1.3.3 - Instal·lació i configuració dels paquets de software

1.3.3.1 - Paquets base i dependències

```
#1 - Basic administration & helper tools

sudo apt-get update
sudo apt-get install openssh-server curl gedit git gnome-disk-utility progress guake
system-config-samba shutter vim build-essential gdb

#2 - Vim customization (https://spacevim.org/quick-start-guide/#linux-and-macos)
curl -sLf https://spacevim.org/install.sh | bash

#3 - xRDP Server

#3.1 - Installation (https://www.c-nergy.be/products.html)

cd ~/TFG/packages/
curl -O https://c-nergy.be/downloads/xRDP/xrdp-installer-1.2.3.zip
unzip xrdp-installer-1.2.3.zip
./xrdp-installer-1.2.3.sh

#3.2 - Configuration (https://mikelk.dk/xrdp-ubuntu-mate/, https://c-nergy.be/blog/?p=16698)

echo "mate-session" > ~/.xsession
cd /etc/xrdp

#3.2.1 - Set up TLS
openssl req -x509 -newkey rsa:2048 -nodes -keyout key.pem -out cert.pem -days 365

sudo vim /etc/xrdp/xrdp.ini

#> certificate=/etc/xrdp/cert.pem
#> key_file=/etc/xrdp/key.pem
#> security_layer=tl
#> ssl_protocols=TLSv1.1, TLSv1.2

#3.2.3 - Add firewall rule
ufw allow 3389/tcp

#3.2.4 - Add support for multiple sessions at once

sudo vim /etc/xrdp/startwm.sh

#> unset DBUS_SESSION_BUS_ADDRESS
#> unset XDG_RUNTIME_DIR

#3.2.5 - Fix some xrdp bugs
sudo chown -R jbericat ~/.cache/dconf/

#3.2.6 - Restart xrdp service
sudo service xrdp restart

#4 - Git profile config

#4.1 - Ajustos generals

git config --global user.name "jbericat"
git config --global user.email jbericat@uoc.edu
git config --global credential.helper cache
git config http.postBuffer 524288000
```

```
# 4.2. Establím l'arrel de la estructura de directoris on crearem el repositori:
```

```
mkdir ~/Workspaces/ && cd ~/Workspaces/
```

```
# 4.3. Clonem el repositori que ja hem creat anteriorment amb la interfície web de github.com
```

```
git clone https://github.com/UOC-Assignments/uoc.tfg.jbericat.git
```

```
# 4.4. Creem la estructura de directoris del repositori
```

```
cd uoc.tfg.jbericat/  
mkdir doc/ bin/ src/ /AirSim/ src/UnrealEngine/ src/UnrealProjects/ usr/ usr/archive/  
usr/lib/ usr/packages/
```

```
# 4.5. Crear l'arxiu .gitignore i afegir el següent:
```

```
vim .gitignore
```

```
#> # Ignore directories:
```

```
#>
```

```
#> usr/
```

```
#> bin/
```

```
#> src/AirSim
```

```
#> src/UnrealEngine
```

```
#> src/UnrealProjects
```

```
+++++(links plantilles c++ i python de github)
```

```
git add .
```

```
git commit
```

```
git push
```

```
# 4.6. un cop hem establert la estructura de directoris i regles "gitIgnore" al "mas-  
ter" branch, podem crear-ne un de nou per a treballar (en farem un per a cada fita del  
# projecte).
```

```
git checkout -b fita_#001
```

```
#5. Mounting external disks and enabling samba shared folders
```

```
#5.1 - Auto-mount secondary / auxilial HDD drive
```

```
sudo vim /etc/fstab
```

```
#> /dev/disk/by-uuid/08F460A74773ACBF /mnt/ARCHIVE auto nosuid,nodev,nofail,x-gvfs-  
show,x-gvfs-name=ARCHIVE,x-gvfs-icon=ARCHIVE,x-gvfs-symbolic-icon=ARCHIVE 0 0
```

```
#5.2 - Enabling samba
```

```
sudo smbpassword jbericat
```

```
sudo vim /etc/samba/smb.conf
```

```
[TFG-Shared]
```

```
comment = TFG Shared Files
```

```
path = /mnt/ARCHIVE/Shared
```

```
writable = yes
```

```
;
```

```
browseable = yes
```

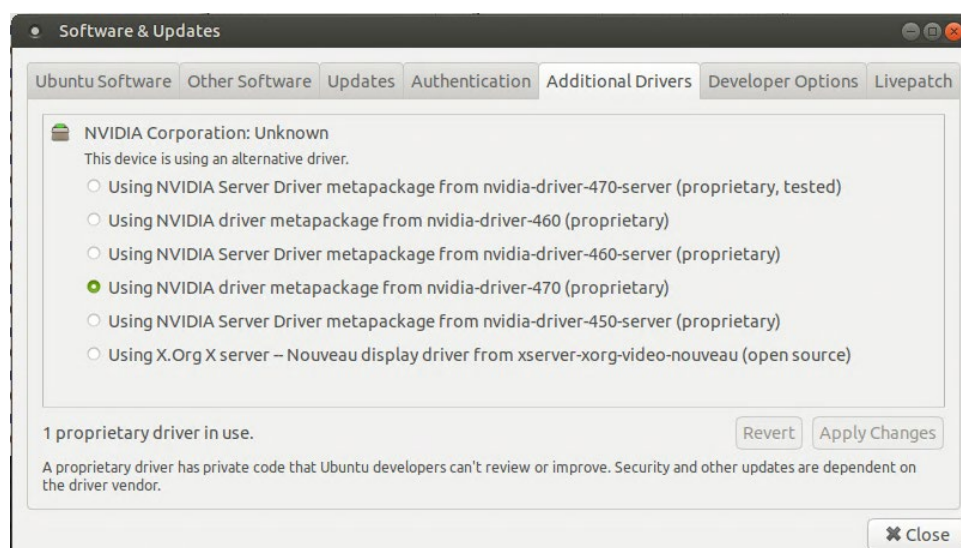
```
valid users = jbericat
```

```
sudo service smbd restart
```

```
sudo service nmbd restart
```

1.3.3.2 - Controladors de dispositiu

A Ubuntu 18.04.LTS s'instal·len per defecte els drivers "open Source" (Nouveau xServer display driver). Per a poder utilitzar AirSim, s'han d'utilitzar els drivers de codi privatiu que proporciona **NVIDIA**:



1.3.3.3 – Instal·lació i configuració de l'entorn Python²

```
# 1. Descarregar i instal·lar el gestor anaconda (la versió més recent):

curl -O https://repo.anaconda.com/archive/Anaconda3-5.3.1-Linux-x86_64.sh
chmod u+x Anaconda3-5.3.1-Linux-x86_64.sh
./Anaconda3-5.3.1-Linux-x86_64.sh

# 2. Afegir path al .bashrc i eliminar el codi afegit per l'instal·lador de
anaconda
sudo vim

export PATH="/home/jbericat/Workspaces/uoc.tfg.jbericat/src/UnrealEngine/En-
gine/Binaries/Linux:/home/jbericat/.local/bin:/home/jbericat/ana-
conda3/bin:$PATH"

# 3. Crear l'entorn (això triga!):

conda create -n condapy373 python=3.7.3 anaconda

# 4. Inicialitzar conda amb la nostra shell (bash):

conda init bash

# 5. Activar l'entorn:

conda activate condapy373

# 6. Instal·lar els packages pip de python

pip install msgpack-rpc-python #(*)
pip install airsims #(**)

# (*) Això no sé si ha de ser amb sudo
# (**) Si al instal·lar el package d'airsims es retorna el següent error:
# "ERROR: Could not build wheels for opencv-python which use PEP 517 and can-
not be installed directly", aleshores s'ha d'instal·lar a mà el següent pa-
quet:

pip install --upgrade pip setuptools wheel

# i instal·lar les dependències que falten:

pip install airsims

# 7. Arrencar el simulador al EU4Editor (play) o bé amb els binaris compilats

# 8. Executar el codi python següent

python ~/Workspaces/uoc.tfg.jbericat/src/AirSim/PythonClient/multirotor/he-
llo_drone.py
```

² <https://conda.io/projects/conda/en/latest/user-guide/concepts/environments.html>
<https://www.usessionbuddy.com/post/how-to-install-python-3.5-python-3.7-and-anaconda-on-centos/>
<https://stackoverflow.com/questions/57518050/conda-install-and-update-do-not-work-also-solving-environment-get-errors#57597032>

1.3.3.4 – IDE: "Visual Studio Code"³

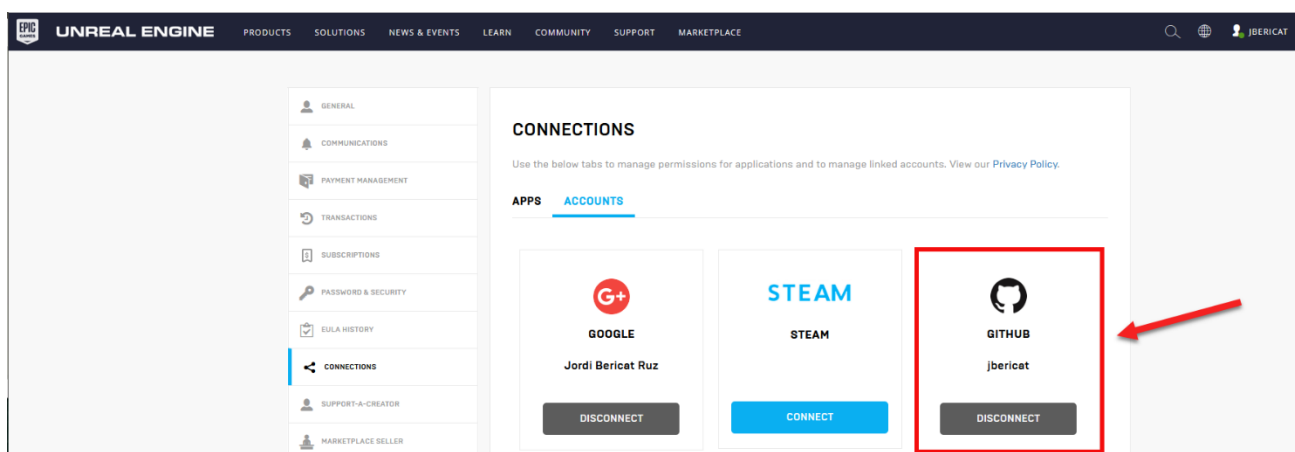
Per a la realització d'aquest projecte s'utilitzarà el IDE de codi obert "*Visual Studio Code*" com a entorn de desenvolupament de software. Per a procedir a la seva instal·lació cal executar les comandes següents al terminal:

```
#Download the package from source
cd ~/TFG/packages/
curl -O https://az764295.vo.msecnd.net/sta-
ble/6cba118ac49a1b88332f312a8f67186f7f3c1643/code_1.61.2-1634656828_amd64.deb

#Manual installation
sudo dpkg -i code_1.61.2-1634656828_amd64.deb
```

1.3.3.4 – Motor gràfic: Unreal Engine

Unreal Engine és el motor gràfic de codi obert que utilitza la plataforma *AirSim* per a renderitzar els entorns virtuals en 3D a temps real, que a més inclou un potent editor (**UE4editor**) que ens permetrà adaptar l'entorn simulat a les necessitats del projecte. Per a poder accedir al repositori mitjançant GitHub, cal registrar-se al web www.unrealengine.com i enllaçar-hi un compte de GitHub⁴:



<https://stackoverflow.com/questions/53250933/conda-takes-20-minutes-to-solve-environment-when-package-is-already-installed>

<https://stackoverflow.com/questions/63732353/error-could-not-build-wheels-for-opencv-python-which-use-pep-517-and-cannot-be>

³ <https://snapcraft.io/code>

⁴ <https://www.unrealengine.com/en-US/ue4-on-github>

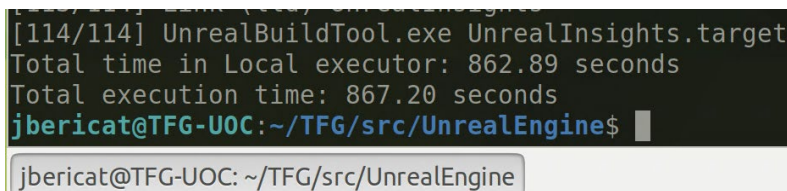
Un cop vinculats els comptes de *GitHub* i *UnrealEngine (EpicGames)*, ja es pot procedir a clonar el repositori a la carpeta de treball corresponent del projecte, tot tenint cura d'instal·lar la versió compatible amb *AirSim (4.25)*⁵:

```
# go to the folder where you clone GitHub projects
cd ~/Workspaces/uoc.tfg.jbericat/src/

# Cloning the repo
git clone -b 4.25 https://jbericat@github.com/EpicGames/UnrealEngine.git
cd UnrealEngine
./Setup.sh
./GenerateProjectFiles.sh
make

# Making life easier :)
echo "export PATH=~/.Workspaces/uoc.tfg.jbericat/src/UnrealEngine/Engine/Binaries/Linux/:$PATH" >> ~/.bashrc
```

Verifiquem que les fonts s'han descarregat i compilat correctament



```
[114/114] UnrealBuildTool.exe UnrealInsights.target
Total time in Local executor: 862.89 seconds
Total execution time: 867.20 seconds
jbericat@TFG-UOC:~/TFG/src/UnrealEngine$
```

jbericat@TFG-UOC: ~/TFG/src/UnrealEngine

1.3.3.5 – Plataforma de simulació: AirSim (Aerial Informatics and Robotics Simulation)⁶

AirSim és la plataforma sobre la qual desenvoluparem la PoC d'aquest projecte i consisteix en un simulador de vehicles no tripulats en codi obert desenvolupat per Microsoft Research sobre el framework de gràfics 3D “Unreal Engine” i orientat a la recerca en intel·ligència artificial i visió per computador. Es pot obtenir del repositori GitHub corresponent:

```
# go to the folder where you clone GitHub projects
cd ~/TFG-root/src/
# Clone from github repository
git clone https://github.com/Microsoft/AirSim.git
# Setup & build
cd AirSim
./setup.sh
./build.sh
```

⁵ En el moment de clonar el repositori, la revisió actual de UE4.24 era la 4.2.x

⁶ https://microsoft.github.io/AirSim/build_linux/

```
=====
AirSim plugin is built! Here's how to build Unreal project.
=====
If you are using Blocks environment, its already updated.
If you are using your own environment, update plugin using,
rsync -a --delete Unreal/Plugins path/to/MyUnrealProject

For help see:
https://github.com/Microsoft/AirSim/blob/master/docs/build_linux.md
=====
jbericat@TFG-UOC:~/TFG/src/AirSim$
```

jbericat@TFG-UOC: ~/TFG/src/AirSim

jbericat@TFG-UOC: ~/TFG/src