

TFG – Arquitectura de computadors i sistemes operatius

Fita #001: Preparació de l'entorn de desenvolupament.

Estudiant: Jordi Bericat Ruz

Professor col·laborador: Daniel Rivas Barragan

Semestre: Tardor 2021/22 (Aula 1)

Versió: ESBORRANY v3

Índex

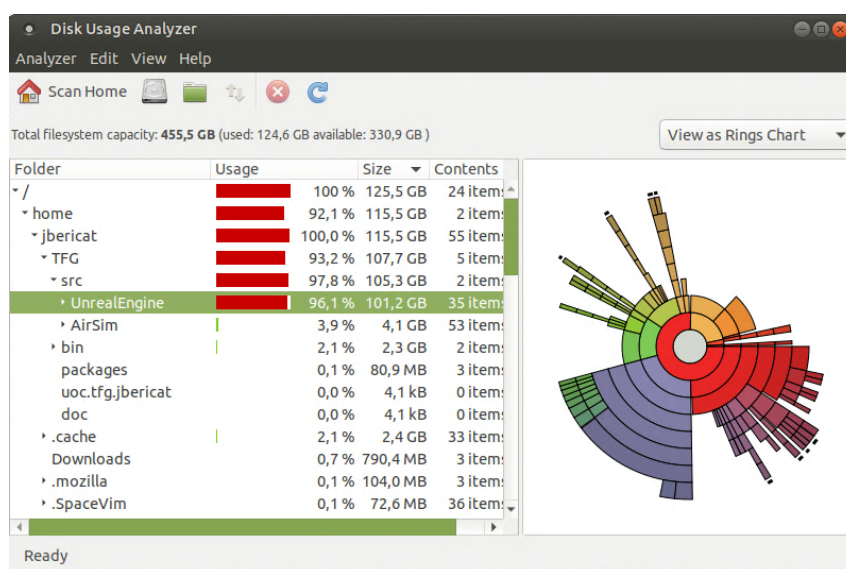
1 - Preparació de l'entorn de desenvolupament.....	1
1.1 – Anàlisi de pre-requisits	1
1.2 – Característiques del maquinari	3
1.3 – Preparació del programari	5
1.3.1 - Selecció i instal·lació del sistema operatiu (Linux Workstation).....	5
1.3.2 – Estructura de directoris del projecte	5
1.3.3 - Instal·lació i configuració dels paquets de software	6
1.3.3.1 - Paquets base i dependències	6
1.3.3.2 - Controladors de dispositiu	7
1.3.3.3 - Visual Studio Code.....	8
1.3.3.4 - Unreal Engine	8
1.3.3.5 - AirSim (A erial I nformatics and R obotics S imulation)	9

1 - Preparació de l'entorn de desenvolupament

1.1 – Anàlisi de pre-requisits

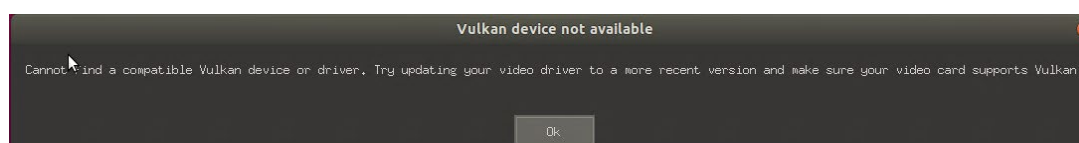
a) Capacitat i velocitat d'accés dels dispositius d'emmagatzematge

- 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

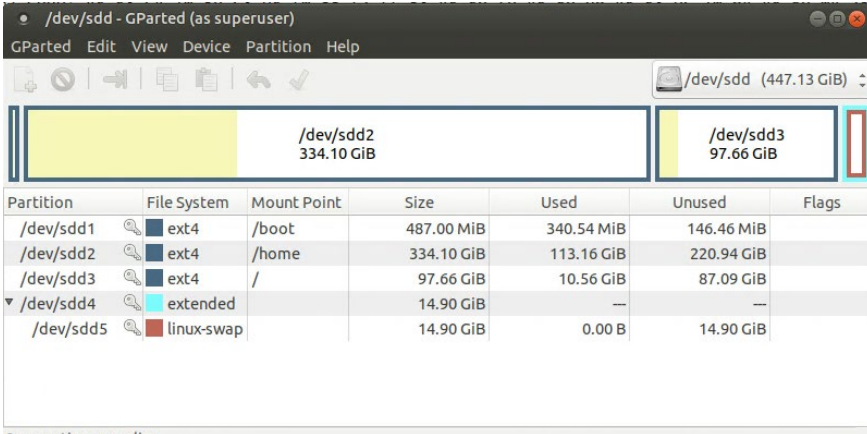
1.2 – Característiques del maquinari

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/TFG_root/
- **root sub-folders:**
 - bin/ → .gitignore
 - doc/ →
 - packages/ → .gitignore
 - src/
 - AirSim/ → .gitignore
 - UnrealEngine/ → .gitignore
 - UnrealProjects/ → .gitignore
 - uoc.tfg.jbericat/

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

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

#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=tls
#> 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
git config --global user.name "jbericat"
git config --global user.email jbericat@uoc.edu
git config --global credential.helper cache
```



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

#6.1 - 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

#6.2 - Enabling samba

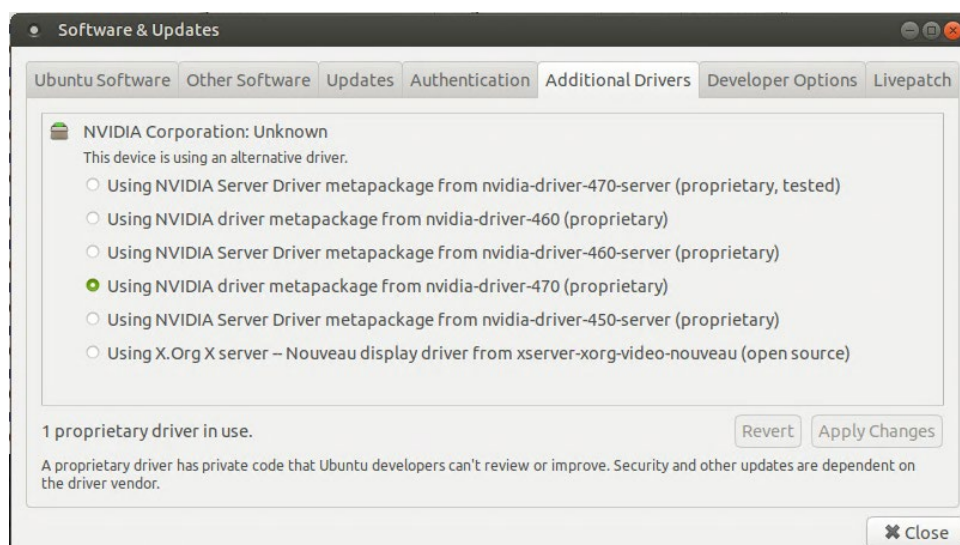
sudo smbpasswd jbericat
sudo vim /etc/samba/smb.conf

[TFG-Shared]
    comment = TFG Shared Files
    path = /mnt/ARCHIVE/Shared
    writeable = 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 - Visual Studio Code²

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 - 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 (**UE4ditor**) 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.epicgames.com i seguir aquests passos:

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

Un cop vinculats els comptes de *GitHub* i *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):

```
# go to the folder where you clone GitHub projects
git clone -b 4.25 https://jbericat@github.com/EpicGames/UnrealEngine.git
cd UnrealEngine
./Setup.sh
./GenerateProjectFiles.sh
make
```

² <https://snapcraft.io/code>

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 - AirSim (Aerial Informatics and Robotics Simulation)³

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
```

```
=====  
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
```

³ https://microsoft.github.io/AirSim/build_linux/