3D MILKY WAY SIMULATION

User Manual

Release 1.0

Design & Development: P. D'Andrea

Supervision: G. Longo, M. Brescia

University Federico II of Napoli
INAF Astronomical Observatory of Capodimonte

Index

1 Introduction

The present document is the user manual of the first release of 3D Milky Way Simulation. The application graphics technology is based on the Unity[1] engine.
The simulation is a 3D virtualization of the Milky Way galaxy, including main astronomical subjects and based on realistic assumptions on its structure.
The simulation has the scope to furnish an efficient tool to represent a physically valid model of the galaxy, for public outreach as well as to visualize and project external data gathered from real observations.
The Unity platform is a very powerful graphics engine, born to develop professional videogames and virtualized scenarios.

2 System requirements

Operating system: Windows XP, Mac OS X 10.6, Ubuntu 10.10 and later versions;

Graphics card: any card with DX9 shade model 2.0 capabilities and, at least, 512MB of memory;

CPU: any (recent) CPU with SSE2 instruction set support;

RAM: 4 GB or more;

Disk space: 1 GB or more of available space;

3 Installing and running the application

3.1 Installation

The application is enclosed into a standalone package. The only step to take in the installation process is to copy the main executable and data directory to the local system. According to the User's platform:

Windows The main executable MW.exe and the folder MW_Data must be copied into the same directory.

Mac OS The only file to copy is the app bundle MW.app.

Linux The main executable MW.x86/x64 and the folder MW_Data must be copied into the same directory.

3.2 Running the application

After the installation step, according to the User's platform, to run the application on:

Windows double-click MW.exe;

[1] http://unity3d.com/

Mac OS double-click MW.app;

Linux open a console session, change directory to the one containing the application main executable and Data directory:

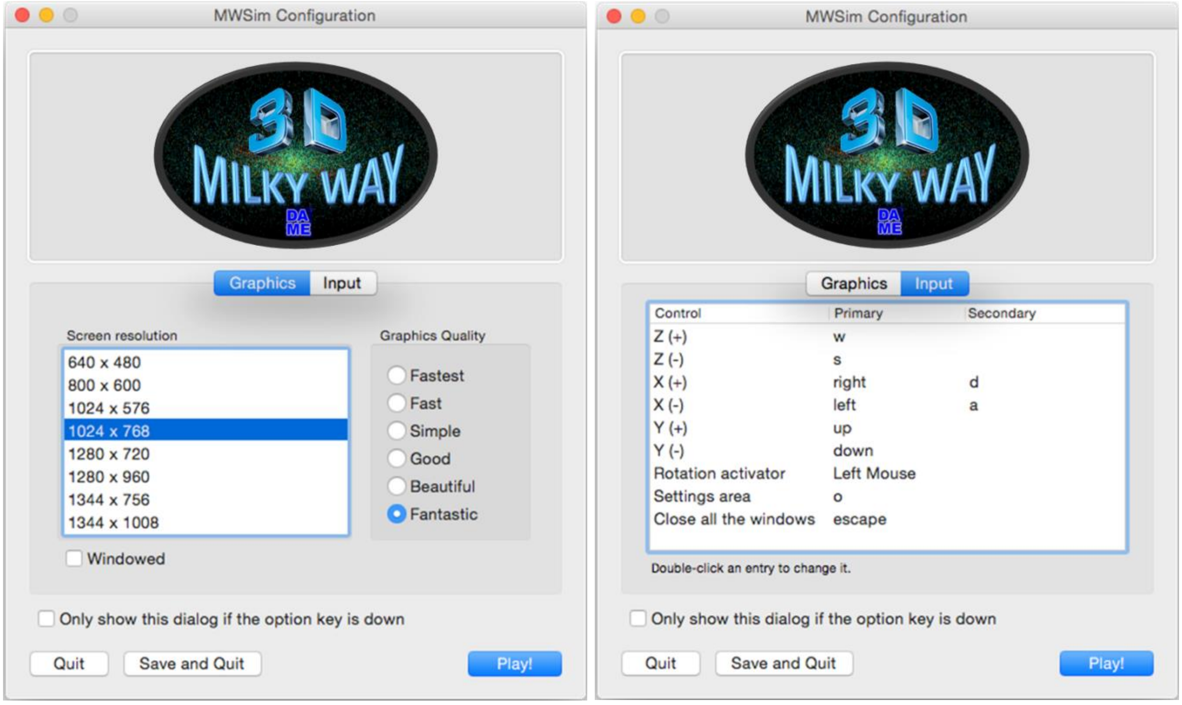
```
$ cd MW/
```

and launch MWSim.x86/64 running the following command:

```
$ . /MW. x64
```

When the application is executed, the User is prompted by the "MW Configuration" window (Figure 1) with a set of options divided in 2 tabs "Graphics" and "Input".

Once all the options are set, clicking the "Play!" button starts the simulation.



Screen resolution

- 640 x 480
- 800 x 600
- 1024 x 576
- 1024 x 768
- 1280 x 720
- 1280 x 960
- 1344 x 756
- 1344 x 1008

□ Windowed

Graphics Quality

○ Fastest
○ Fast
○ Simple
○ Good
○ Beautiful
● Fantastic

□ Only show this dialog if the option key is down

Control	Primary	Secondary
Z (+)	w	
Z (-)	s	
X (+)	right	d
X (-)	left	a
Y (+)	up	
Y (-)	down	
Rotation activator	Left Mouse	
Settings area	o	
Close all the windows	escape	

Double-click an entry to change it.

□ Only show this dialog if the option key is down

Figure 1 – (left) Graphics settings, (right) Input settings

Graphics options

Screen resolution: lowering the resolution can be helpful on older hardware to run the simulation while keeping a good frame rate.

Graphics quality: controlling the graphics quality level can, for the same reason as lowering the resolution, be helpful on older hardware.

Windowed: if this box is left unchecked the application is executed in full screen mode.

Input options

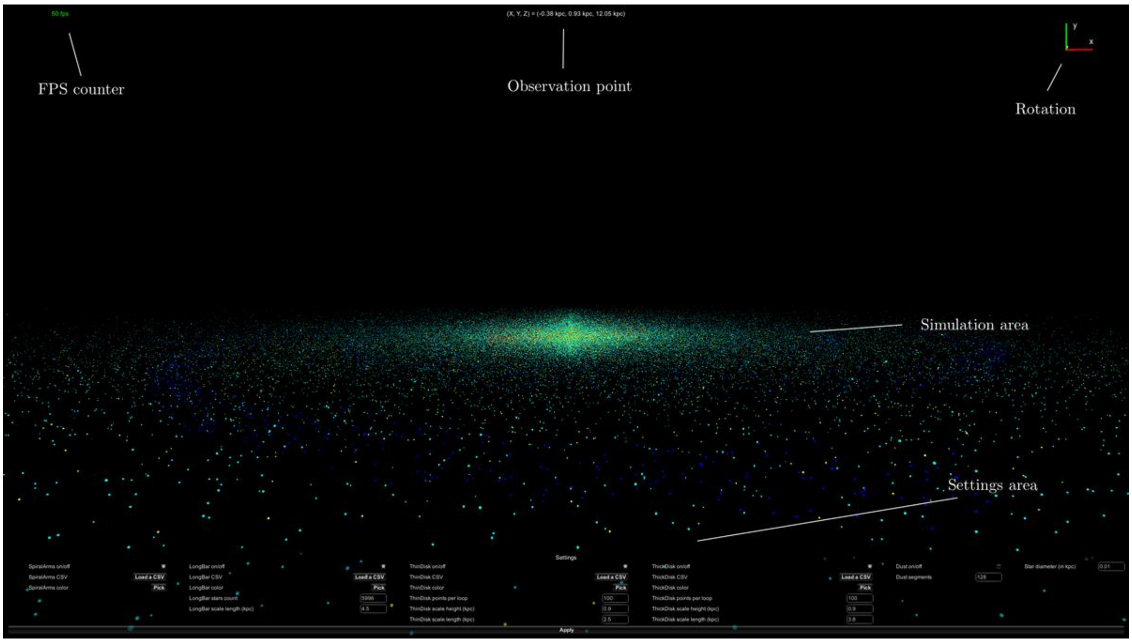
By double-clicking the entries of the table contained in this tab (Figure 1 right), the User can customize the mouse and keyboard key mappings. In the table, the value of the Control column specifies, choosing from the following list, the function assigned to the keys in the Primary and Secondary columns:

X (+/-): move the observation point left (-) and right (+);
Y (+/-): move the observation point down (-) and up (+);
Z (+/-): move the observation point backwards (-) and forwards (+).

Rotation activator: the key to hold to enable the observation point rotation (that is performed using the mouse).

Settings area: hide/show the settings area.

The input methods and the default key mappings are fully described in Sect.5.



50 fps

(X, Y, Z) = (-0.38 kpc, 0.93 kpc, 12.05 kpc)

Settings

SpiralArms on/off ■
SpiralArms CSV Load a CSV
SpiralArms color Pick

LongBar on/off ■
LongBar CSV Load a CSV
LongBar color Pick
LongBar stars count 5996
LongBar scale length (kpc) 4.5

ThinDisk on/off ■
ThinDisk CSV Load a CSV
ThinDisk color Pick
ThinDisk points per loop 100
ThinDisk scale height (kpc) 0.9
ThinDisk scale length (kpc) 2.5

ThickDisk on/off ■
ThickDisk CSV Load a CSV
ThickDisk color Pick
ThickDisk points per loop 100
ThickDisk scale height (kpc) 0.9
ThickDisk scale length (kpc) 3.8

Dust on/off □
Dust segments 128

Star diameter (in kpc) 0.01

Apply

Figure 2 – Main window

4 User Interface

FPS (frames per second) counter: displays the current frame rate. In case of low frame rate, lowering the graphics quality and the resolution (see Sect.3) can smoothen the observation point position and rotation changes.

Observation point: relative position of the observation point, in kpc, from the galactic center at (0; 0; 0).

Rotation: observation point rotation indicator.

Simulation area: where the simulation output is rendered.

Settings area: contains all the options the User can set to control the output of the simulation.

Sect.6 describes these options in depth.

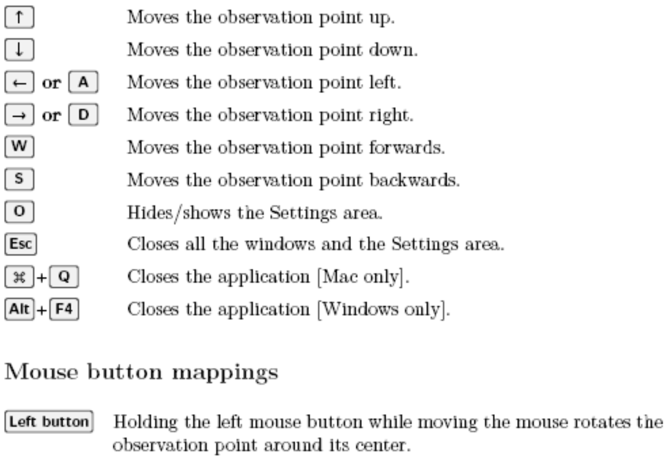
5 User input

The User has the ability, using the mouse and the keyboard:

- to control the observation point position and rotation and explore the Galaxy (in 3D);
- to control the status of the user interface elements;

To do so, some of the keys are mapped, by default, to the functions detailed in the Figure 3.

Keyboard and mouse key mappings



↑	Moves the observation point up.
↓	Moves the observation point down.
← or A	Moves the observation point left.
→ or D	Moves the observation point right.
W	Moves the observation point forwards.
S	Moves the observation point backwards.
O	Hides/shows the Settings area.
Esc	Closes all the windows and the Settings area.
⌘+Q	Closes the application [Mac only].
Alt+F4	Closes the application [Windows only].

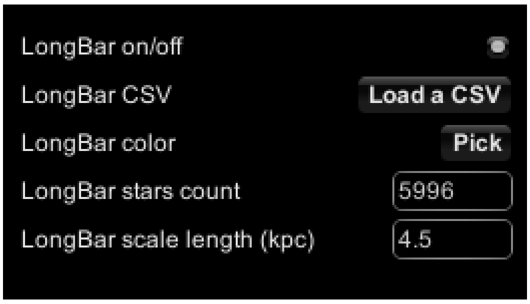
Mouse button mappings

Left button	Holding the left mouse button while moving the mouse rotates the observation point around its center.

Figure 3 – (up) keyboard keys, (bottom) mouse keys

6 Settings

In the lower part of the screen, the *"Settings"* area groups, in sections (one for each component of the galaxy), a set of options that the User can configure in order to adapt the simulation to his/her needs. The available options are explained in depth, for each section, in the following pages.



LongBar on/off ●

LongBar CSV — Load a CSV

LongBar color — Pick

LongBar stars count: 5996

LongBar scale length (kpc): 4.5

Figure 4 – "Long bar" menu section

6.1 Component settings

Some of the options in the settings area have the same function whichever component they are linked to:

- On any component a "**on/off**" checkbox, to turn off and on its rendering, is always available.
- If the component is a group of stars (as the disks, the bar and part of the spiral arms are), these settings are also available:
    - A "**color**" field to select the color used in the stars rendering process. Clicking the "Pick" button next to the color field the "Select a color" window is presented (Figure 4), here the User, using the R, G, B, A sliders, can adjust the amount of each channel (Red, Green, Blue, Alpha) to compose in the final color.
    - A "**CSV**" field to load a CSV file (formatted as specified in the Sect.7) containing a list of 3D coordinates where the stars of the selected component have to be placed. Clicking the "Load CSV" button next to the CSV field the "Select a file" window is presented (Figure 4). This window allows the User to navigate through the local filesystem folders and select the desired .csv file that is then parsed and imported into the simulation.

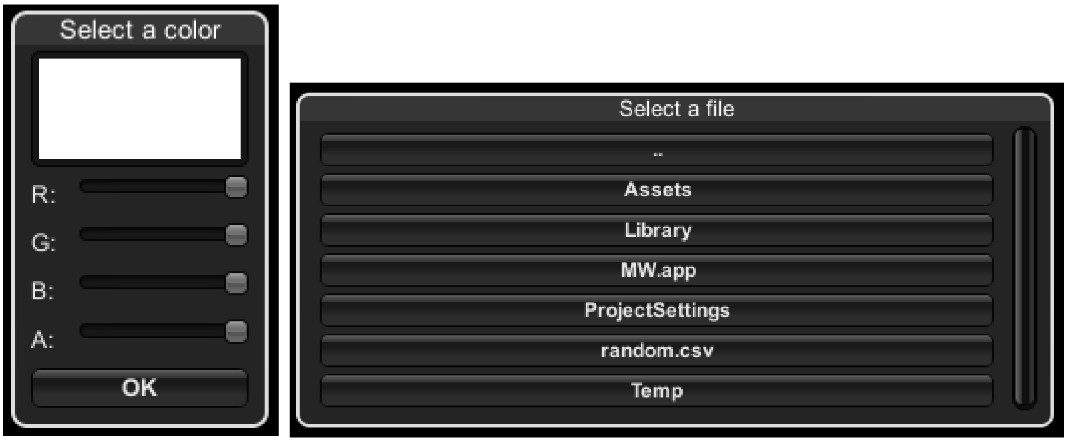


Figure 5 – (left) panel to select colors, (right) file selection panel

Long Bar

In the *Long Bar* section of the Settings window, in addition to the fields listed in the Sect.6, there are:

- A "stars count" field to set the (total) number of stars to place, arranged according to the model underlying the simulation, in the *Long Bar*;
- A "scale length" field to set the volume of cylinder in which the Long Bar is built, given the axial ratio of (1.0, 0.4, 0.4) kpc.

Thin and thick disk

Both disks, in addition to the fields listed in the Sect.6, have the following ones:

"points per loop", a multiplier used in the disk creation process, representing the maximum number of stars added in the specified disk, at each loop. To better understand how it works, given a radius *R* and an height *Z*, the populating algorithm computes the number of stars to be randomly placed at (*R;Z*) using the following:

$$\frac{\rho(R,Z)}{\rho(0,0)} \times PP_{loop}$$

where $\rho(R,Z)$ is the stellar density function and
PP_{loop} are the points per loop field value.

"scale height" (h_Z) and "scale length" (h_R) used when distributing the stars into the disks (at (*R;Z*)) according to the stellar density function:

$$\rho(R,Z) = \rho_{■} \times e^{\left(-\frac{Z-Z_{■}}{h_Z} - \frac{R-R_{■}}{h_R}\right)}$$

where $\rho_{■}$ is the stellar density in the Sun neighborhood,
(R,Z) is the point of the disk and $(R_{■},Z_{■})$ is the Sun's location in the disk.

Dust

The dust component rendering results from the alpha composition of the colors of a certain number of disks located in the galactic center and having growing radiuses ranging from ~1.5 to ~10 kpc.
The composed colors alpha channel increases exponentially from the disk center towards the outer bounds according to the:

$$\alpha(r) = e^{-\frac{r}{innerRadius}}$$

where $\alpha(r)$ is the Alpha channel at distance r, and
$innerRadius$ is the current inner radius of the disk.

The number of disks used is set by the "**dust segments**" field.

Stars

As for the stars, while their color can be set on a per component basis, their size is independent from the component to which they belong and is set changing the value of the "**star diameter**" field.

Applying the changes

The changes made to the:

- on/off fields,
- color fields,
- CSV file fields

are applied immediately, as the User changes any of their values.
Any change made to the text fields (such as the stars diameter) requires the simulation to be reloaded and has, on the other hand, to be applied clicking the **Apply** button in the settings area.

7 Data sources

CSV

As described in the Sect.6, the stars located in the disks, in the spiral arms and in the bar can be disposed loading their position from a CSV file.
A CSV (*comma-separated values*) is a file format used to store tabular data in plain-text form. The file consists of a list of entries (one per line) each of which is divided, by a comma, into data fields.

Format

The CSV files supported by the simulation software have the following structure:

[Optional row]	Field name Field name ... Field name
Entry 1	Field value Field value ... Field value
...	
Entry *n*	Field value Field value ... Field value

Valid data types

Field value: floating point numbers in the $[\pm 5.0\ x\ 10^{-324};\pm 1.7\ x\ 10^{308}]$ range. Examples of valid values are: 1, 0.12345678, 1234.5678, …

Field name: any (not empty) value that is not a field value.

Default fields

In the CSV files used by this software, each entry represents a star position into the simulation space. For this reason, for a CSV file to be valid, the following fields are mandatory:

x *x* component of the 3D coordinates for this star, in kpc.
y *y* component of the 3D coordinates for this star, in kpc.
z *z* component of the 3D coordinates for this star, in kpc.
flag Unused when the CSV is loaded for the disks and the bar. In the case of the spiral arms, this field is the index (in [0; 3]) of the arm into which the star described in the entry is placed. These in the list are also assumed as the default field names, therefore, in the following CSV files, given as samples, the entries are mapped to the same fields (x, y, z, flag), even though in the second file the names are not defined on the first line.

```
x,          y,          z,          flag
   0.826612, 0.418852, 0.938524, 1
   0.772445, 0.496801, 0.096820, 0
   0.999375, 0.727521, 0.653219, 2
   0.826612, 0.418852, 0.938524, 1
   0.772445, 0.496801, 0.096820, 0
   0.999375, 0.727521, 0.653219, 2
```

Example of a valid CSV file

oOo