Gate Documentation

Release 0

2011, Ghislain LEVEQUE

November 28, 2011

CONTENTS

CHAPTER

ONE

ABOUT

My project is to create an OpenSource and Free game inspired by Portal.

CHAPTER

TWO

STATUS

The Proof Of Concept is ready and you can get it on my github.

You'll need Panda3D to run the program. I'm using the .deb version on my Ubuntu.

CHAPTER

THREE

CONTENTS

3.1 Installation

Contents:

Try this:

```
wget http://www.panda3d.org/download/panda3d-1.7.2/panda3d1.7_1.7.2~maverick_i386.deb
dpkg -i panda3d1.7_1.7.2~maverick_i386.deb
# Check and solve dependecies if needed
git clone https://github.com/court-jus/Gate_OpenPortal.git
cd Gate_OpenPortal
ppython main.py
```

3.2 Launching the game

The simplest way is:

```
ppython main.py
```

If you want to play a specific level : just pass its name as the first argument to `main.py` (without the `.lvl` extension). For example:

```
ppython main.py level3
```

3.3 Playing the game

Your goal is to reach the exit in each level. The exit is a big white sphere. You can use "portals" to go from one point of the level to another unreachable point. Just create a portal where you want to go, create a portal near you, go into the portal and "tadam" you're on the other side.

Here are the available keys (AZERTY keyboard by default, change this in `Gate/constants.py` if needed):

- `Z, Q, S, D` : strafe and move
- `SPACE` : jump
- `LMB` : create "left" portal
- `RMB` : create "right" portal

- `E` : erase portals
- `C` : clear portal status (for debug purpose only)
- `R` : reset position
- `P` : print position
- `B` : enter pdb

3.4 Level editing

3.4.1 Edit the `.lvl` files

Each level consists of a `.lvl` file (look at `level1.lvl`, `level2.lvl`... for examples).

The structure of `.lvl` files is :

- a JSON header that contains some settings for the level :
  - `origin` : player starting point
  - `next_level` : name of the level to load when this level is won
  - `pointlights` : list of coordinates for the lights
- a line containing `-LEVEL-` that begins the level "model"
- many "slices" of ASCII chars separated by lines containing `-Z-`

The "slices" are ASCII representations of each Z level of the level world. For example, this is a floot with a hole in its center:

```
#####
#####
##  ##
#####
#####
```

Here are the ASCII chars availables :

- Normal cubes with different textures
  - `D` : a "orange" texture with black curves on it
  - `#` : a stone texture
  - `=` : a wood texture
- Cubes that cannot receive portals
  - `M` : a metallic rusty texture
- Deadly cubes
  - `L` : lava
- Friendly cubes
  - `X` : exit

3.4.2 Use the ingame editor

If you don't want to edit `.lvl` files by hand, I'm working on an inline editor. To launch it, just start the game with the `-e` flag before the name of the level:

```
ppython main.py -e mylevel
```

When in editor mode, here are the available keys (AZERTY keyboard by default, change this in `Gate/constants.py` if needed):

- `A` : fly up
- `W` : fly down
- `Z, Q, S, D` : strafe
- `R` : reset position
- `P` : print position
- `B` : enter pdb
- `L` : add a light at the current position
- `U` : undo last edit
- `F11` : save the level

Here is what you can do in editor mode :

- Copy an existing cube : clic on a cube and a copy of it will appear on the face you were looking at
- Make multiple copies of a cube : look at a cube without clicking it and use the number keys from 1 to 9 to create (1 to 9) copies of the cube. It's the same as clicking on a cube, then on its copy, then on the copy of its copy and so on
- Delete an existing cube : right-clic a cube
- Make a rectangle : look at a cube and use the `X` key to create a rectangle from this cube to where you are (the camera)
- Make a room : look at a cube and use the `Shift-X` key to create a room (non-filled parallelepiped) from this cube to where you are (the camera)

Loading an existing level

You can edit an existing level by launching the editor on it:

```
ppython main.py -e level1
```

Notes

When saving the level, the camera position in the editor is saved as the `origin` position in the `.lvl` file so it will be the player's original position.