Assignment tips

Aidan Delaney

January 26, 2017

Introduction

These notes are provided to support you in making good progress. Furthermore, students also often have difficulty in structuring project work. These notes should also provide some help with a project structure.

Installation

You will need a development environment in which to work. We provide a development environment in our labs. However, if you want to recreate this environment at home I suggest you use `https://www.virtualbox.org/` in which to virtualise a Fedora Workstation installation[1]. The setup of Fedora should be straightforward, noting that you'll probably need around 30Gb of free space.

[1] `https://getfedora.org/`

Once Fedora is installed, you'll need to ensure you have the required compilers and libraries:

```
$ su # to get superuser access
$ dnf groupinstall "C␣Development␣Tools␣and␣Libraries"
$ dnf install SDL2-devel SDL2_image-devel gcc-c++
```

I'm reasonably sure that this should install everything needed. If not, drop me an email.

First Phase

The first phase of development should concentrate on making the player sprite move up and down in addition to the current left and right. To do this you should identify where the code to "go left" and "go right" resides. You'll then have to modify this code to add "go up" and "go down" functionality. Thereafter, you will have to make a key press activate your new functionality. Between reading, writing and making mistakes, I'd allocate an entire day for the completion of this phase. That's 7 hours. If you complete this in less time, then you can progress more rapidly.

In order to expand the event subsystem you might want to look at:

`SFEvent.h` Extend this with a SFEVENT_PLAYER_UP and

`SFEvent.cpp` Extend this to translate the `SDL` key up event to an `SFEvent`.

Thereafter, you may want to look at SFApp::Event to handle your new SFEVENT_PLAYER_UP. Finally, you may want to add some "go up" functionality to the `SFAsset`.

Second Phase

Repurpose an existing sprite as a "wall" (assuming we're making a Pac-Man clone). The Alien doesn't look like a wall, but if we can use it as one, we can then replace it with something more appropriate in the near future. Initially I'd create only two "walls", each of which are an existing Alien. I'd do this because I'm interested in making sure that when I collide with a wall the sprite doesn't move into the wall. This, again, should take about a day or possibly two.

You should note that there may be an error in my collision detection code. Therefore the collisions may be off by half the width of the sprite. For the moment we just have to live with this.

Third Phase

Replace the repurposed Alien with a Wall sprite, extending `SFAsset` as necessary (1 day). Then some aliens back into the level, with a few more walls.

Fourth Phase

Add an "end of level" system. This could be some kind of collectable that ends the level. Ideally, this should then tell me my game score (I'm happy if the score is simply printed to cout.

Fifth Phase

Extend, extend, extend. Do something really exciting. Add audio, improve the smoothness of the sprite movement or add some levels. Above all, do something I've not asked for in the brief.

Conclusion

The phases 1 through 4 above should take a week of work. This amounts to 40 hours. You may find that it takes you two weeks. In which case, take your time. It's more important to understand what we're doing than to race through it.

References