Reflective Document

Due Date: 17/10/08
Assignment 3 - Reflective Document
Name: Robert Knight
Student Number: 41777903
Project Title: Super Prin Prin Bash Excess

The Pygame project just completed by myself, Super Prin Prin Bash Excess, was partially successful. The design underwent major changes through development, although it does still achieve the goals set in the initial design document. It was meant to be a mildly to very fun platformer and it remains so. The game still has many problems, and appears to be in an unfinished state if the player reaches the end of the game.

One of the issues was clearly in not using time effectively. I began working on it later than I should, and had done so with next to no knowledge of Pygame or how it was best implemented. Since I was learning Pygame as I was creating the project, my initial design did not allow for many Pygame features such as Surfaces. I later realised that it would have been far easier to use Sprite objects instead of a combination of Rects and Surfaces but because of the late start, did not have time to change this. Similarly, the main game function is lacking proper structure, to the point that it actually goes through some of the bigger loops several times within one frame. The code remains rather 'unclean' with conventions that were made in some classes lacking in others, and many numbers and algorithms simply there because they worked. Many different integers were used to represent pixel sizes and I would have found it far more difficult to use variables for all these instead of just passing in the relevant equation; it would be hard to justify many of the numbers I used. I believe I even keep three copies of the same list of dictionaries without reason but time won't let me clean it up.

A problem that I am still unsure of as I write this, is whether or not the game will be accepted as being fun by the users. Many aspects of the game that I as a developer found acceptable may not be to those playing the game for the first time. Collision detection with the player's weapon, for example, may be seen as incredibly frustrating by first time users, as where player actually hits does not match up with the image. This could also been seen as a problem with graphics.

Returning to time management, another issue was the fact that I am not very good at graphic creation, and found on many days that I was spending more time trying to draw things than code them. Looking back, I should not have included the cutscene on beginning the game, as I took a long time to draw it and the result is not very impressive. These are the sorts of things I could have been working on before even learning Pygame though, but did not.

I've found bugs for the most part to be fairly easy to fix. Sometimes I've even left some unintentional programming errors in because they worked in my favour. For example, the fading in and out during cutscenes was not my original intention but looked well enough to not fix up. There are still a few remaining bugs in the latest version. The ones identified are:
- The player can 'stick' to walls when using the short jump. I believe this is due to the block to player collision section of the main game loop being designed badly.
- The music for winning a level and losing a level plays sometimes and not others
- The boss character jerks around when reached the position they should be in
- During the boss fight, occasionally an enemy will spawn without the spawn graphic appearing
- Sounds do not play immediately. eg the jump sound is heard several milliseconds after the game displays the jump
- Much surrounding the ladders is not currently as it was supposed to be. Depending on the ladder

position it may be very difficult to get off, it may cause the player to move 'through' walls and the player may continue along two ladders as though it were one.

As previously stated, the game does not feel complete. I wanted to get at least one level in a better than just satisfactory state before moving on and so in the end the game is very short. The boss battle in the fourth level and the ending cutscene were implemented at almost last minute to attempt to address this issue.

All sounds and music was made at the last minute also. As with the graphic design, I am not very experienced in making music so most tunes are very short. To compensate I made the sound effects drown out the music a bit, hopefully not to the point of being irritating though. After putting the sounds into the game, I noticed the .zip file was 19Mb. Luckily changing the music and some of the higher resolution graphics to lossy formats reduced it down to under 5Mb.

Last minute coding means lack of testing. I have not as of writing this attempted to try the latest version on any computer systems other than my own, let alone an OS other than Windows. The version I had at the start of this week worked fine on the university computers but I've added much since then, and especially sound has the potential to have different effects on different computers.

Enemy spawn times and level design also would have required much playing to get to a state acceptable for a commercial game. At the moment, the level design doesn't show off the best of what the enemies were designed to do and seems a bit unnatural. Something that may have improved this is by making some enemies appear after certain events rather than being completely time based, so that it can be a similar experience whether the player speeds through the level or not.

Many features I proposed in the initial design document did not end up being implemented. The game lacks: joystick support, rotatable monitor support, full screen mode, high score saving and a level editor. The first three of the above would have required Pygame modules I am unfamiliar with, but high score saving would have been done simply by asking the player for a name, loading the current .txt file high score list, appending the new player, sorting the list, and then saving it back to the text file. A level editor would have taken much more time to implement, and may have even been easier to make using tkInter. Games, generally should not need a level editor, in my opinion, because the default levels should be good enough as they are.

I was slightly unsure about what exactly was needed in the design document. A list of all the classes main variables and functions were provided but may not have been needed as it should have been seen in the code anyway. Similarly, analyses of the game's enemy patterns and storyline may not be important for the client.

Overall I hope that the end result is of a satisfactory level, but the project's problems are very evident. The design goal was to be fun more than anything else, since I am a player of video games rather than a programmer of video games, it is the aspect I believe I know most about. Unfortunately too much time was spent on graphics and not enough time was spent on the assignment overall, and the result falls a little short of what was proposed. However, it is still an arcade-platformer and with a few minor tweaks would be about the standard of a freeware computer game.