CSSE1001 Assignment 3
Reflection
October 2008

Name: Nicholas Hughes
Student Number: 41785636
Project Title: PolyominOhs!

As far as how the game is to play and use, I am very satisfied with how it turned out. I implemented every feature I wanted to, and I have not discovered any bugs after much testing. I underestimated the complexity of coding my own GUI and because of this the menu system is a little less user-friendly than I would have liked, but this is my only real concern with the game. Overall, I am very happy with how everything turned out.

However, there are a few minor problems 'under the hood' of the game that I would have fixed had I had time.

The menu system and menu items are all implemented through a kind of half-object-oriented/half-not approach which stemmed from lack of planning. I didn't foresee the complexity of the underlying systems which work a GUI and so my planning was inadequate. I am not proud of this, as the rest of the game is implemented through a generally well thought-out, modular, object orientated system. The way it stands, adjusting the menu or extending it would take a lot of work. If I had more time, I would have fixed this and created an extendable and 'clean' object orientated menu system.

Some of the graphics code also suffers from the same lack of planning leading to a somewhat 'hacked together' implementation. This is only in parts though, and I think that overall the code for the visual aspect of the game is well put together. There is definitely room for improvement, though.

The algorithm for generating the polyominoes needs to know how many polyominoes it is looking to generate. This isn't a huge problem, but the algorithm really should enumerate the polyominoes as well as generate them. I am not terribly knowledgeable in that particular area of mathematics and as such I implemented the algorithm as best I could.

I have no problems with efficiency anywhere in my code. There are some parts of the display code that draw static text to the screen every frame, and the polyominoes generating algorithms takes a few seconds to run, but overall these things don't affect the game in any noticeable way.

I am happy with the quality of my code. I have used descriptive variable names nearly everywhere and there are few sections where I would say the code in confusing. I feel that this high level of readability combined with the comments I have provided for every file, class and method, as well as around code that may be hard to understand, gives a clear picture of how the game works. I am confident that any decently proficient programmer would be able to understand the code with relative ease.

I made few deviations from my initial design document. The main additions were a lot of getter methods for attributes, a function to generate a number of colours with evenly spaced hue (to colour the generated polyominoes) and some helper methods and classes. My overall design of the classes and their interaction stayed the same as I originally planned.

I didn't run into any significant problems while implementing the game. Some of the algorithms I needed to develop regarding the polyominoes (e.g. generating them, generating colours for them and finding their pivot point to rotate them correctly) were more difficult than I initially thought, but I solved the problems eventually.

Overall I am very happy with how my game turned out, when viewed from the outside and played. It runs fast and bug-free and is in my opinion fun to play, so for a game I think it achieves its purpose quite well. There are some things I would change about how the application operates inside if I had more time, but these are relativity minor on the whole.