

CSSE1001

23/05/13

Assignment 3 – Reflection Document

Student: Sean Manson

SID: 42846413

Project: GUI-based Riichi Mahjong implementation with A.I and Mahjong modules.

The goal of this assessment project was to develop an implementation of Riichi Mahjong into python using pygame to develop an intuitive and effective interface for users. As part of this, it was planned that complex AI, a fully-fledged saving/loading system and a modular back-end to the program would be incorporated, essentially making the final version of the program chock full of features and usable elements for many users. Overall, this project was partially successful in achieving these goals, but a multitude of factors resulted in several of the planned features and concepts being scrapped in favour of bug testing and increasing the efficiency of existing code. Despite this, I am quite pleased with how it turned out, with the majority of it actually exceeding my original expectations, and I certainly would not call the project a failure.

One of the major differences between the original planning and the final application is the use of Tkinter for popup windows for the user. When I began this project, I had made the firm decision to use pygame for the GUI, essentially programming all elements of the game from scratch. I had felt that using external widgets programmed by someone else would restrict my ability to develop freely, as I would be forced to keep within the constrictions of those widgets. As well as this, I was motivated to prove that I had the ability to create a game purely from nothing but my own knowledge and skills, rather than using the work of another as a crutch.

The issue with this approach, as I soon discovered, was that GUI programming proved to be far more time consuming than I had ever considered, and I found myself spending a great deal of time merely planning how to put together menu systems that the user can interact with, let alone actually implementing these systems. I made the decision that it would be far more beneficial to spend my efforts on the actual gameplay sections of the program and to just go with the easy solution of using popup windows instead. In hindsight, I believe that this was the correct choice, as the Tkinter windows are more than functional enough for the scope of this project and I wouldn't have been able to finish on time if I had not done so.

In fact, many of the difficulties I had with and changes I made to my original plans for this project can be attributed to simply not having enough time to finish and polish all the elements of the game to my liking. For example, one quite noticeable difference between my plans and reality was in the way that the saving system worked. I had originally decided to let the user save games to specific 'save slots', which would be much more dynamic and functional compared to the single slot system I ended up with. Because of the time constraint, however, this feature had to be scaled back to a (in my opinion) rather quick-and-dirty solution, which I am not particularly happy about.

This is also the reason why the high-scores portion of the game had to be scrapped. As much as I would have loved to try implementing auto-generated graphs and massive lists of player statistics, it simply was not feasible to complete these modules of the program in time, and attempting to do so would merely prove to hurt the quality of the rest of the project.

The factor that really compounded this time constraint for me was that I never realised just how time consuming bug-fixing and error testing would prove to be, especially for such a complex project as this. Carelessly attempting to add one feature at one point of the program could cause the entire set of code to come crashing down a series of bugs that could take hours to fix, which was both counter-productive and quite damaging to my personal work ethic. It did not help that the randomised nature of the rules of mahjong meant that many of the bugs would be quite obscure and may not even show up until hours after having been introduced.

In the end, however, I was still able to accomplish all of my core goals before the deadline for the project. The final version of the game is able to provide a very accurate and engaging game of Riichi Mahjong against the computer. The final UI turned out even better than I had hoped, and I feel that the Tkinter windows actually add to the experience, which is a success. There are no real bugs in the program that I'm aware of and, while not a huge concern for a program as graphically simple as this, I believe that the game process is quite efficient, and there doesn't seem to be any issues with slowdowns on older computers.

The last goal I had when coming into this project was to make it modular. I wanted to let other people use elements of my code as they wished in their own programs and, personally, I feel that the program has accomplished this to some extent. The mahjong_rulebase package that I developed allows anybody to use my definitions of Tiles, TileCollections and Players as they wish easily in their own programs, and should prove to be quite versatile in making other, similar games.

Overall, while there were quite a few difficulties and changes to the program made due to issues with time management, I feel that this project was quite successful and very rewarding. While, ideally, there are some elements that I would have liked to change given the time, I am quite satisfied with what I managed to achieve in the time-frame for this assignment.