**Revised Software Proposal**

In my original software proposal, I included the idea of making an adventure game including Mario or Indiana Jones, and I decided to use the theme of Mario with some other popular themes from social media websites like twitch.tv and reddit.com to increase the marketability of the game and increase attraction with older age groups that might find the mechanics a little too simple. I changed some things along the way from what I originally planned; I changed it to a 2D room style game where you go to the other side of the screen instead of moving across rooms in a tile-based game that I originally planned because I felt that this had a better display of programming knowledge and made the game actually interesting to play rather than just presenting a bunch of questions for users to answer every once in a while. I believe that I was quite successful from what I had planned at the beginning as a whole, even though I had to make some changes along the way.

The programming structures used in this game were mostly variations of graphics, arrays, basic for/if statements, a lot of various Boolean logic, some ‘Random’ methods, and movement by interpreting keyCodes from the int.getKeyCode and KeyListener class, which was implemented into the Game class. In addition, I used arrays to ensure the coordinates of some of the obstacles since there were issues with them when the character interacted with them. What I did to start off my ISU was research a lot about graphics; I looked up tutorials by other people, googled a lot of questions and found some useful information pertaining to graphics on stackoverflow answers where other people had already struggled with some of the things I was planning on doing, and so I made a better plan early on rather than doing everything the last day. Most of my code was related to Boolean logic and automated rather than hard-coded, except for the collision detection and movement, and this allows the user to play the game multiple times through. I tried to be efficient in my if statements with Boolean logic and I also used the switch case when necessary; although there was a lot of selection involved, the difficulty was actually plotting the coordinates on the JPanel grid because it’s not like drawing on paper, and it’s moderately difficult to visualize where objects would go on the first try. I learned the use of timers in different classes and I also introduced my own jumping physics, which aren’t perfected yet since they require some more in-depth knowledge of character movement, including the consideration of things like air drag and time spent in the air. I was thinking about using hashtables instead of arrays to show variety, although there were no values that needed to be constantly added to (hastables are mutable), although I believe one improvement to programming structures would have been a score system of some sort.

In conclusion, I’ve included most of what I was planning on doing originally and I believe that I’ve covered a lot of programming structures in this game and I’ve made it visually/graphically pleasing, which add on to its desirability towards the mentioned age group. I used arrays, OOP, graphics, Boolean logic, keyboard input, JPanels/JFrames, and I believe that while I spent a lot of time on my game, there are still some bugs with game physics (i.e. jumping) as my game engine is quite basic in comparison to bigger games that are on the market currently.