

Daniel Squicciarino  
Software Development  
Professor Arias  
13 October 2017

## Final Project Milestone

### Abstract:

I am working on creating a simple game in Java for my final project. Although it is far from complete at the moment, the plan is to create a game that depicts the way it feels for college students with serious anxiety to force themselves to go to class. The game, simply called “Get To Class”, is meant to illustrate how even simple tasks can be incredibly difficult for people who suffer from anxiety on a day to day basis. The game will incorporate a title screen / main menu. The gameplay will consist of the player pressing a single key repeatedly to move their character forward. As the player progresses, they begin to face increasing pushback, and must press the button faster to continue moving forward. The game ends when the player gets to class or gives up, at which point the game will display a screen giving the player the option to either play again or quit. The game will incorporate a simple pause function that freezes the game and displays a button to give up.

### Introduction:

Although I feel my skills with Java and my overall knowledge of how to code leave something to be desired, I wanted to try to take on a project that was somewhat manageable, something I was interested in making, and something that would allow me to flex my creative muscles a bit as well. In the end, I settled on a game, not only because I have a general knowledge of how games work and how to design games, but also because I thought it would be fun to attempt to make one on my own. When I was brainstorming ideas for what kind of game to create, I wanted to try to make something that was unique rather than just making a generic pong game, asteroid shooter, or platformer. I also wanted the game to have some sort of overall takeaway that would be thought provoking and have an emotional impact on players. These were my goals, along with conceptualizing something that I believed to be possible within the time constraints and my overall coding abilities. Eventually I settled on “Get To Class” as a way to make a game that paints a portrait of what simple tasks feel like for people who suffer from anxiety. This is meant as a way to bring awareness to what anxiety feels like, as it is often difficult for people without anxiety to imagine what their anxiety-ridden counterparts are going through. It is meant to help people understand just how hard anxiety can be, and leave players feeling more empathetic towards those who struggle with anxiety.

### Detailed System Description:

So far I have created a functional main menu for my game using a JFrame and JPanels. There are two separate classes so far. The main menu class houses the JFrame as well as the main menu JPanel, which houses two JLabels and two JButtons. The two JLabels are the name of the game and the name of the author (me) The two buttons are a play button and an exit button. The buttons use ActionListener and Action Event in order to perform their functions. The exit button leads to a System.exit(0) that closes the application. The play button removes everything that is currently in the frame and replaces it with a different JPanel in a separate class, which is where I plan to make the real meat of the game. The menu is organized with a grid layout.

UML's:

Main Menu
<ul style="list-style-type: none"> <li>- window:JFrame</li> <li>- menu:JPanel</li> <li>- title:JLabel</li> <li>- author:JLabel</li> </ul>
<ul style="list-style-type: none"> <li>+ MainMenu()</li> <li>- <u>menuLayout():void</u></li> <li>- menuLayout():void</li> <li>- displayButtons():void</li> </ul>

Gameplay
<ul style="list-style-type: none"> <li>- <u>serialVersionUID:long</u></li> <li>+ <u>WIDTH:int</u></li> <li>+ HEIGHT:int</li> </ul>
<ul style="list-style-type: none"> <li>+ Gameplay()</li> <li>+ paintComponent(Graphics):void</li> </ul>

Requirements:

This system is meant to address the problem of how the player is going to navigate the application. Instead of just having the game begin, I wanted to add a title screen as a starting place, showing the player the title of the game and giving them the option to play or close the application. This gives people a way to close the game from within instead of having to exit the

window, although I also included a line of code that allows for the application to be closed this way as well. The menu is meant to be a logical stepping stone to the actual gameplay.

#### Literature Survey:

Almost every game ever made, except perhaps some of the very earliest, has incorporated some sort of main menu or title screen before the start of their game. Having your player jump right into the game is jarring. The title screen / main menu allows players to see some basic information about the game first, like the title and the creator, to provide a bit of context for what it is they are about to play. It is fairly common practice in the games industry, and I knew that I wanted to figure out how to program a similar sort of interface for my game.

#### User Manual:

Ideally, when the game is done, it will be exported as a runnable application, so when that application is opened it will open the main menu. From there, players can click either the play button to play the game, or the exit button to close the game. When the game is finished being developed, players will have to repeatedly press a key (probably the spacebar) to move their character forward. Once the game is complete, another menu will appear and show them their results, and will again offer them the choice to play or exit. Pausing the game will provide an option to give up, which will lead to the same results screen.

#### Conclusion:

I have ambitious goals for how I want the game to look and operate in the end, but what I have completed so far addresses only the user interface portion of the project. It is the first step that allows the user to interact with the application and choose what they want to use next. I will use this framework to create the rest of the menus that I want to incorporate in the game, namely the results screens when completing the game or giving up.

#### References:

“Build a 2D Platformer Using Java.” *Build a 2D Platformer - JAVA*, 22 Mar. 2014, pp. 1–170., [www.brainycode.com/downloads/Building2DPlatformerV10.pdf](http://www.brainycode.com/downloads/Building2DPlatformerV10.pdf).  
Point, Tutorials. “SWING JButton Class.” *Www.tutorialspoint.com*, Tutorials Point, 15 Aug. 2017, [www.tutorialspoint.com/swing/swing\\_jbutton.htm](http://www.tutorialspoint.com/swing/swing_jbutton.htm).