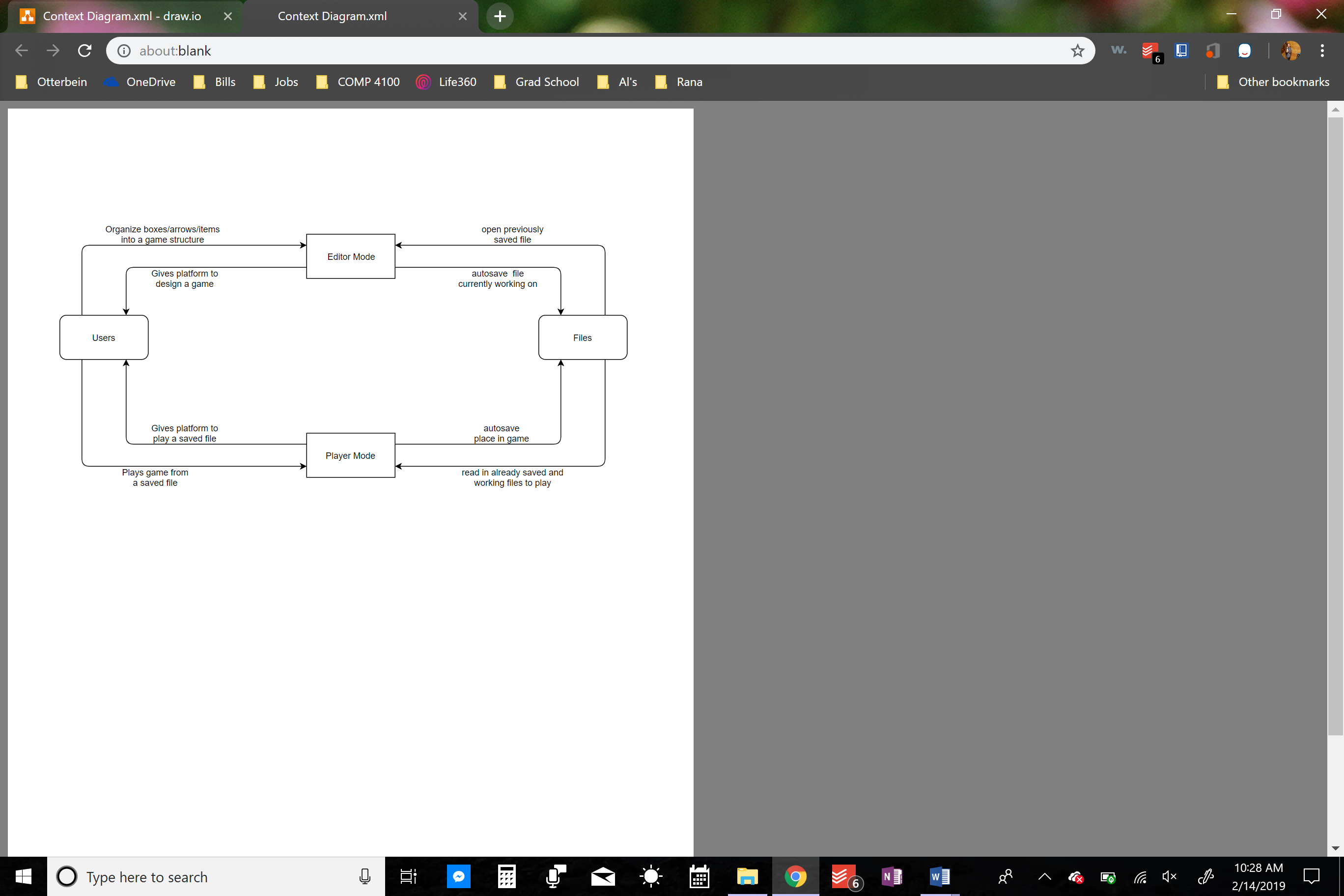
|  |
| --- |
| Pick-a-Path |
| Design Document |
| Olivia Langley, Lucia Ristea, James Erpenbeck, Logan Murphy |

|  |
| --- |
|  |

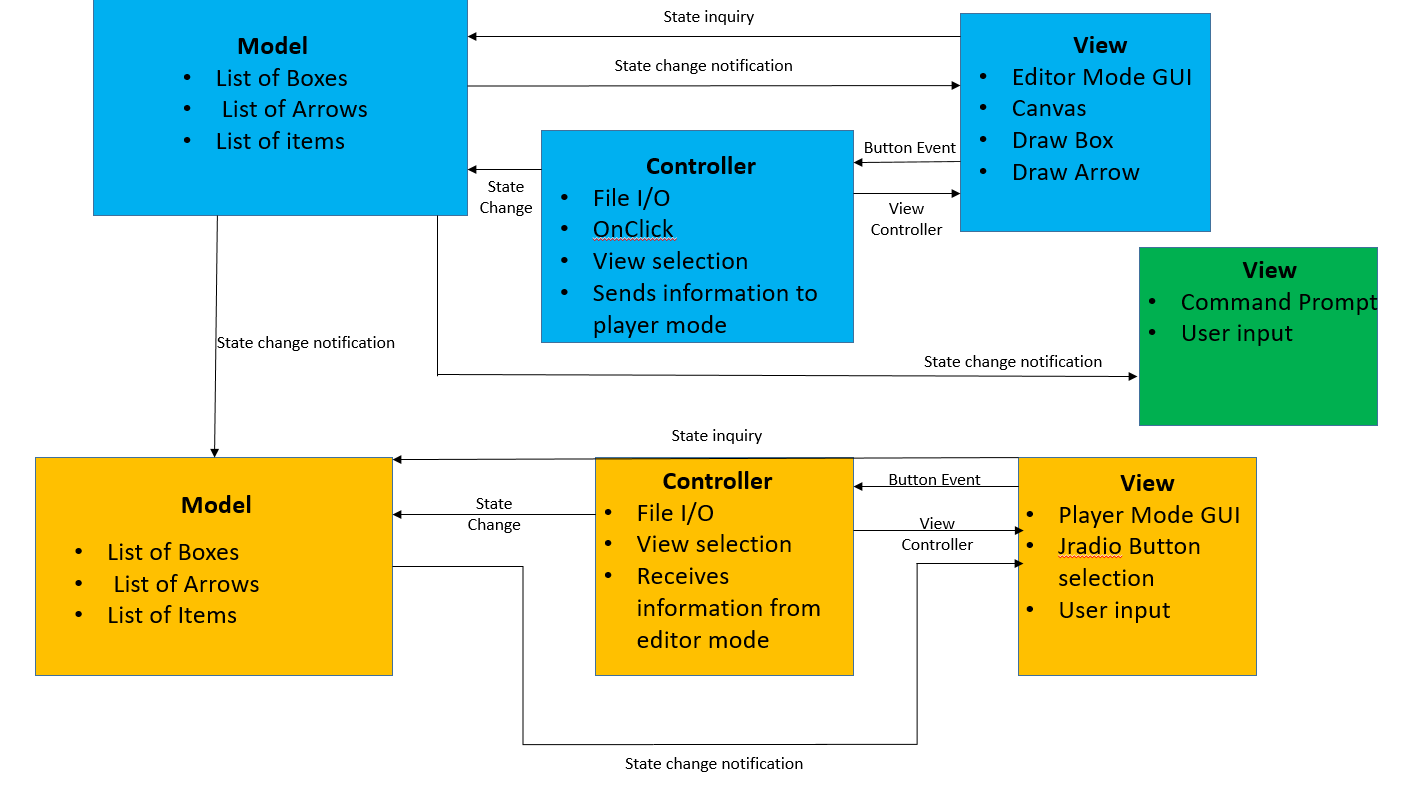
1. Context

Pick- a-Path will interact with users and files through its editor mode and player mode. In the editor mode, a user will be able to create their game by rearranging boxes as situations/scenarios, connecting those boxes by arrows as choices, and Items attached to arrows. The user’s progress will continuously save. After a scenario is made, the user can open it in player mode directly from editor mode or open the player mode separately to play it. Player mode will also automatically save your progress, so you can stop playing and come back to that point. This is also shown in the diagram below.

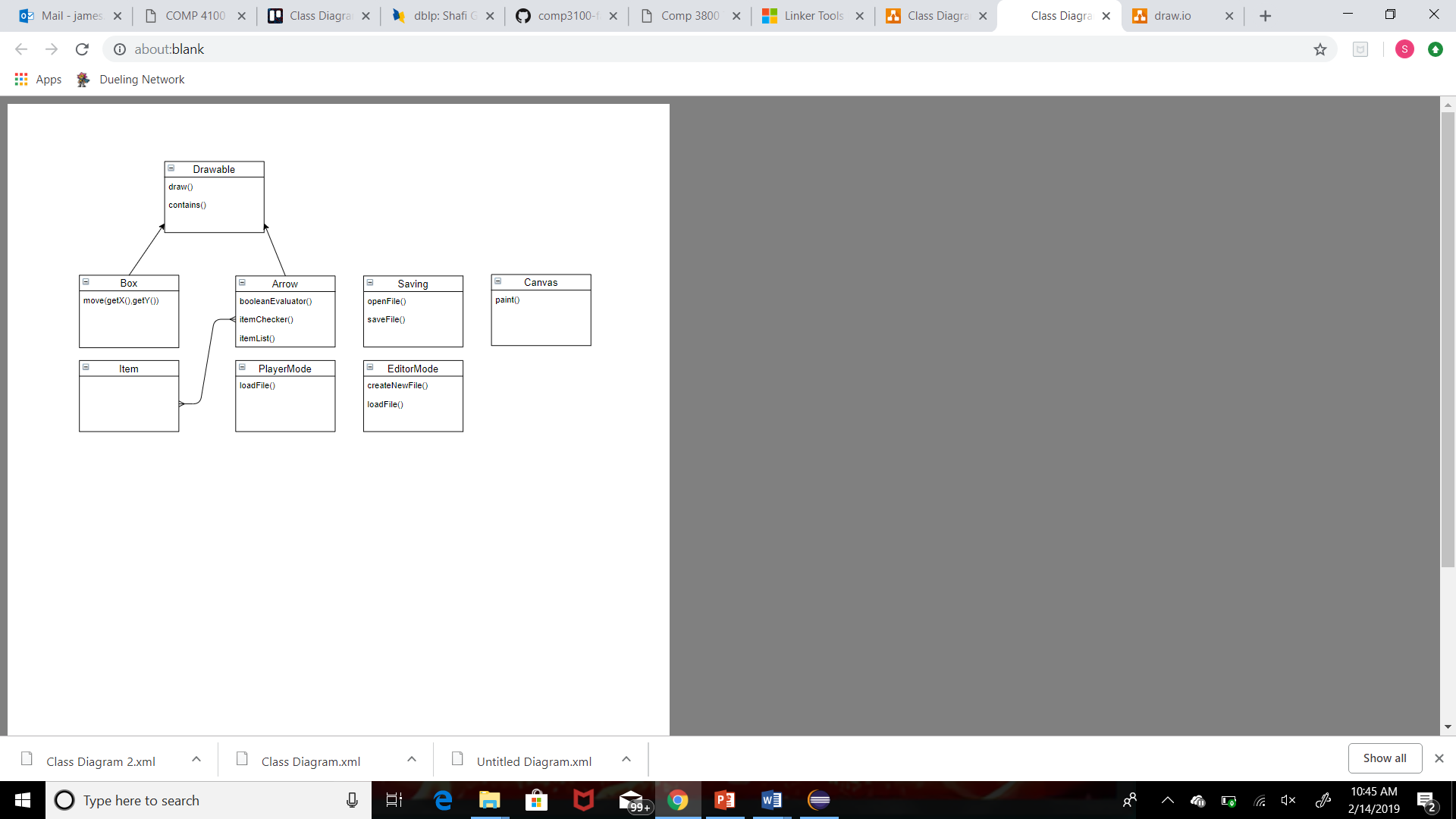


1. System Architecture

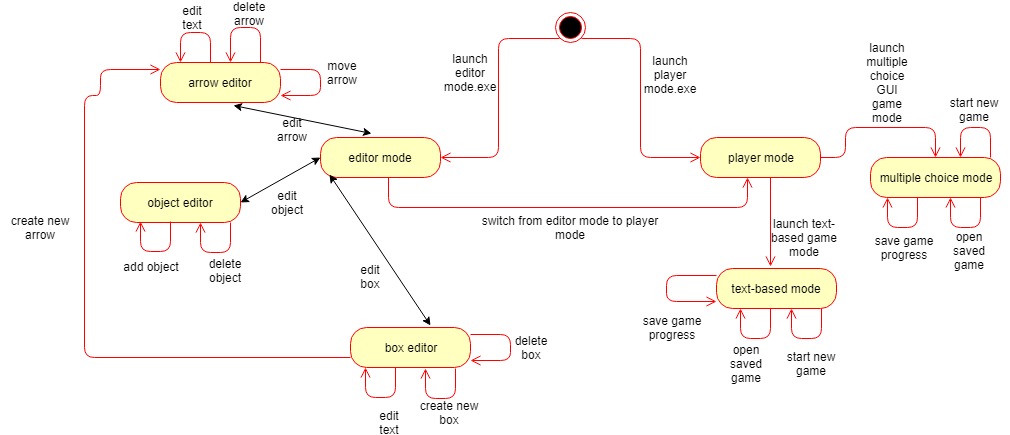
Our model-view-controller architecture outlines the components of our three systems, the editor mode, the player mode, and the command prompt.  The editor mode is colored blue, the player mode is colored orange, and the command prompt view is green.  The models manage the data of the application and their behaviors.  The view outlines the user interface and what user interface is being viewed in each mode.  The controller manages the flow of information and receives user input and output to perform the correct actions.  This MVC outlines the flow of information and how the components interact within Pick-a-Path.



1. Class Diagram



1. Design Models
   1. State Diagram



* 1. Use-case diagram



Context: 4/5

Your context is detailed, but you didn’t state the most important things: It’s a Java application running on a desktop. Thus, users will interact with your system by running a Java program on a local machine.

System Architecture: 12/15

Your system architecture is fine, but it’s confusing, and at least one of the arrows is wrong. Shouldn’t the Command line View be updated by the yellow Model representing the playable game state? Why did you paste a picture into the Word document? Pasting in a diagram directly from PowerPoint would have maintained visual quality even when zoomed in.

Class Diagram: 19/20

Your class diagram is good. I think it’s important to mark the two methods in the Saving class as static, usually done with italics.

Design Models: 12/15

In your state diagram, I think the command-line version of the game should be launched with its own, separate .exe file. It’s not like you’ll choose a command-line version from a GUI. Your use-case diagram is a little odd as well. It seems to show the player mode program and the editor mode program making decisions, as if it was itself an actor. The use-case diagram doesn’t really add anything beyond the state diagram.

Prototype: 20/30

If I haven’t made it clear before, your group is in a different situation than the other groups. You already had a semester to work on this project, so the value of your submissions must be in the additional work that you do this semester. Your prototype had prettier lines and arrows but little else beyond the old code. A command line version would have been a good choice for a prototype.

Trello Timeline: 4/5

Your Trello timeline is fine, but it could use more detail. Where is the command-line tool? What about the ability to double-click on an icon and run the game?

Spelling, Grammar, and Style: 8/10

More text in your design document would have helped make it clearer. I’d still like more comments in your Java code.

Total: 79