**Castelazo - Application Controller Pattern**

The application controller is used to centralize the logic at one point to permit an easy maintenance and a central logic point. The application controller pattern is the pattern that permits the centralization of all view logic and promotes a unique process to define the flow. You use this so that you can prevent duplicating the same code in different parts of the program. Instead, the needed code is written once, and then it can be used by different views. This is useful because if you need to make changes to a piece of logic, you only need to do it in one spot, and that change is implemented to all parts of the program that need it.

This is done by placing all the flow logic in an Application Controller. Input controllers then ask the Application Controller for the appropriate commands for execution against a model and the correct view to use depending on the application context.

As you can see in the code example, in the file called GabeControl.java, I created the ***saveGame()*** code which is used to save the game. Since this can be implemented throughout the game from different views, it is best to use it in the application controller so that it does not need to be copied in multiple places of the program. The code for ***loadGame()*** is used in the beginning, but again, ***printReport()*** can be used in various places.

Below is the code that was called from our game controller. I am responsible for writing lines 236-269 of this section.

public static void ***saveGame(Game game, String fileName)*** throws GameControlException {

try (FileOutputStream fops = new FileOutputStream(fileName)) {

ObjectOutputStream output = new ObjectOutputStream(fops);

output.writeObject(game);

} catch (Exception e) {

throw new GameControlException(e.getMessage());

}

}

It was called from our SaveGameView class, which in turn was called form our GameMenuView. Below is the excerpt from SaveGameView class

@Override

public boolean doAction(String inputs) {

String fileName = inputs;

try {

GameControl.saveGame(FarWestGame.getCurrentGame(), fileName);

console.println("You have Succesfully Saved Your Game to "+ fileName);

return true; //didn't throw error so it'll go back to previous menu

} catch (GameControlException ex) {

Logger.getLogger(SaveGameView.class.getName()).log(Level.SEVERE, null, ex);

}

return false;

}