

IST 261 Guide

Study Group

Session #1

Introduction to the Model-View-Controller (MVC) Software Design Pattern

Join the GSG GroupMe!



Table of Contents



1. Introduction
2. How this is going to work
3. MVC refresher
4. Group Breakout Session
5. Solution to example problem
6. Conclusion

Who am I?



- **Name:** Jason C. Nucciarone
- **Year:** Fourth-Year
- **Major:** Human-Centered Design and Development
- **Application Focus Area:** Data Science
- **Career Interests:** ML/AI, High-Performance Computing, Security and Privacy in Data
- **Fun Fact:** I have been fencing for about 12 years

How this is going to work

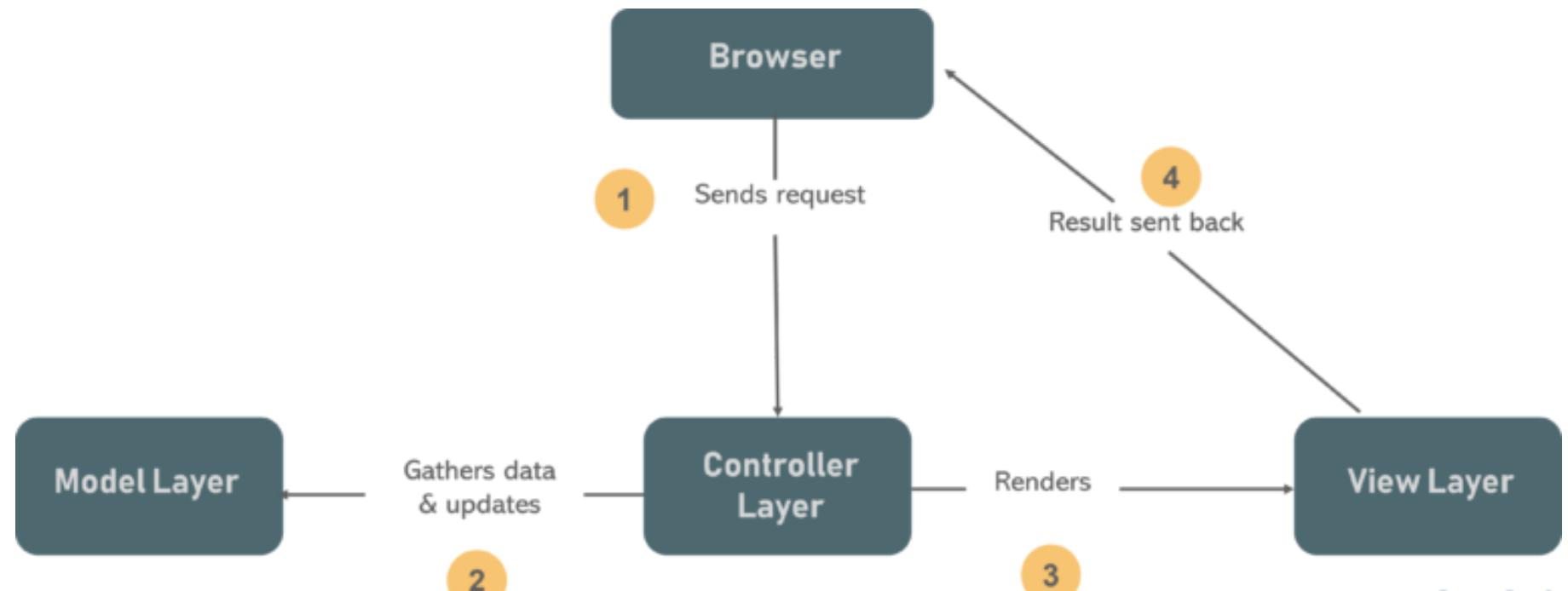
15-20 Minutes: Basic lecture going over MVC

35 Minutes: Working through example problem

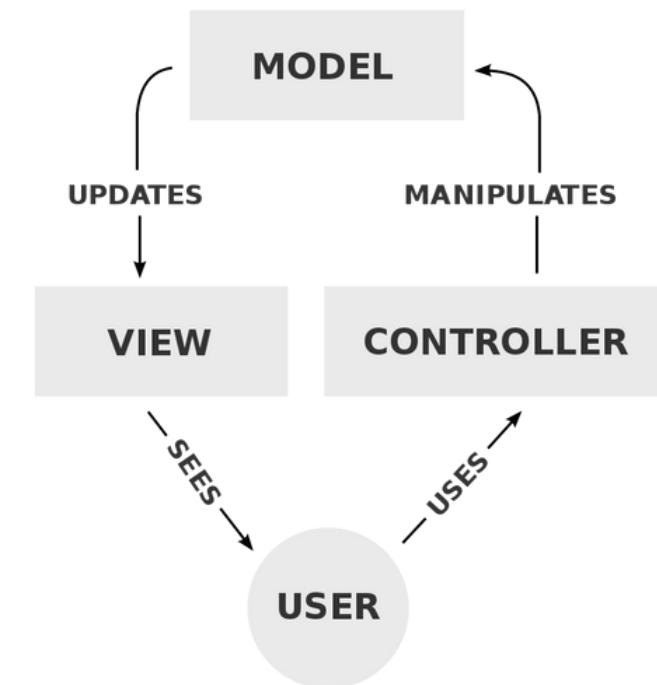
5 Minutes: Go through solution

What is the point of MVC?

- Separates the logic from the user interface in software.
 - Ex. does away with a single, giant Main.java file. Makes your application more manageable.
- Model classes can be reused in multiple Controllers and Views.
 - Ex. saves yourself the headache from redefining the same routines over and over again.
- Each component has low dependency on one another, which improves maintainability.
 - Ex. you can modify a routine in the Model without breaking everything.



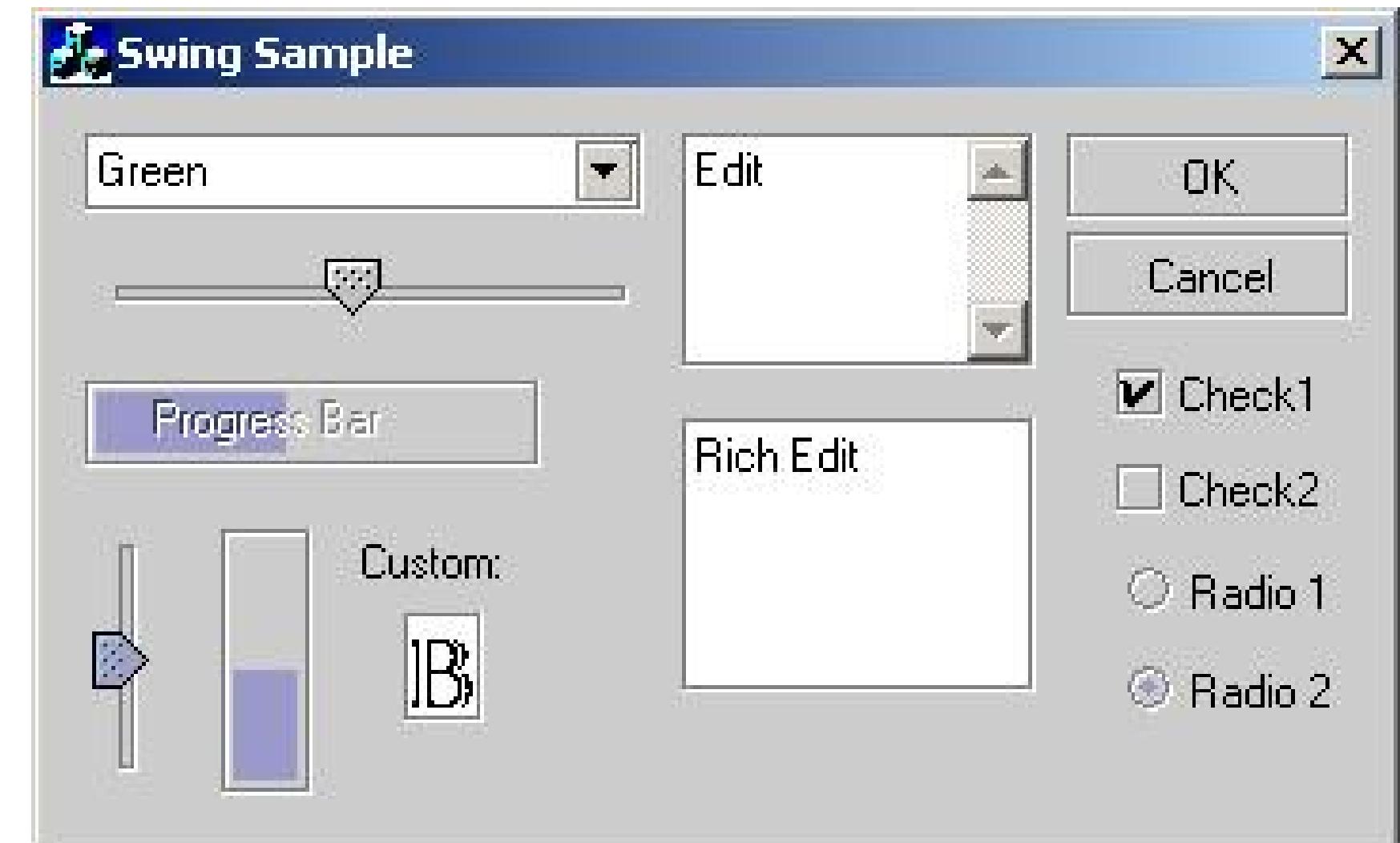
Source: <https://www.edureka.co/blog/mvc-architecture-in-java/>



Source: <https://wecancodeit.github.io/java-slides/mvc/intro-to-mvc-java.html#/7>

Responsibility of View class

- Defines what is shown to the user.
 - Think of HTML and CSS - they define what is displayed in the browser.
 - In Java's case, we have FXML templates or Swing.
- For IST 261 - View class contains all the Swing widgets and ActionEvent bindings.
- View class should not contain any business logic.
 - Ex. no math, no database calls, etc.



Source: <https://www.codeproject.com/KB/miscctrl/swing/swing.jpg>

Responsibility of Controller class

- Controls the flow of the application.
 - Think of how water flows through a landscape.
- Renders the View class.
 - Ex. update JLabels, JTextField, etc.
- Captures user input from the View.
 - Ex. grabs the String from a textbox after the user clicks the "submit" button.
- Calls Model classes to work on data and then returns the result to the View.
- Typically a Singleton.
 - Only one instance of the Controller exists at a given time.



Source: <https://external-content.duckduckgo.com/iu/?u=http%3A%2F%2Fwww.besthdwallpapergallery.com%2Fwp-content%2Fuploads%2F2020%2F06%2FFlowing-River-Photo-HD.jpg&f=1&nofb=1>

Responsibility of Model class

- "The Business Layer" of the application.
 - Ex. calculate the sum of two doubles, connect to a database, modify a string, etc.
- Called by the Controller when performing an operation on data.
 - Result is returned to the Controller which is then passed back up to the View.
- Multiple instances can exist within a single Controller.
- Can be used by multiple Controllers.
- Completely unaware of the View class.
 - Ex. Model should get its arguments from the Controller, not the View.

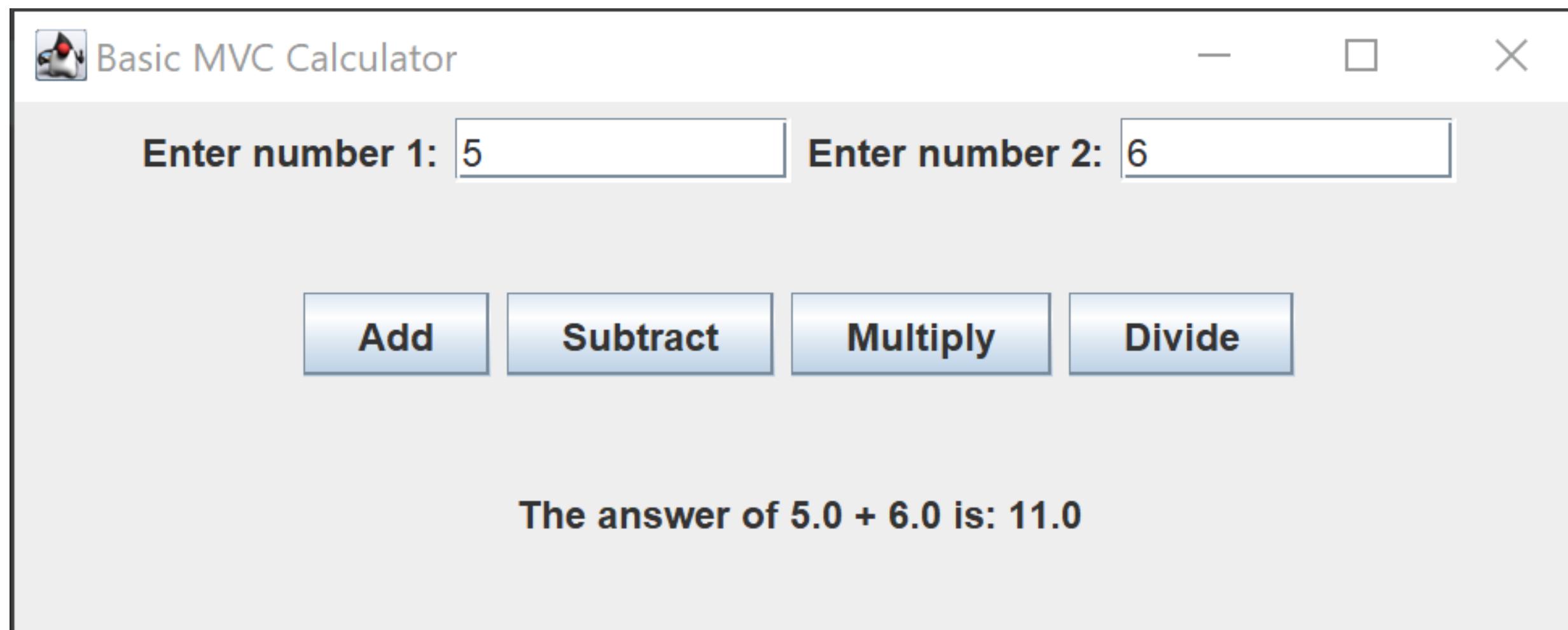


Source: <https://external-content.duckduckgo.com/iu/?u=https%3A%2F%2Ftse3.mm.bing.net%2Fth%3Fid%3DOIP.HuSg7jn4pfiGfKkaQhZ-kAHaHF%26pid%3DApi&f=1>

Example Problem Time

Please get into groups of 4

A Basic MVC Calculator



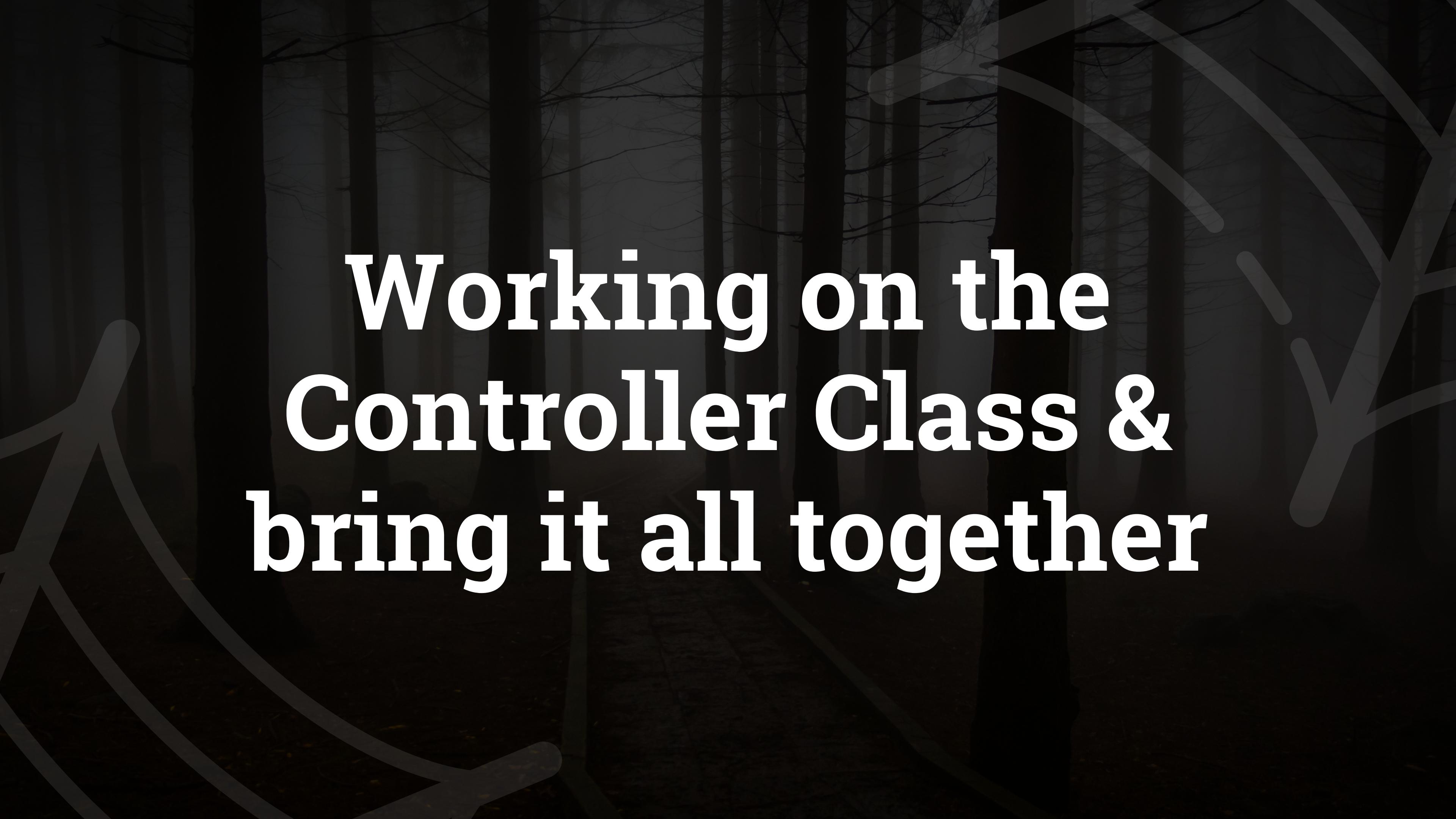
Clone the starter code here:

```
git clone https://github.com/NucciTheBoss/ist261_gsg_tutoring.git
```

After cloning the repository, open *basic_mvc_arithmetic* as a project in IntelliJ IDEA

Working on the View Class

Working on the Model Class



Working on the
Controller Class &
bring it all together



Solution Walkthrough

Join the GSG GroupMe!

