

GCIS-124-02 SoftDev II

2/12/2024

TOC

- [Housekeeping](#)
- [Notes](#)
- [Syntax](#)
- [Model View Controller](#)

Housekeeping

1. Given UML, write classes; one of the methods is spelled incorrectly
2. no classes, static stuff, interface with files and strings. Unit tests given and write code to make them pass
3. given description of a problem. Unit tests given and write code to make them pass
 - One of the methods is spelled incorrectly in some of the unit tests
 - You can look at your old code/repo during tests. No old notes. PC okay

Notes

Syntax

- **Switch statement:** switch(<value to check>); if the value to be checked fits the cases, run the inside blocks, else execute the default case.

```
switch(square.getOccupyingColor()) {  
  case BLACK:  
    squareView.setImage(ReversiGUI.BLACK_PIECE);  
    break;  
  case WHITE:  
    squareView.setImage(ReversiGUI.WHITE_PIECE);  
    break;  
  default:  
    squareView.setImage(ReversiGUI.BLANK_PIECE);  
    break;  
}
```

Model View Controller

Using Reversi is an example - The array that represents the game board is **the model** - The board that displays the black/white chips is **the view** - The buttons and their eventHandlers are the controller are **the controller**

Appearance - Each square is a button with a background image of the green square.

The making - Before hooking the GUI to display the updated moves, we first hook it to the model and display its updated string representation first (aka hook it to update changes to the model, but not displaying those changes through the GUI). - **Observer**: + When the model is updated, the observer notices that and notifies the controller, and the controller notices that and updates the view.