

**Title:** Undercooked

**Team:** Charles Zhang, Dom Lee, John Lee

**Work Done:**

John: Created Order class that includes a decorator pattern and OrderFactory class that creates an order with a random number of randomly configured ingredients. Created Store class which will serve as the Model in an MVC.

Dom: Player

Charles: Grid, Tiles, Item

**Issues Encountered:**

Removed State pattern for ingredients, since it follows a simple linear pattern.  
Simplifying behaviour of player into 2 actions (interact, place) that vary depending on the state of the program.

**Patterns:**

Decorator - Implemented within order class to have different kinds of burgers/additions to burgers customers can order

Strategy - Implemented within the action method of tiles to change item/ingredient objects

Factory - Implemented within OrderFactory class to create new order objects

MVC - not implemented

Observer - not implemented

**Plans:**

Compare function to calculate score/tips

Timer for food prep

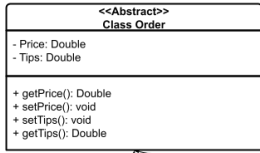
Queue for orders

GUI using JavaFX

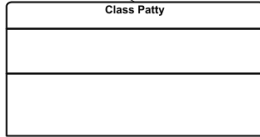
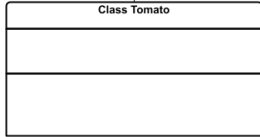
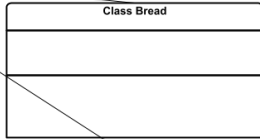
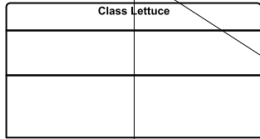
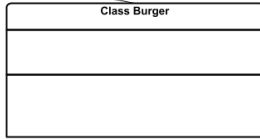
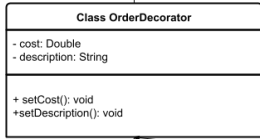
Observer (probably as manager class similar to announcer)

MVC to control player and possibly "minimize" and "maximize" screen buttons

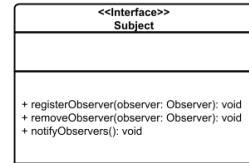
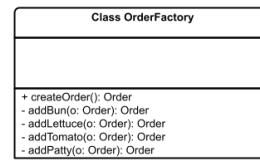
Timer to create new orders or wait until current order is complete to create new one



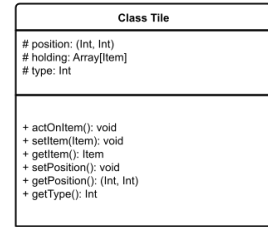
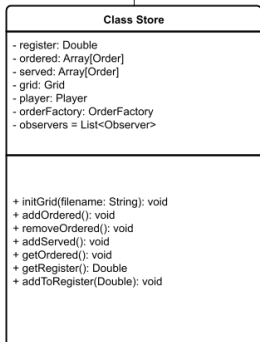
Decorator



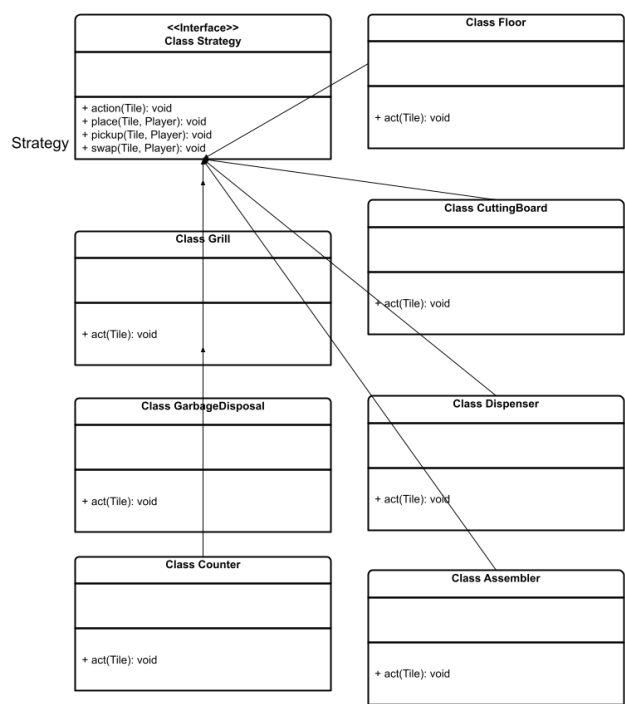
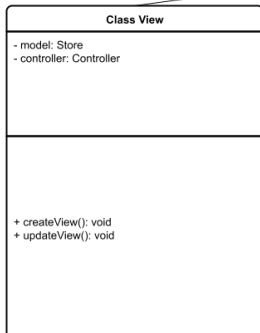
Factory



MVC



Observer



Strategy

