



Notches

The incredibly frustrating text
based game

Dylan Ahearn, Nick Smith, Leah Rogers

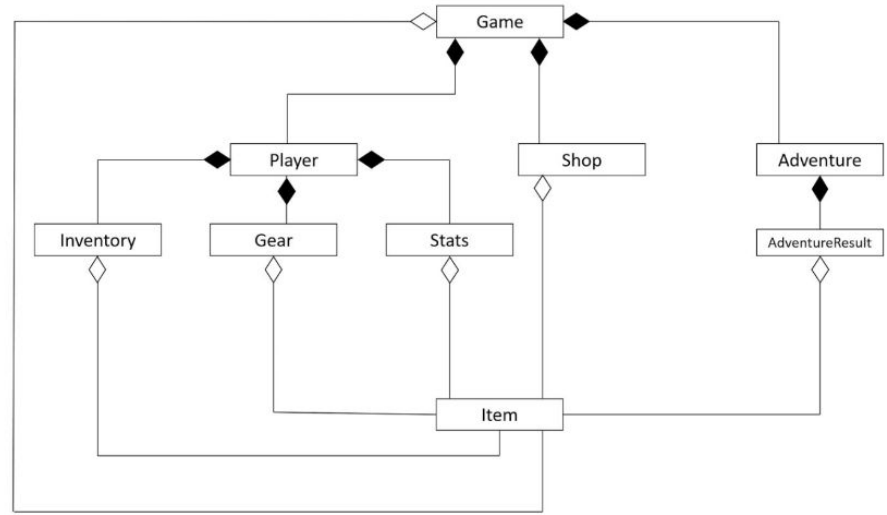


Demo (2-3 minutes)

[https://github.com/dyah7099/Notches/blob/master/12 Notches demo.mp4](https://github.com/dyah7099/Notches/blob/master/12%20Notches%20demo.mp4)

Class Diagram

- Plan increase starting time
- Easier to code
 - Filling in missing method
- Kept us Object Oriented
- Understand relationships



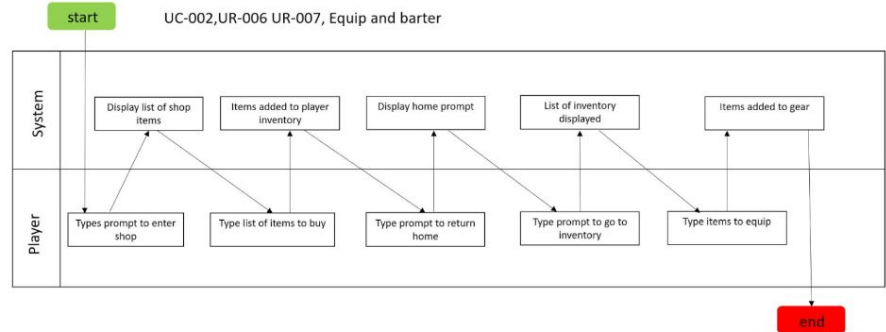
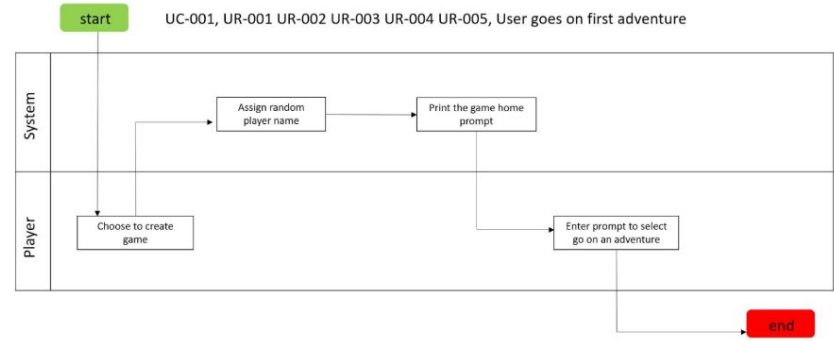
Player

- name: string
- inventory: Inventory
- gear: Gear
- stats: Stats

- + updatePlayer(items: item[], price : int): void
- + updatePlayer(coins: int, experience: int, loot: Item[]): void
- + updatePlayer(items: Item[]) : void

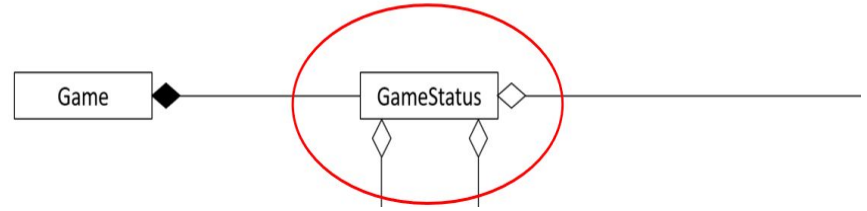
Activity diagrams

- Useful for testing
- Missing functionality
- Starting point



State Design Pattern

- New class “GameState”
- Outer/Wrapper class “Game” doesn’t directly access other game components
- Designation of responsibilities shift from component classes to “GameState” class



Object Oriented

- Implemented functionality without inheritance
- Utilized Composition and Aggregation
 - Aggregation for Player, Shop to GameStatus
 - Composition for Inventory, Gear, Stats to Player
 - Composition for Item to Shop, Inventory, Gear

```
public class Inventory {  
    private ArrayList<Item> items = new ArrayList<Item>();  
  
    //removing when adding to gear
```

```
public class Player {  
    private String name;  
    private Inventory inventory;  
    private Gear gear;  
    private Stats stats;  
    public final String[] names = {"Joffrey", "Tyrion", "Cersei", "Sansa", "Eddard"};
```

```
import java.util.ArrayList;
```

```
public class Shop {  
    private ArrayList<Item> items = new ArrayList<Item>();
```

