

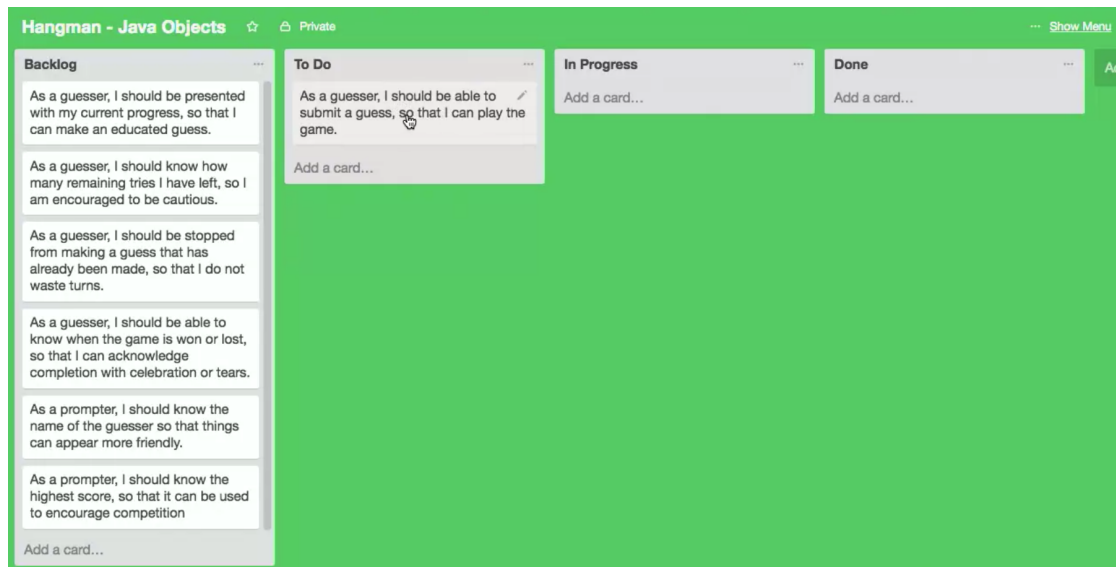
Java Objects Part 3 Notes

Team Treehouse

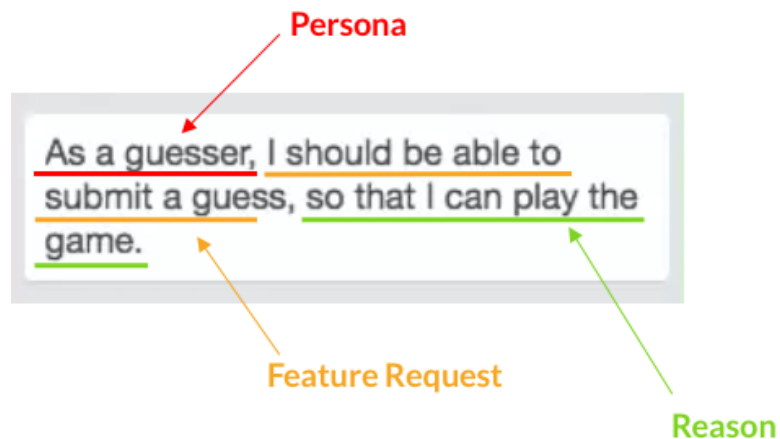
May 23, 2020

1 Planning the MVP

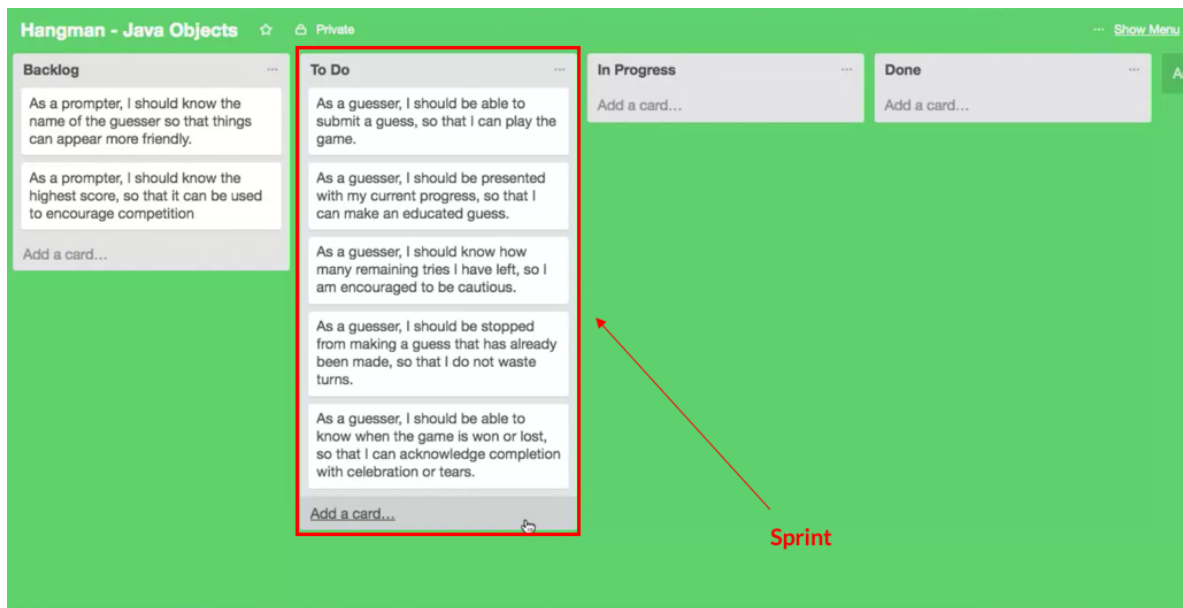
- Prioritized backlog of user stories \Rightarrow a common way of handling scope
 - Uses Kanban board



- Common format of a user story



- Sprint \Rightarrow Set period of time a list during which work will be completed and will be ready for review (i.e. By the end of the day, dun dun dun...)



2 Quiz 1

1. The set of software development practices we talked about exploring is called

- A. CS
- B. Bubble Sort
- C. Agile
- D. Brogramming

Answer: C

2. An MVP can be created by

- A. defining the minimum requirements to prove that the product is working as hypothesized.
- B. taking existing code and refactoring it so it is smaller and more compact
- C. minimizing the number of people working on the project so that the knowledge is with one person only

- D. coming up with every possible roadblock and feature that might occur and map it out on a Gantt chart

Answer: A

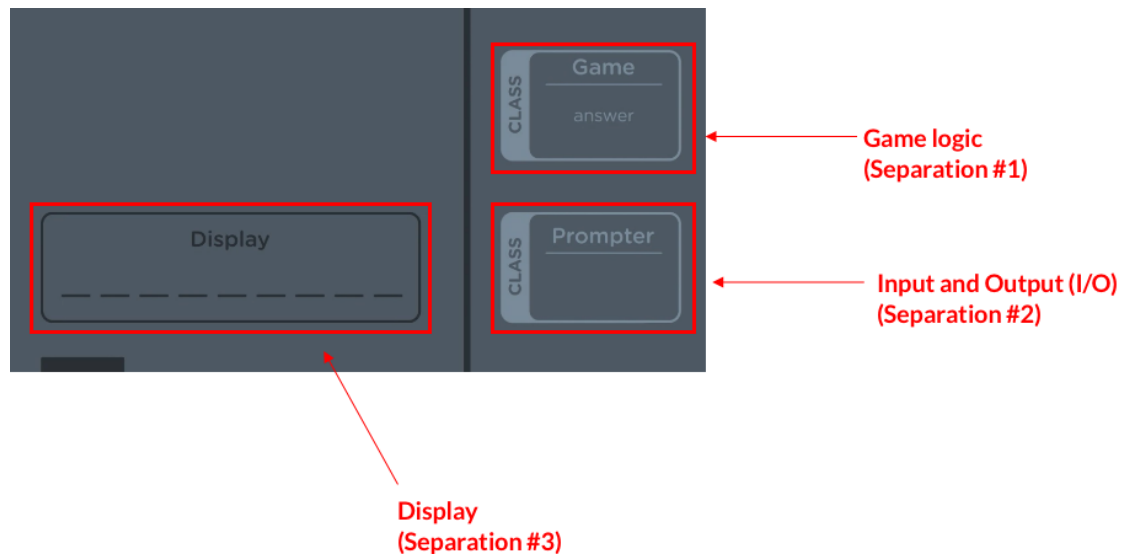
3. An MVP in our context stands for

- A. Mostly Verbatim Process
- B. Most Valuable Player
- C. Minimum Viable Product

Answer: C

3 Getting Started

- Separation of concerns heavily considered
 - The Goal is to make the code as reusable as possible, i.e. The same game logic should be applicable in desktop, website, phone in medium other than console



```
1 public class Game {  
2     private String answer;  
3  
4     public Game(String answer) {  
5         this.answer = answer;  
6     }  
7 }
```

```
6     }
7   }
8
```

Listing 1: lesson_03/Game.java

```
1   public class Hangman {
2
3       public static void main(string[] args) {
4           Game game = new Game("treehouse");
5       }
6
7   }
8
```

Listing 2: lesson_03/Hangman.java

4 Quiz 1

1. What concerns have we chosen to separate?

- A. We are separating the answer from the guesser.
- B. We have chosen to separate the display of the game from the logic, or state, of the game.
- C. We are separating letters from each other so that we can make spaces.

Answer: B

2. Due to the separation we have chosen, what outcome can we expect?

- A. We will be able to run this in other languages like JavaScript or Python.
- B. We can more easily generate code using external tools.
- C. We will be able to use the same game logic in other applications, such as console applications, web sites and apps.

Answer: C

5 Storing Guesses

- *STRING_VAR.indexOf(VAL)*: tells the index of beginning value of *VAL* in string

```
1  public class Game {
2      private String answer;
3      private String hits;
4      private String misses;
5
6      public Game(String answer) {
7          this.answer = answer;
8          this.hits = "";
9          this.misses = "";
10     }
11
12     public boolean applyGuess(char letter) {
13         boolean isHit = answer.indexOf(letter) != -1; // <- this guy
14         here :)
15         if (isHit) {
16             hits += letter;
17         } else {
18             misses += letter;
19         }
20         return isHit;
21     }
22 }
23
```

Listing 3: lesson_05/Game.java

6 Exercise 1

- Solution included in *exercise_1.java*

7 Prompting for Guesses

- *Scanner*
 - Is similar to *stdin* in *C*
 - Lives in *java.util* package

```

1  import java.util.Scanner;
2
3  public class Prompter {
4      ...
5
6      public boolean promptForGuess() {
7          Scanner scanner = new Scanner(System.in); // <- This
           little guy here :)
8          ...
9      }
10 }
11

```

Listing 4: lesson_07/Prompter.java

8 Exercise 2

- Solution included in *exercise_2.java*

9 Current Progress

The screenshot shows a Trello board titled "Hangman - Java Objects" with a green background. There are four columns: "Backlog", "To Do", "In Progress", and "Done". Each column contains task cards. The "In Progress" column is highlighted with a red border, and a red arrow points to it from the text "New Task >:!!" located below the board.

Backlog	To Do	In Progress	Done
As a prompter, I should know the name of the guesser so that things can appear more friendly.	As a guesser, I should know how many remaining tries I have left, so I am encouraged to be cautious.	As a guesser, I should be presented with my current progress, so that I can make an educated guess.	As a guesser, I should be able to submit a guess, so that I can play the game.
As a prompter, I should know the highest score, so that it can be used to encourage competition	As a guesser, I should be stopped from making a guess that has already been made, so that I do not waste turns.	Add a card...	Add a card...
Add a card...	As a guesser, I should be able to know when the game is won or lost, so that I can acknowledge completion with celebration or tears.		
	Add a card...		

- *STRING_VAR.toCharArray()*: Turns string into a char array
 - Is very similar to python's *[x for x in STRING_VAR]*

```
1      public class Game {
2          public String getCurrentProgress() {
3              String progress = "";
4
5              for (char letter : this.answer.toCharArray()) { // 1.
6                  <- This little guy here :)
7                  ...
8              }
9              return progress;
10         }
11     }
12 }
```

Listing 5: lesson_09/Game.java

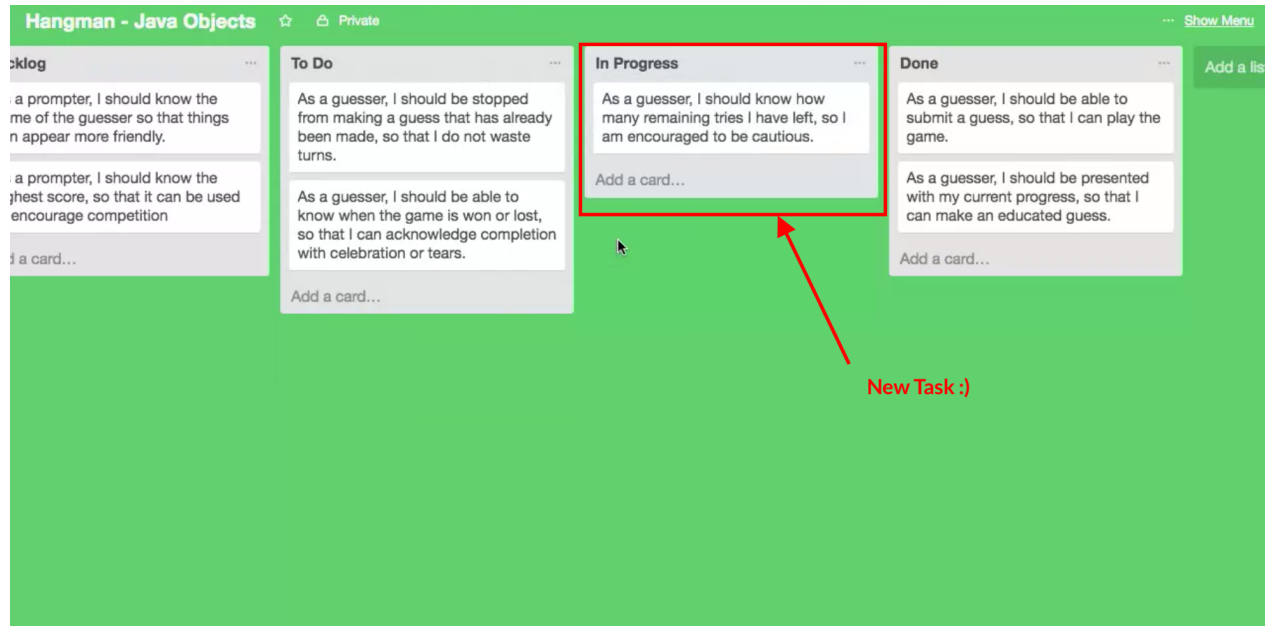
Notes:

- Files can be compiled and displayed by typing *javac Example.java* && *java Example* in terminal

10 Exercise 3

- Solution included in *exercise_3.java*

11 Remaining Tries



- `STRING_VAR.length()`: Returns the length of string
 - Is similar to `len` in Python

```

1  public class Game {
2      ...
3      public int getRemainingTries() {
4          return MAX_MISSES - this.misses.length(); // <- This little
guy here :)
5      }
6      ...
7  }
8

```

Listing 6: lesson_11/Game.java

12 Exercise 4

- Solution included in `exercise_4.java`