PURPOSE

The goal of this test project is to provide us with some understanding of how you approach problems, how you demonstrate solutions in your code.

The expectation from the solution is not to cover 100% of every edge case, but it should cover the main features in a robust way, there should not be errors during runtime, all requested features should be implemented and tested. Other than checking if the project is working or not, other main consideration points are how you approach the problem and solve it.

DEFINITION

We would like you to implement the basics of a turn-based RPG (like Idle Heroes, Disney Sorcerer's Arena)

We expect your project to generate 10 random heroes with different shapes, sizes, colors and calculate the proper stats according to these visual characteristics for each hero. After that, we would like to pick 3 of these randomly generated heroes and enter a battle with an arbitrary enemy to play a basic turn-based RPG.

FIRST PART

We would like you to create an architecture for a basic stat system.

All heroes have a name and 2 main stats:

- Name
- Health points (HP)
- Attack points (AP)

All heroes have 3 visual characteristics. These are:

- **Shape:** Circle or Square
- **Size:** Big or Small
- **Color:** Red or Green or Blue

Main stats of the heroes are defined based on their visual characteristics. A hero can only have one type of shape, size and color attribute.

Basic Hero Stats: 100 HP and 10 AP

Additional Shape Stats

1. Square: +100 HP and +20 AP
2. Circle: +200 HP and +10 AP

Additional Size Stats

1. Big: +100 HP
2. Small: -50 HP

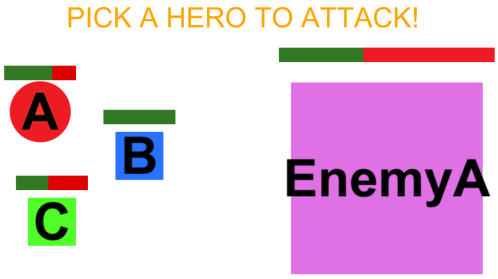
Additional Color Stats

Colors will affect main stats based on shapes.

1. Red
    a. Square: +10 AP
    b. Circle: +50 HP
2. Green
    a. Square: +100 HP
    b. Circle: +20 AP
3. Blue
    a. Square: +150 HP
    b. Circle: +30 AP

Please note that new shapes, sizes or colors could possibly be added in the future or the current details of these characteristics could possibly be updated at any time. Architecture of the codebase and the project details (Prefab types, folder hierarchy, asset naming conventions) should be as convenient as possible to support these.

SECOND PART

	
Example look of the hero selection scene	Example look of the battle scene.

- Hero selection section requirements we expect to see:

- 10 characters with random visual characteristics therefore random stats should be generated at the beginning of a session.
- Player should select exactly 3 heroes to enter a battle.
- The game should only allow the player to start the battle if 3 heroes are selected.

- Battle section requirements we expect to see:

- When the battle is initiated, the game will display the battlefield. 3 heroes on the left and 1 enemy on the right side.
- Stats of the enemy can be arbitrarily defined by you.
- When a hero is tapped by the player in the battle, that hero should attack the enemy and do damage.
- Visual representation of the attack should be implemented.
- Only 1 hero or enemy attack at a time. Game should wait for the attack action to be completed before accepting any other input from the player.
- Player can pick any hero to attack.
- Player shouldn't order an attack when the attack turn is at the enemy.
- All of the player's heroes or the boss, whichever dies first, should end the battle.
- Player should be able to return to the hero selection scene through an interface that would appear at the end of the battle.

If you think a detail is not covered, feel free to improvise about it. Also, please tell us about it.

The duration of the project is **1 week**.

Your project will be evaluated through these terms:

- Scalable, reusable, modular code and project architecture.
- Mastery of OOP, SOLID and Clean Code principles.
- Consideration of issues, synchronization and orchestration of game flow.

Additional notes:

- Preferred engine is Unity 2019.4 (LTS) and preferred language is C#.
- Do not focus on art assets, just use primitives or art from the internet. Focus on smooth gameplay, code structure and bug free code rather than visuals.
- Do not create any dependency on the OS.
- Do not create any dependency to IDEs or Unity versions if possible.
- If you create any dependency or any kind of special situation intentionally, please document this requirement properly.
- Make sure your project supports both touch and mouse events for editor and mobile devices.

Please send us the following elements:

- Unity Project (assets + project settings) completely covers listed requirements.
- A concise README file about the stuff we should be aware of, how long did it take to complete each part and anything you would like to tell us about your project.