

Procedural Generation Assignment #2: Collections and Shuffling

[Due 02/10/2020 before the start of class]

The goal of this assignment is to familiarize (or re-familiarize) you with the basics of using C# Lists to collect and randomize (or shuffle) data.

If you're new to using C# lists or need a refresher, a good set of example code is available here: <https://www.dotnetperls.com/list>

(**Note:** those examples aren't written for Unity. To reproduce them in Unity you can use "Debug.Log" in place of "Console.WriteLine" and any "Start" function in place of the static "Main" function).

To start the assignment, download the template project from this address:

<https://github.com/badtetris/ProcGen2019Assignment2>

1. Open the scene named "Task1". Modify code in Task1Generator.cs to complete these tasks:

- a. **Complete the function named "getRandomCardHand"** so it returns a list of randomly chosen card prefabs.

There shouldn't be any repeat cards in the generated hand. If you successfully complete the function, you should see the randomly chosen hand under the label "Fully Random Hand" when you run the game.

(**HINT:** there is an implementation of List and Array shuffling available in GlobalFuncs.cs).

- b. **Complete the function named "getFilteredRandomCardHand"** so it returns a list of randomly chosen card prefabs that **pass a filter** (i.e. the function "cardPassesFilter" in CardFilter.cs returns true for every card in the hand).

Note: This function shouldn't be affected by your code in **1-a** (i.e. both functions should have access to all 52 cards for their hands).

If you successfully complete the function, you should see the filtered hand under the label "Filtered Random Hand" and a description of the filter (hit R to refresh the scene a few times to make sure your code works for a bunch of different filters).

[2 points]

2. Open the scene named “Task2”. This scene combines the platformer from Assignment 1 with the cards from Task 1 of this assignment. Your task is to change the prefabs used by Task2Generator.cs so it spawns a **randomly generated platformer level** instead of a hand of cards. Feel free to do any of the following:

- Create as many new prefabs as you want.
- Modify the public variables on _Task2Generator (such as the number of cards spawned, the spacing between them, and the card prefabs used).
- Modify the function named “getRandomCardHand” in Task2Generator.cs (it currently doesn’t randomize the cards so as to avoid spoiling the answer to Task 1-a).

Avoid modifying anything else (i.e. don’t change any code in Task2Generator.cs outside of the “getRandomCardHand” function and don’t add any new objects to the scene).

[1 point]

3. Open the scene named “Task3”. For the final two points, expand or mod the existing game (from either Task1 or Task2) in some way and save your mod to this scene. How you mod it is up to you, but your goal should be to make something neat. Here are some **suggestions**:

- Create an infinite platformer by continuing to draw cards as the player moves through the level.
- Make some other game that uses a random hand of “cards” to generate a level.
- Create a two dimensional hand of cards and some method of navigating it.
- Re-create Task1 but use only ONE card prefab (instead of 52) while still spawning 52 different types of cards.

[2 points]

[Total: 5 points]

Turn in this assignment the same way you turned in Assignment 1.