Procedural Runner – Part 2 - One Randomized Screen

An endless runner with proc gen levels

# Project Administration (10 pts)

* Add annotations to this rubric: fill in the blanks, answer relevant questions, etc.
* Save the rubric in Word / .docx format.
* Put a copy of your annotated rubric into the root folder of your project (in .docx / Word format)
* When you are done, push a commit to the cloud named “Part2\_Complete”

# Base Requirements (73)

* Respect the sprite constraints:
  + You may use geometric sprites and Playground Sprites
  + You may not use any other sprites
* Respect the setup constraints:
  + You may place GameObjects in-Scene:
    - GameObjects may have only two Components: a Transform, and a Setup-type script.
    - GameObjects may have no children
  + You may use Prefabs
    - Prefabs may have only two Components: a Transform, and a Setup-type script.
    - Prefabs may have no children
  + All other setup must be done via code
* Create a ScreenManager script that does the following:
  + Uses Instantiate & Prefabs
    - Use Instantiate to create a new instance of a prefab
    - Store a reference to the newly-created object
    - Use code to rename the newly-created object
    - Change something else about the newly-created object.
    - *Explain: What prefab did you use, and what did you change?*
  + Makes a useful thing from nothing
    - Use new GameObject() to create a brand-new GameObject, and give it 3+ children
    - Write additional code to turn these GameObjects into something functional, something that plays a role in your level.
    - *Explain: What does your code create? What is the name of this thing?*
* Randomized Vectors (“build” = build with code)
  + Build something that has randomized position and randomized rotation.
  + *Explain: What is the name of this object?*
* Sometimes Obstacles (“build” = build with code)
  + Build a new deadly enemy or obstacle. Make it common, so that it shows up in 90% of screens.
  + Build a new deadly enemy or obstacle. Make it uncommon, so that it shows up in 50% of screens.
  + Build a new deadly enemy or obstacle. Make it rare, so that it shows up in just 10% of screens.
  + *Hint: The easiest way to make deadly things is to use Playground scripts. This means using code to add ConditionCollision or ConditionArea Components,then adding a Destroy action, and using code to set all the variables appropriately.*
  + *Explain: What are the names of your three enemy/obstacle things?*

# Stretch Goals:

* !! Continue to respect the sprite and setup constraints in these stretch goals.
* (+3) Prepare for, and participate in next week’s Show ‘n Tell.
  + At least one hour before the next class, send me an email with an animated gif that shows 10-60 seconds of gameplay.
  + Share 2-3 sentences about your animated gif in our next class meeting.
* (+1 to +5) Obfuscated Code Show ‘n Tell
  + Clean code is important for building good games, but many people find messy code to be a fun challenge in their off-time. The International Obfuscated C Code contest rewards people for writing the most confusing C code.
  + Create a new script called “ObfuscatedOne”
  + In the Start() function, write one line of code: Debug.Log(1);
  + Attach the script to an empty GameObject, and run your game. The script should print “1”
  + Revise your code. Change the Debug statement to read: Debug.Log(3-2);
  + Run the game. Notice that the script is more complicated, but it \*still\* prints “1” to the console.
  + Your task: Complicate the Debug.Log statement so that it uses the most complicated expression possible, but still prints “1” to the console.
  + !! Bonus points for using logical operators: >, <, >=, <=, etc.
  + !! At least one hour before class, send me an email with a screenshot of your code.
  + !! Demonstrate your code in our next class meeting.
* (+5 to +20) An elaborate enemy or obstacle
  + Create a new type of enemy or obstacle
  + Give it some sort of motion (using Playground scripts)
  + Make it elaborate and fancy. Do one or more of the following
    - Use multiple child objects with sprites, so it \*looks\* fancy.
    - Use a complex motion script, like Patrol
    - Give it multiple forms (when destroyed, it creates a new obstacle in the same place)
    - Something else. your choice.
  + *Explain: What did you make?*
  + *Explain: Why is this thing elaborate or fancy?*
* (+1 to +20) Other. Build an additional feature related to this week’s topics: using setup-type scripts to create a scene2D.
  + *Explain in detail: What is your nifty thing, and where have you used this thing in your project?*