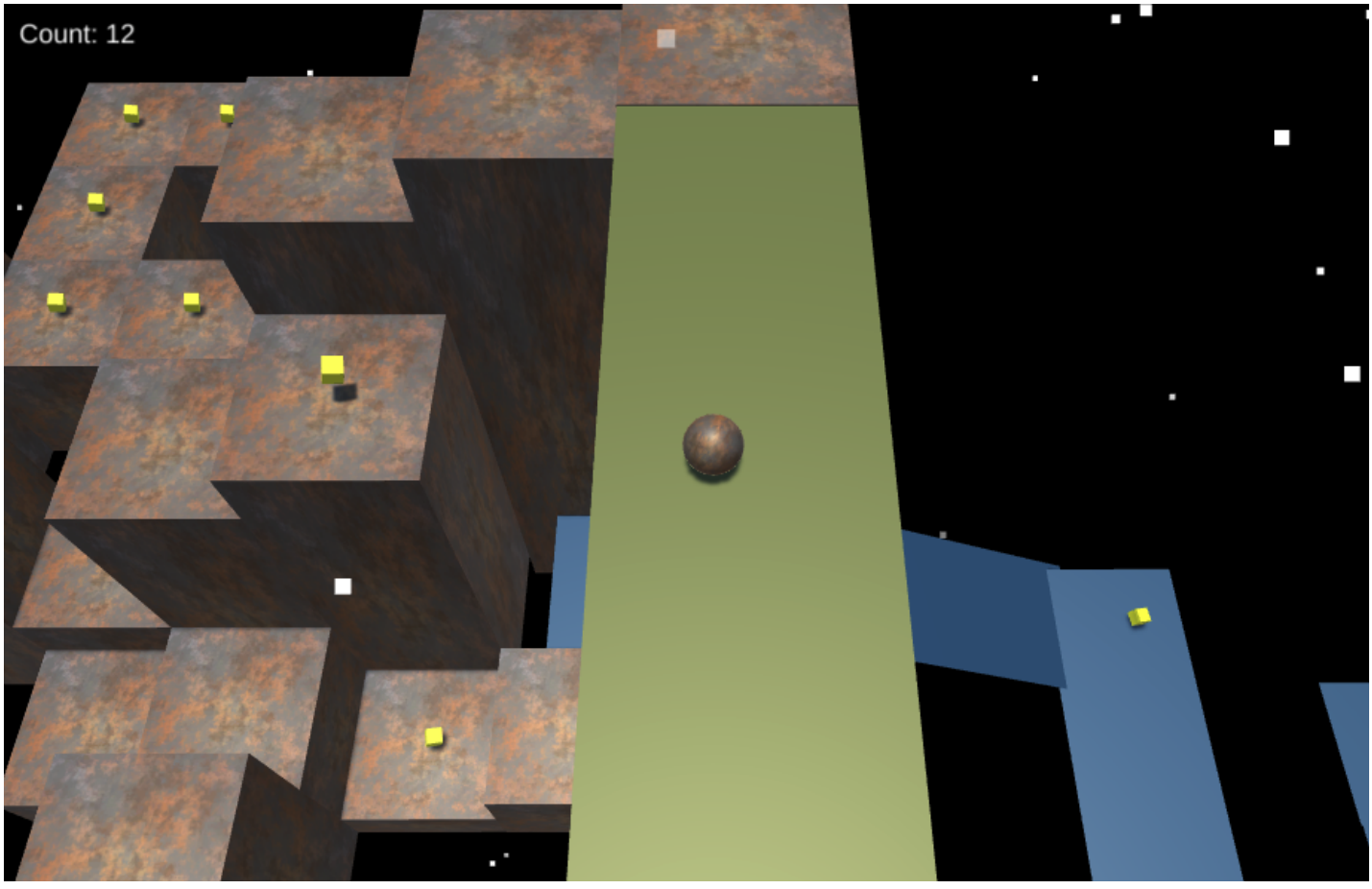
Tutorial 1

Introduction to Unity : Super RollABall

Estimated Time Investment: 2 hours

Students per team: 1

Restriction: No external assets are allowed, except for player avatar, collectible, or stage aesthetic customization. No external scripts are allowed.



Unity's Classic "Hello World" tutorial, *RollABall*. This screenshot shows a customized version, which may be experienced here. For an even more deluxe implementation of RollABall, see *AtmaSphere* by Mazen Games.

Pitch

Tutorial 1 is an assignment that tasks students with following and implementing an official, step-by-step tutorial from Unity, the creators of the Unity3D Game Engine, before customizing the game with some new features. Students will quickly come up to speed with Unity, potentially creating their very first video game in the process.

Purpose

EECS 494 is the only game development course at the University of Michigan, Ann Arbor, and thus demands a quick ramp-up of Unity skills and experience. This assignment will--

- Establish basic familiarity and experience with the Unity game engine.
- Introduce students briefly to the composition design-concept ("has-a" vs "is-a").
- Verify understanding and exercise design chops by requiring students to introduce new mechanics and level design to the tutorial game.

Tasks

Roll-a-Ball Implementation

1) To avoid crashes, install and use Unity's latest LTS (Long-term-support) version.
    a) Open Unity Hub (or install it if you haven't). The program looks like this.
    b) Check your "installs" tab and install the latest LTS version. If you're on windows, check the "mac support" box. If you're on Mac, check the "Windows support" box. You must be able to build games for both platforms.
2) Complete the "Roll-a-Ball" tutorial, implementing the game as depicted in the videos. Consider increasing playback speed on the tutorial videos (modern alternative tutorial here, but be sure to set your Active Input Handling to "Both" via Edit -> Project Settings -> Player Settings -> Other Settings, or the controls may not respond)
3) Customize "Roll-a-Ball" with the following new mechanics (Hints)
    a) Add a "jump" mechanic to the game, where the ball will hop when the player presses the space bar. The ball should only be able to jump when it is on the ground (touching a wall is also ok).
    b) Add a falling-reset mechanic, where the game restarts if the ball falls too far underneath the level (items get reset, the player gets reset-- all of it resets).
    c) Add additional floors, walls, collectables, and move everything around to create your own unique "Roll-a-Ball" level (please only submit this level). It doesn't need to be a large level, just different than the one the tutorial leaves you with. An example is shown above, but it need not be that different or that substantial.
    d) Customize the player avatar gameobject's aesthetics by
        i) Locating the Unity Asset Store.
        ii) Search for a free 3D model that entertains you (use the free assets filter).
        iii) Click "Add to My Assets" -> "Open in Unity" -> Download -> Import -> Import, and you should find new files added to your Project view.

iv) As seen in this video, locate the 3D model you downloaded, drag it into your player gameobject ("parenting" it), then disable your sphere's sprite renderer. You may need to adjust the local position of your new 3D model, which is shown in the video.

v) (Optional on this assignment in F20) Remember to add a credits.txt document to your project and list your chosen external assets within it.

vi) (Optional) If necessary, find the material that came with your 3D model and drag it onto the model using the scene view.

(1) You are welcome to make changes to the collectibles, textures, etc too (not required). Please be sure to keep the roll-a-ball gameplay and custom features intact however for grading purposes.

Game Distribution

4) Create a Web-Build by following this tutorial.
5) Once you have your web game working and hosted, submit this URL into the "project_progress" channel of this semester's EECS 494 server. Consider checking out your peers' games if you have time.
6) Submit to Canvas (follow the instructions in the *Deliverables* section below).

Deliverables

Submit to canvas one .zip file, **with your uniqname(s) in the title**, containing...

1. One Assets folder (the top-level folder titled "Assets" in your Unity project).
2. One ProjectSettings folder (the top-level folder titled "ProjectSettings" in your Unity project).

Test your WebGL build by visiting "www-personal.umich.edu/~<uniqname>/WebBuild". Re-download and test your submission to verify you submitted what you think you did.

Tips

- When installing Unity, make sure you opt-in to OSX, Windows, WebGL, Android, and iOS export functionality. You will need the ability to create these kinds of builds for other assignments in the course.
- Commit to doing 30 minutes of work the day the assignment is released. This will give you a good, early feel for how much time the assignment will require.