
Procedural Dungeons

Description

Make a procedurally generated dungeon for a 2D top-down game.



Requirements

Level Generation (70%)

- You should create a collection of scripts that can generate a functional and interesting level for the game pieces provided.
- When the scene is reloaded (upon victory or player death), a different dungeon should be generated using the same set of rules.
- Your level must be bound on all sides by walls.
- All objects that comprise the level should be spawned on Awake (to avoid conflicts with Start)
- A level should last roughly 2 minutes of normally paced gameplay.
- A Portal goal object must be reachable, denoting the end of the level.
- Your Level must include at least one enemy of each type (Enemy, Tough Enemy and Boss Enemy)
- Your level must include at least one Key-Door pair that needs to be used to reach the goal.
- Walls and Floor objects must not be smaller than a 1 by 1 and must have round integers for X and Y dimensions (i.e. 5 x 3 is acceptable while a 4.5 x 4.5 is not) and should not be rotated.

Varied Levels (20%)

Your algorithm should be able to produce rooms and levels that are noticeably different from each other in interesting ways and produced varied outcomes.

- **Room:** Size, Shape, Connections
- **Item:** Types, Placement, Number, Grouping
- **Enemy:** Types, Placement, Number, Grouping

| Grade | Description |
|----------|--|
| 0 – 5% | Little to no variety of rooms items and enemies within the level |
| 6 – 10% | Decent variety of rooms items and enemies within the level |
| 11 – 15% | Variations are fairly interesting or produce distinctly varied outcomes |
| 16 – 20% | Lots of variety with interesting features and distinctly varied outcomes |

Engaging Levels (20%)

Your algorithm should be able to produce rooms and levels that are nicely tuned and produce challenge and exploration based engagement.

- **Structure:** Hook, Development, Turn, Resolution
- **Tension:** Peaks & Valleys

| Grade | Description |
|----------|---|
| 0 – 5% | Levels are never / rarely structured in a particularly engaging way |
| 6 – 10% | Levels commonly have peaks and valleys in engagement |
| 11 – 15% | Levels commonly have distinct turn or deliberate peaks and valleys in engagement |
| 16 – 20% | Levels almost always have a distinct turn and plenty of peaks and valleys in engagement |

Submitting the Assignment

Once you've successfully ran the program and got the desired output, you're ready to submit.

1. Navigate to your project folder and verifies that it contains all relevant files & folders:
 - Assets folder (containing scenes, prefabs, scripts, sprites, etc)
 - Library folder
 - (All other folders are non-critical and should not be included)
2. Copy the folders mentioned above into a new folder called **Project**
3. Within Unity, make sure that your Build Settings are properly configured.
4. Your export folder should contain the following files:
 - Project executable (.exe)
 - Project Data folder
 - Mono folder
 - UnityPlayer.dll
 - UnityCrashHandler (optional)
5. Right click the export folder and rename it to **Game**
6. Include your **Project** and **Game** folders into a submission folder with the following convention:

CS299[Section]_[DigiPenLogin]_Project1

7. Replace [Section] and [DigiPenLogin] with your own information; yours should look like this:

CS299A_john.doe_Project1

8. Right click the submission folder and select SendTo -> Compressed (Zip) Folder
9. Upload the file to the appropriate link on the Moodle course page.
10. Once the upload is complete, download it from Moodle to verify it contains all appropriate files.

Submission Folder Structure

- CS299A_john.doe_Project1
 - Project
 - Assets
 - Library
 - Game
 - GameName.exe
 - GameName_Data
 - Mono
 - UnityPlayer.dll