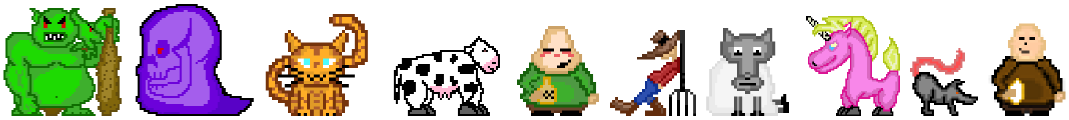
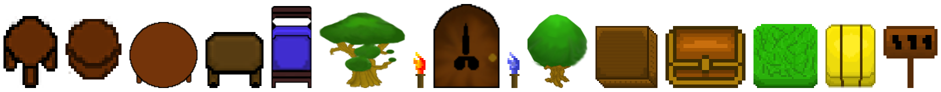


UNT4 Group Hydra Final Report

Ethan Dunham
Dylan Markovic
Katheryn McHolick

Group Hydra's goal was to create a turn-based rpg similar to an old final fantasy game. We decided to make a light hearted and satirical game. If something seems funny or weird, it was probably meant to be that way. From the talking animals, weird shaped door markings, to the obscene amounts of gold that doesn't seem to actually do anything; we tried to make a parody of the overproduced and over marketed RPG. We created a fun game that has a very high level cap. The max level is only limited by the max integer value for experience. Thus, a player could keep fighting random enemies until they hit level 46,340. We also added in a hedge maze for the player to find their way through. This maze provides a fun, and possibly frustrating, puzzle to solve and a brain teaser to remember the path back.

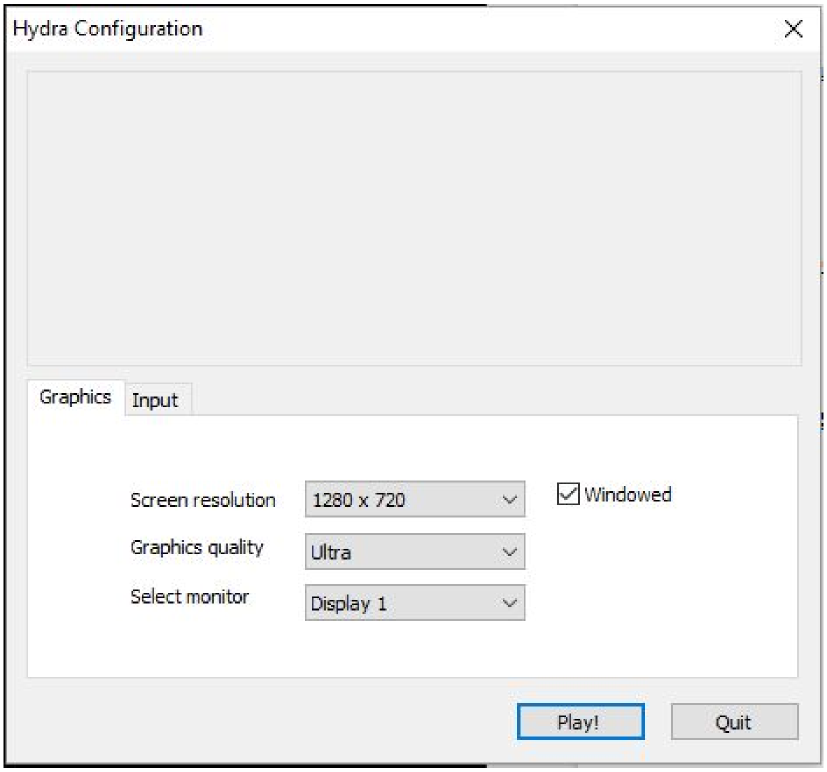
Throughout the term we discovered how to use unity, grew our knowledge in C#, and learned how to team code share using github. There were some difficulties we overcame, such as some of us learning C# and unity for the first time, and members working in the same scripts at the same time resulting in some github errors. However, we overcome these challenges to complete our turn-based rpg computer based video game.



How to have fun

Executable Game Application:

The game is pre-built into an executable calle hydrarpg.exe that will run on Windows x86 systems. When the application is launched it will bring up the ‘Hydra Configuration’ box that will allow you to select your desired resolution, quality, and monitor. Press ‘Play!’ to begin.



Screen resolution: 1280 x 720 ☑ Windowed

Graphics quality: Ultra

Select monitor: Display 1

Example Game Settings

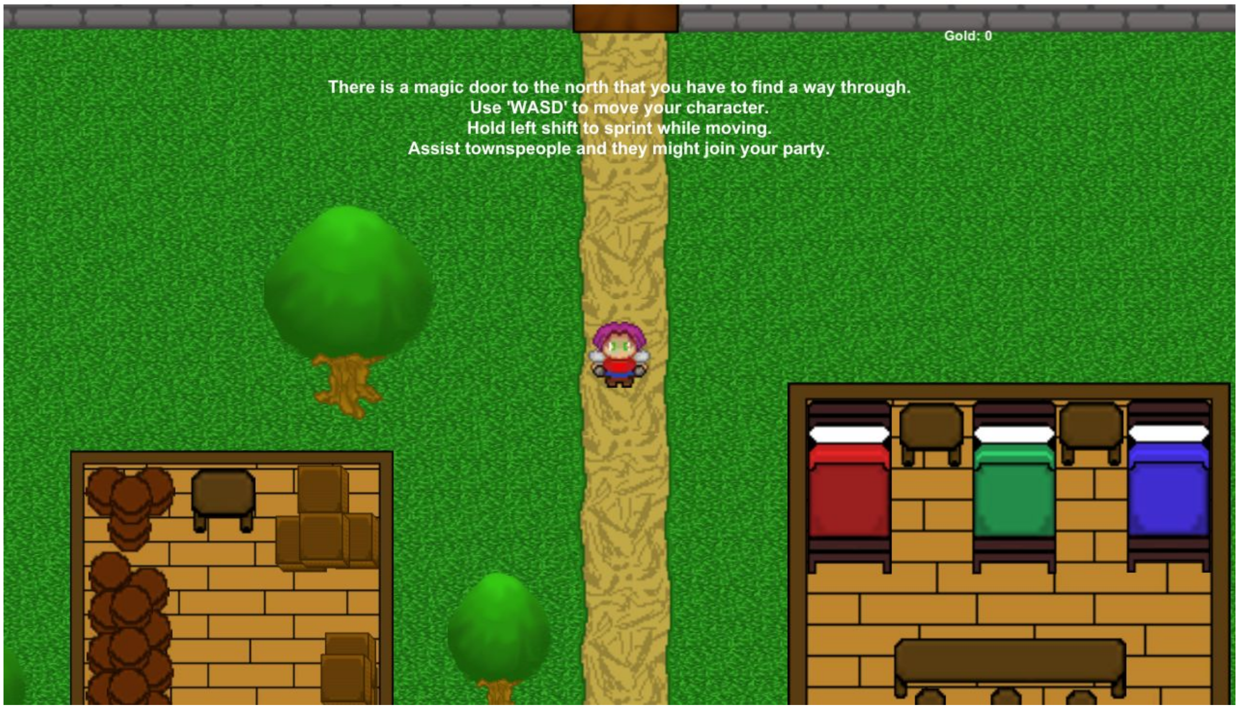
<u>Controls</u>:

- W-S-A-D or Arrow Keys for movement
- [Esc] for pause menu which included save & quit, and character menu
  - This is where you will level up your characters when you see the pop-up at the end of a battle telling you that a member of your group leveled up.
- [left shift] to run
- [space] to talk to npcs and interact with other objects.

<u>Playthrough Instructions:</u>

1. Unzip the submitted file “Hydra_OSU-master.zip” and run hydrarpg.exe
   a. If this does not work, open unity and select the Hydra folder as your project
   b. Then go to file and click build and run

2. Choose “New Game” and enter the name of your Adventurer
3. Follow the Game Objectives listed below.



There is a magic door to the north that you have to find a way through.
Use 'WASD' to move your character.
Hold left shift to sprint while moving.
Assist townspeople and they might join your party.

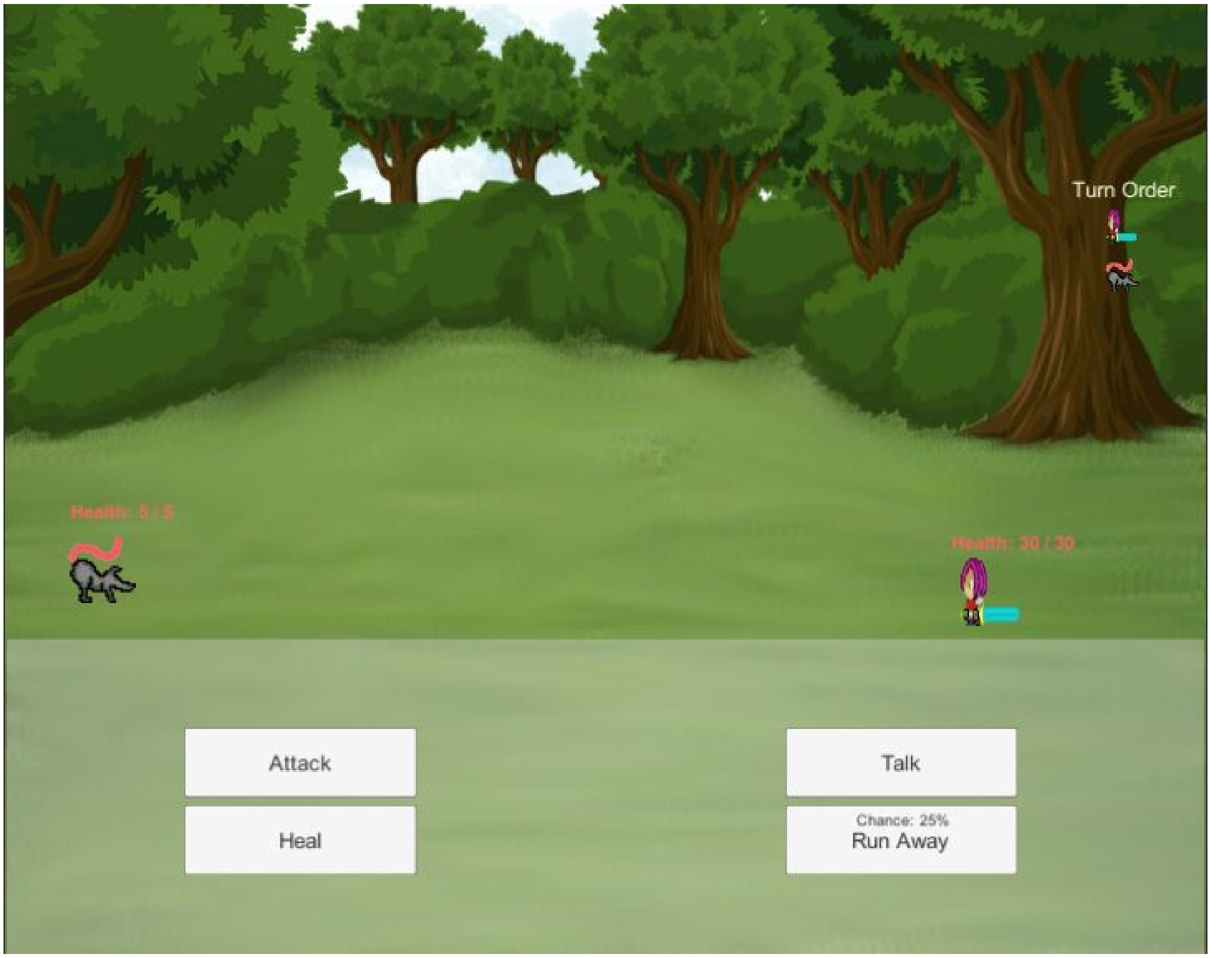
You’re ready to begin your new life as an Adventurer!

Game Objectives :

1. Help NPCs around the village
    a. This is recommended because some of them will join your party and aid you in fighting enemies
2. Interact with the Boss Door to the north
    a. It requires a unicorn horn and conjoined goblin twin heads to unlock
3. Kill the goblin twins in the cave and collect their heads
4. Kill the unicorn on the other side of the hedge maze and collect its horn
5. Unlock the Boss Door
6. Kill the final boss
7. Rejoice in victory!

Tips & Spoilers:

1. You should take some time to level up your characters before engaging the bosses. Level up by fighting random mobs or bosses

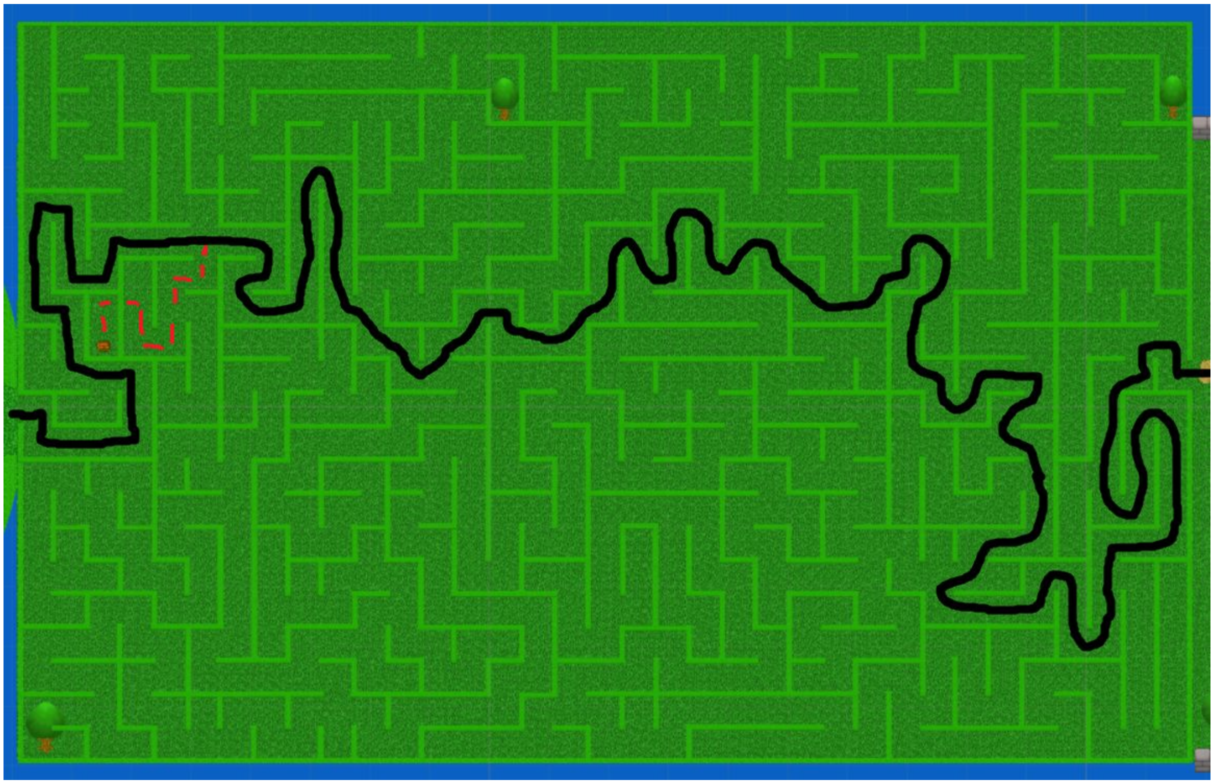


Battle...Start!
The turn order can be seen in the top right side of the screen.
Use the buttons to select actions that you want to do.

2. If you ask about the closest store enough times, you might magically find one
    a. Please, let us know if you do, it will actually be magic
3. The cat wants milk, ask the cow nicely to obtain milk
    a. killing the cow will make it impossible to recruit the cat
    b. The cow can be killed after gaining the milk without negative side effects
4. The sheep and farmer don't like each other, you will have to decide to kill one, both, or neither
    a. If you kill one, the other becomes willing to join your party
5. The drunkard wants unicorn gin, you can purchase some from the innkeeper if you have the gold
6. There is a chest in the cave that is locked
    a. Interact with the nearby torch and turn it blue to unlock the chest
7. There is a chest in the hedge maze that looks impossible to get to
    a. Walk through the east wall near the north side to get to the chest

    b. This part of the maze is an illusion

8. The cat is the best at healing
9. The sheep is best with magic attacks
10. The farmer and drunkard are good at melee attacks
11. Strength increases your melee attacks
12. Piety increases your healing
13. Wisdom increases your magic damage
14. Dexterity increases your chance to hit
15. Resistance makes you feel stronger, but has no discernible effect
16. Maze solution:



The red line is the hidden path to the impossible chest

How To View Code:

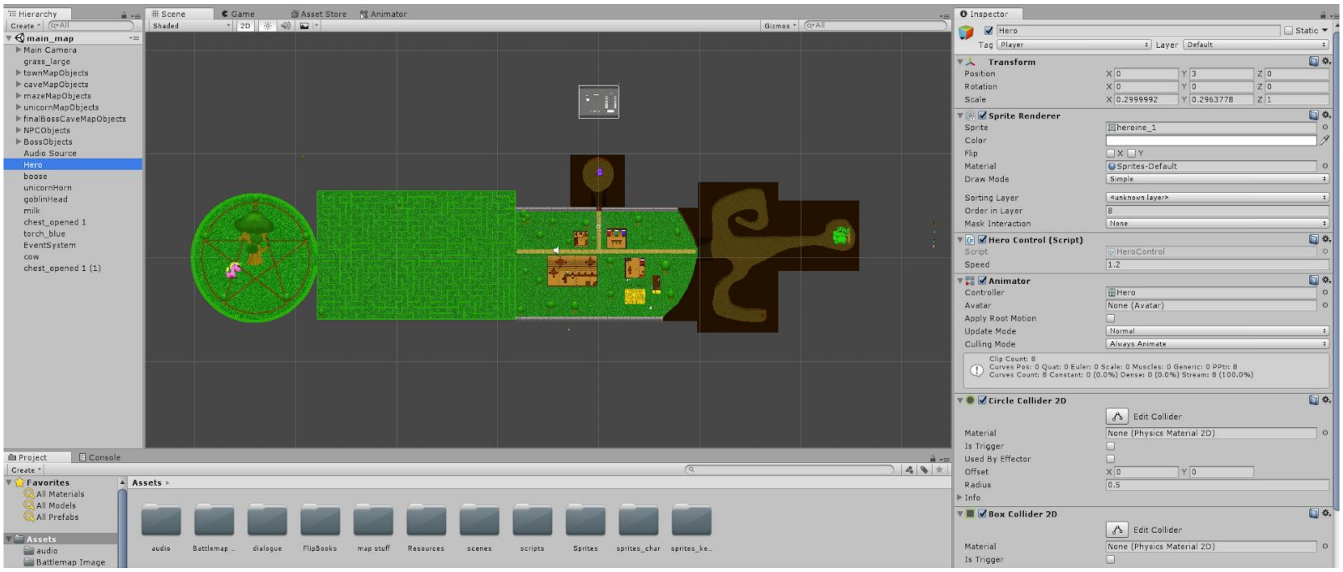
- You must have MS Visual Studio 2017 to open the code with this method:
- Using the Project tab, navigate to Assets/scripts
- Double click on any file to view it in MS Visual Studio
    - to view code in any other text editor, you can find the code here:
    - ..\Hydra_OSU-master\Hydra_OSU-master\Hydra\Assets\scripts
        - where the '..' indicates the location you unzipped the file to.

How It All Works

This project used Unity as our main core Software System. To download and install unity go to : https://unity3d.com/ .
This game functions by compiling and creating a game based on sprites, scripts, and defined behavior of objects. It creates a compiled exe to run the program with the selected operating system.

Unity Basics:

- Asset: An asset is any item that is used in a unity project. This could be the background music, a character image, or a C# script.
- Scene: A scene in Unity is where you create your game environment or menu. Game objects are placed in a scene to essentially build a game. Our game is comprised of a series of scenes. From the main menu scene, the battle scene and many more.
- Script: A script at its simplest form is C# code. They are essential to building a game in Unity. We used all sorts of scripts from simple scripts that changed the Scene, to scripts that handle the battle mechanics.
- Animation: For a two dimensional game in Unity, animation is accomplished by switching between numerous pictures in sequence. The animation takes images, and a frequency, and transitions between the images at the given frequency.
- Animator: The Animator is responsible for transitioning between differently animations for any given game object. This is done by linking all of the animations of a character and adding boolean expressions for when each animation should be taking place.
- Hierarchy: The Hierarchy window displays all the game objects in the current scene. Game objects can also be grouped and given “parent” or “child” dependencies.
- Inspector: The Inspector window displayed information about the currently selected game object. In the Inspector you can edit a variety of properties depending on the game object type. This could range from position, scripts used by the object, sprite renderer, colliders, and much more.



☑ Hero □ Static

Tag Player Layer Default

Transform

Position	X 0	Y 3	Z 0
Rotation	X 0	Y 0	Z 0
Scale	X 0.2999992	Y 0.2963778	Z 1

☑ Sprite Renderer

Sprite	heroine_1
Color	
Flip	□ X □ Y
Material	Sprites-Default
Draw Mode	Simple
Sorting Layer	<unknown layer>
Order in Layer	8
Mask Interaction	None

☑ Hero Control (Script)

Script	HeroControl
Speed	1.2

☑ Animator

Controller	Hero
Avatar	None (Avatar)
Apply Root Motion	□
Update Mode	Normal
Culling Mode	Always Animate

```
Clip Count: 8
Curves Pos: 0 Quat: 0 Euler: 0 Scale: 0 Muscles: 0 Generic: 0 PPtr: 8
Curves Count: 8 Constant: 0 (0.0%) Dense: 0 (0.0%) Stream: 8 (100.0%)
```

☑ Circle Collider 2D

Material	None (Physics Material 2D)
Is Trigger	□
Used By Effector	□
Offset	X 0 Y 0
Radius	0.5

Info

☑ Box Collider 2D

Material	None (Physics Material 2D)
Is Trigger	□

Overall main map scene view in Unity

Tools Used

<u>Gimp 2</u>: was used to create and edit images and sprites for the map and dialogue. Gimp is photoshop on a budget. Gimp is a free image editor that can be found here: https://www.gimp.org/

<u>GraphicsGale:</u> was used to develop the NPC sprites and animation sprites. It can be found here: https://graphicsgale.com/us/

<u>Visual Studios:</u> was used to create and edit scripts. This is a commonly used IDE that works well for creating and editing code.

<u>Unity:</u> was used to create the game. The included Unity libraries were used in the scripts. Unity is a great tool to create games with

<u>Git & Github:</u> was used as the main project repository. It was crucial for keeping record of project changes and making sure all group members were up to date.

Team Member Contributions

Ethan Dunham

Ethan was responsible for creating all the map sprites, dialogue, and the battlemap background. He created all custom map sprites, created the map layout, and added collision detectors. He also added in interaction with NPCs and chests. He created scenes for dialogue interaction with the NPCs with several different paths and options. He created code that allowed interaction with the player, improved the level system Katheryn added, and created variables to keep the game state current. Finally, he created a victory scene that showcases the player's party accomplishments after defeating the final boss.

Dylan Markovic

Dylan was responsible for creating the sprites for the Heroine, party members, enemies, bosses, battle sprites, and collectibles. He also animated the Hero and bosses. Writing code to control the player, control the camera, collect and store items in the inventory, and randomly generating enemy battle encounters were accomplished by researching the documentation. Aside from coding, needing to learn many aspects of the Unity software was very important. Understanding the animator and animation applications was crucial for the game to have motion, and being able to collide with objects, walk underneath tree tops, and be blocked by walls was necessary for the game to behave as expected.

Katheryn McHolick

Katheryn was in charge of setting up the basic screen flow of the game (main menu, new game creation, game map, save and quit, load), the battle scene and its mechanics, and the pause menu (character menu,level up, save and quit). She created a multitude of scripts from handling the player name input, to loading the character's stats into the Character menu, and even scripts to save and load the game. She single handedly built and scripted the battle sequences. At the beginning this required hours of online research on how to write C# scripts and how to implement features in unity. As the term progressed she gained the knowledge and skills needed to write C# code and work in unity.

Conclusion

Hydra worked together as a team to learn Unity, create a fun game, and remove any bugs or issues found. We worked diligently to achieve these goals within the allotted time. More time was put into learning how to do something in Unity than time put into the actual project. However, that is to be expected and is usually the case with any new technology that is learned.

By the end of the project, we were able to complete much more in less time. Overall, the project turned out nicely and we created a strong Unity foundation to build future projects on.

Sources

hydraRPGTitle - https://cooltext.com/
Pause menu scripts reference:
https://www.sitepoint.com/adding-pause-main-menu-and-game-over-screens-in-unity/
Battle scene and main menu background:
Bush: https://pixabay.com/p-575514/?no_redirect
Tree: https://cdn.pixabay.com/photo/2014/12/22/00/07/tree-576847_960_720.png
Grass:
http://maxpixel.freegreatpicture.com/Grass-Texture-Summer-Structure-Green-Meadow-Halme323571
General Unity Information:
https://docs.unity3d.com/Manual/index.html?_ga=2.47631215.978897069.1521251424-264015964.1515436911