

## Start up GUI:

The GUI that starts the game is implemented with the unity GUI. Details can be found in the Unity documentation under GUI Scripting Guide. Button scripts are attached to GUITextures. A GUITexture is a Unity GameObject. It is placed at a location on the screen. The script does everything else. Currently, we have a title, a start button and an exit button. Activating the start button will start a new version of the scene specified in the script. The exit button will exit the program, but not in the editor or in a web browser.

## Stage 1

### On Rails Mode:

On Rails Mode is implemented using [Unity's Navigation Meshes](#). Look in the Unity Documentation under Navigation and Pathfinding for details. A NavMesh is a plane that objects are allowed to navigate on. Complicated paths are simulated using waypoints. Waypoints are GameObjects with the Waypoint script attached to them. Waypoint objects provide locations for the Waypoint Path script. Objects that are to use the NavMesh must have a Nav Mesh Agent and the Waypoint Path Script with attached Waypoints.

### Game Components

[Waypoint](#)

[Waypoint Path](#)

[Follow a Waypoint Path on the navigation mesh](#)

[Follow a Waypoint Path in 3-space](#)

[Follow an object using the navigation mesh](#)

[Bind position to another object's position](#)

### Editor Components

[Waypoint Editor](#)

[Waypoint Path Editor](#)

## Throwing:

Throwing is implemented by attaching the Throw Object script to the camera. The script requires that an object be identified to throw, a key chosen to be the throw trigger and an object speed.

## Taking Pictures:

[Camera shutter animation](#)

[Camera mode toggle](#)

[Camera picture scoring](#)

## Editing Stage 1:

[Spawning Targets with GameObjects](#) # Stage 2

## Holding and Placing Items:

All objects that the player will be able to place in stage 2 need to have the [Placeable](#) component.

The [Inventory](#) component handles displaying the objects, and showing the confidence dialog when items are placed by the player. This component needs to be attached to the Player object.

## Stage 3

## Data Collection

## Other

## Tutorials

tutorials are instantiated prefabs found in the [tutorials folder](#) with the [Tutorial](#) component. The [GameState](#) has fields where sequences of these tutorials can be defined. When a game stage begins, the corresponding tutorial prefabs will be instantiated in sequence by a call to `GameState.BeginTutorialSequence()`

## **Terrain Stitcher:**

Unity has a problem where if two terrains are placed next to each other and one is raised, the other does not raise with it. The Terrain Stitcher scripts TerrainStitcher and TerrainStitcherWindow create a new tab in the editor. Clicking on TerrainStitcher... under the tab Terrain opens the stitcher. To use the stitcher, just drag the two terrains to be stitched into the open slots and click stitch. Terrain stitcher introduces a bug where Unity can not build the game. This is bypassed by simply removing the terrain stitcher and returning it once the game has been built. TerrainStitcher is a borrowed script.

## **Importing 3D Models:**

3D models tend to not come complete. Often the models are gray, and the color and textures are in a separate file. After downloading a 3D model, unzip the zipped file. Click the assets tab and go to import assets. Import the model and textures. Sometimes the model with have texture slots in the Inspector window. If this is the case, you may drag the textures from the unity editor folder into the empty box. Otherwise, you may be able to drag the texture directly on top of the object in the scene view. Sometimes the model might become textured by first importing into Blender or Maya and then exporting into a file format that Unity recognizes better.

## **misc.:**