Stainless Reality

Pub Brewery Asset Pack

Documentation

Contents

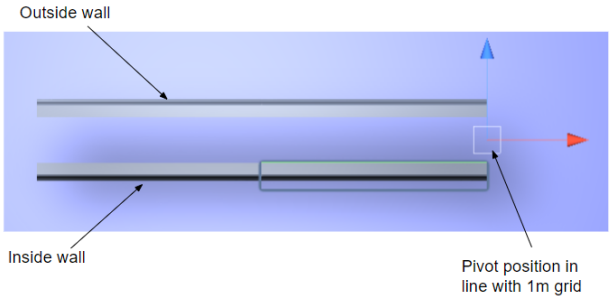
Introduction

Thank you for purchasing this asset pack. It was created with the intention of providing all the required assets to build a simple scene featuring a pub / bar, pub cellar, and micro-brewery.

Folder Structure / Organization

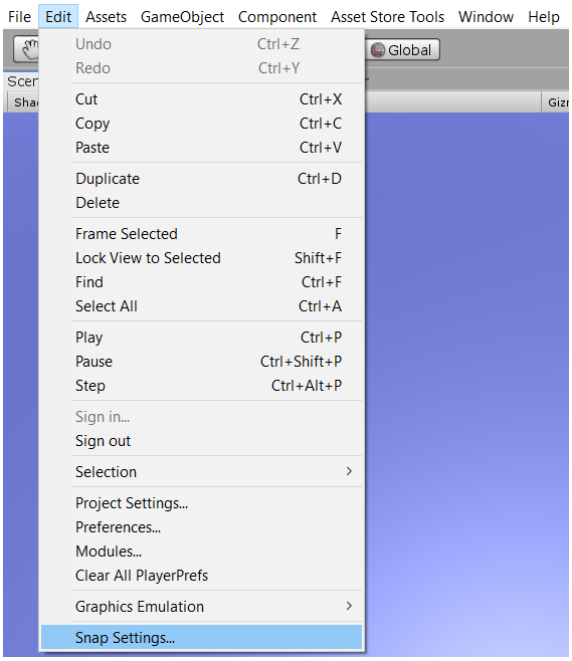
- PubBreweryAnimation
    - Animation Controllers for all animated objects. Animation clips are in the Models folder.
- PubBreweryMaterials
    - All materials used by the assets. Textures are in the Textures folder.
- PubBreweryModels
    - All the raw imported FBX files forming the basis of the assets. All animation clips are contained within the respective model.
- PubBreweryPrefabs
    - EquipmentFurniture
        - All prefabs which could be classed as equipment or furniture.
    - WallsCeilingsFloors
        - Prefabs for creating the walls, ceilings or floors of the environment.
- PubBreweryScenes
    - Contains the Example demo scene as well as the LineUp scene which serves to provide an overview of all assets. There is also a folder with baked lightmaps and data
- PubBreweryScripts
    - Scripts for the on-click animation and player controller functions.
- PubBreweryTextures
    - All textures used by the asset materials. Broadly speaking you will find one folder for each material. The materials themselves are in the Materials folder.

Snappable Assets



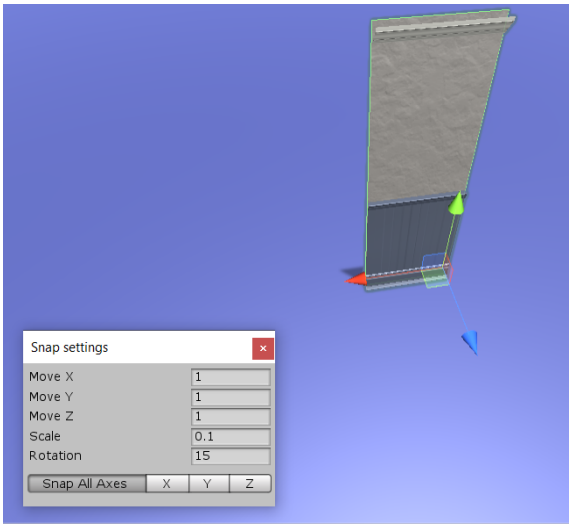
The wall, floor and ceiling assets (as well as certain furniture assets such as the bench and bar sections) have origins positioned to allow for easy 'snapping' and thus fast creation of interior environments. The all sections are based on a 1 x 1m grid.

The faces of the walls are offset from the origin by 100m, making a 200mm thick wall across two rooms / areas.



The process of adding and editing snappable assets is straightforward:

1. Click on Edit -> Snap Settings to bring up the Snap settings window
2. Check the snap increments. This pack was created with 1m settings in mind, but you may wish to change this for your particular application
3. Click Snap all axes, or select only the axis along which you wish to snap the asset
4. Now that the asset is aligned, hold down Ctrl as you move along an axis or plane to keep it snapped to the grid.
    a. Please note that Unity's snapping is not perfect and has a tendency to deviate as you move the asset more, therefore once you're finished positioning it is best to "Snap all Axes" again just to ensure perfect alignment

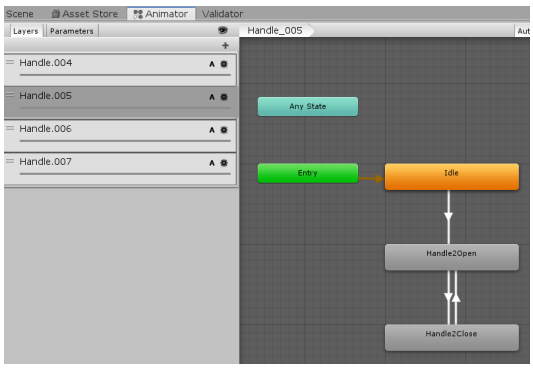


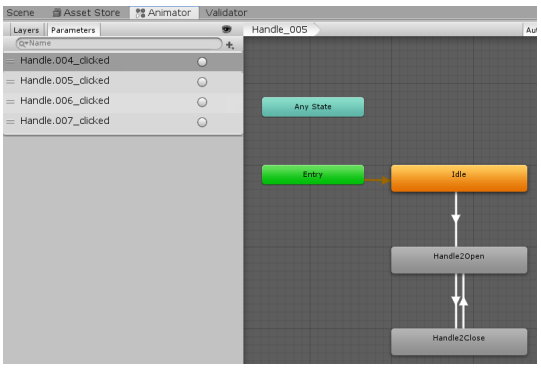
On-click Animations

If there is part of an object which looks like it should move, it probably does. This pack contains a single script which enables the on-click animation of any prefab you add to the scene. Simply create an empty object to hold the script or attach it to a logical existing object, such as the player. I will concede that the script could be more elegant, but my focus was on making it transparent, and ultimately on the 3D art.

The system works as follows:

- An object is clicked
- The script uses a physics raycast to identify the collider and object which was hit
- The script looks for an Animator component in the clicked object
- If it fails, it moves up and checks the parent, repeating this up to four times until an Animator component is found, otherwise it stops trying.
- Once an Animator component is found, it uses the collider game object's name to construct the correct name of the trigger parameter to be fired in the Animator Controller
    - Trigger parameter must be of the form "<collider's game object name>_clicked"
- An animated prefab may have several parts which are animated independently, but it uses one Animator Controller to control all of these. Each animated part has its own layer in the controller (weight = 1, blending = additive), its own trigger parameter, and usually two animations (e.g. Open and Close)
- The trigger parameter is fired and the animation plays
- Don't forget: if you mark an object as entirely static, it will not animate. Only select 'Lightmap Static' if you need to bake its lighting





PBR Textures & Materials

The vast majority of the assets use a normal map, ambient occlusion map, metallic texture with alpha channel representing smoothness, and albedo texture. The Textures folder holds all the maps and textures. They are grouped in folders for each material, and while many materials are unique to one prefab / model, some materials are generic seamless textures (for use on walls or floors for example.

Some textures are shared by multiple prefabs, such as the cove, skirt, or light fittings for example.

Light Fittings

The prefabs LightWall and LightPendant use the LightFittings material, which has an emission mask set up for the light bulb and can be used for baked lighting in a scene (as seen in demo scene).

Demo Scene

The PubBreweryExample scene is a compact scene made with the aim of illustrating the potential use for all the available prefabs, and demonstrating how the snappable elements can be used to build an interior.

The lighting is entirely baked GI using the Progressive GPU lightmapper, and a handful of reflection probes. It also includes a simple FirstPersonPlayer object for exploring the environment and object interactions.

Player Movement

The FirstPersonPlayer object was made using the fantastic video from tutorial from Brackeys: https://youtu.be/_QajrabyTJc

The video provides a far better explanation of the script and design than I could attempt here.