**Unlisted Games Content Implementation Test**

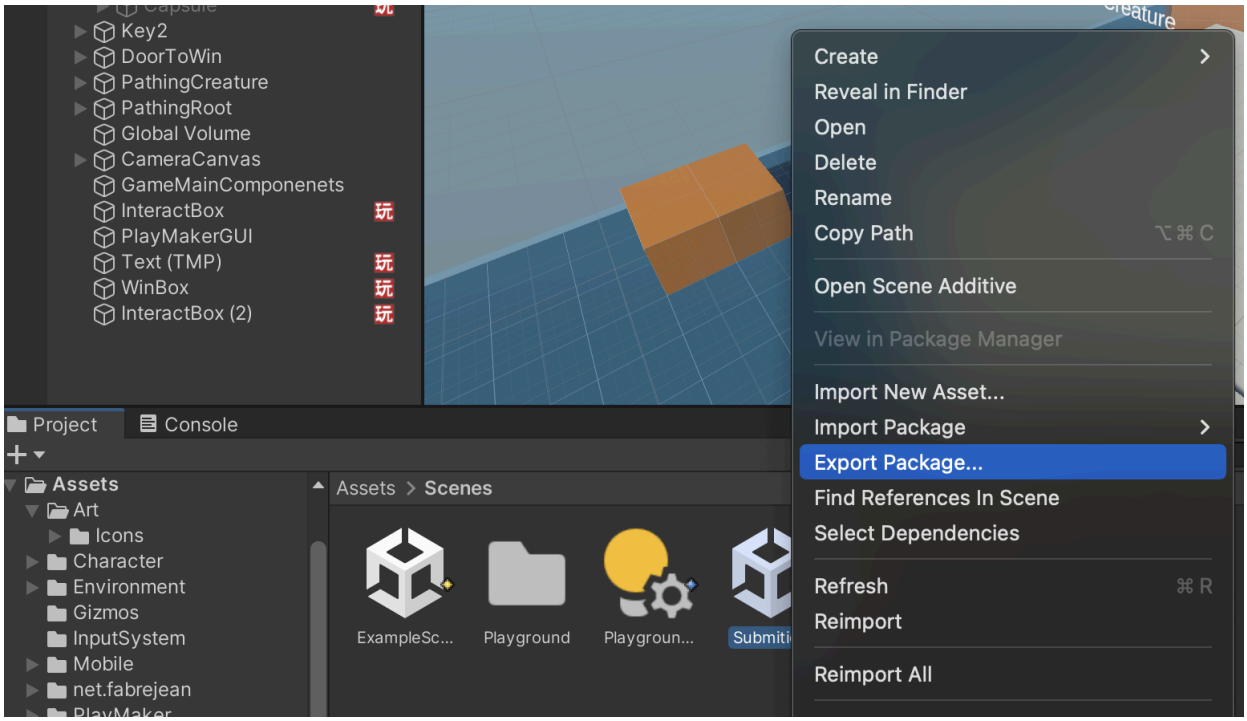
**Overview:**
The purpose of this task is to test your ability to design and implement game content into Unity3D. First understand the systems available to you by looking at the example project and reading this doc in full. You are then asked to design and implement a puzzle level of your own into a scene using Unity3D. You will have some mock objects, some provided simple building tools, ProBuilder for level creation, and Playmaker visual scripting. Feel free to reach out to us back on email if you have any questions or issues.

**Submission Criteria:**
- The level should utilize at least:
    - A Camera feature (explained later in this doc)
    - Have an end state which enables a “GameWin” UI. (explained later in this doc)
    - Have a “polished/juicy” element
- For your puzzle **you should take this in a unique direction and avoid using similar puzzle elements and prefabs of the example.** Avoid a final key door, and the mechanic of taking a picture of a creature which pops out a key, etc. Please invent your own mechanics.
- We want to have a challenge that creates some objective for the player to get to the end state.
- **3d art assets are not important for this test**, imagine this as a mock implementation from a visuals side, but the gameplay logic and **feel/joy** should be playable and complete to ensure your vision is clear.
- No external guides should be necessary to complete your level and all information required to beat it should exist within the level. **The player's knowledge expectations at this point would be they know how to move, how to use the camera tool, and the interact key.**
- We want you to stay in 3D and use space, but other than that, we want you to not reuse the elements that exist in the example scene, and to explore your own creative ideas.
- Bug Free

**Rating Criteria:**
We will be judging submissions based on the following criteria:
- The unique use or creation of camera mechanics
- The ability to give a sense of polish (**juice**) to the feature
- Environment layout and design that is clear and passively tells a story
- Bug free
- Understandability of puzzles
- Fun & Delight

**Submission Instructions:**

1) Upload your submission scene using a unity package and provide a link to download it in the email.  To make a package of your scene, right click and select Export Package. **Make sure to Include Dependencies** and the size will likely be 70-80 MB and too big to include as an email attachment.




2) Provide a brief explanation of what you would want to do if given more time, and had access to additional engineering & art resources.
3) **Include a video recording** of you playing through your level from start to end.

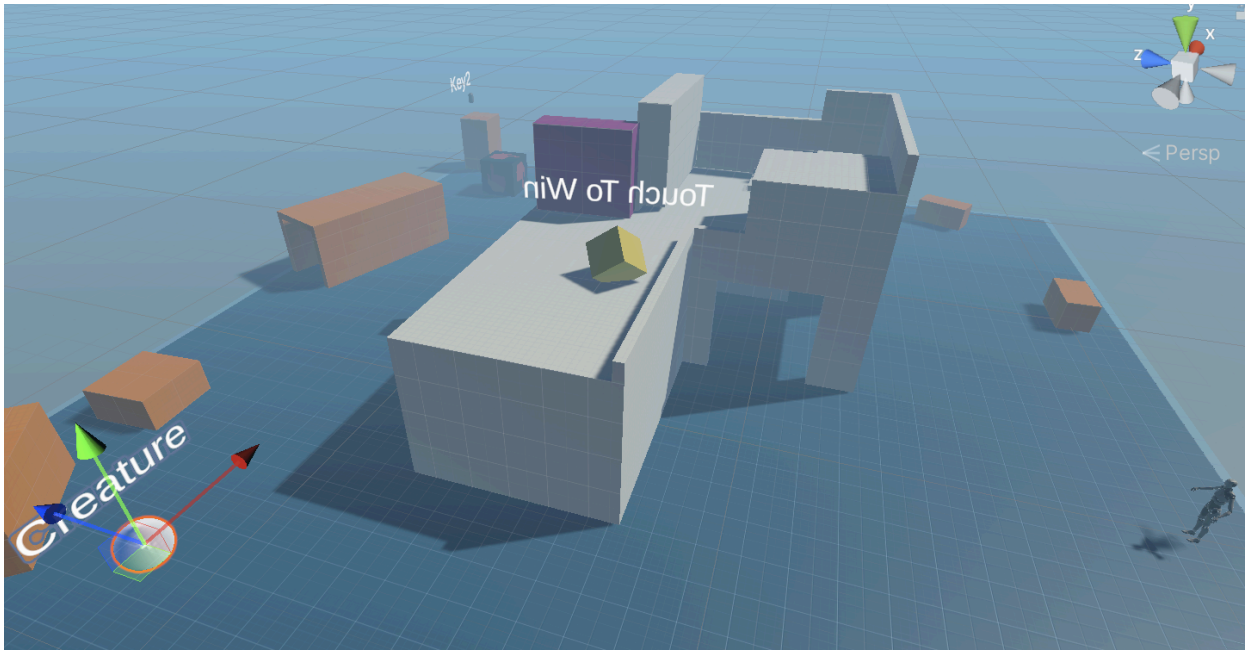
**Download The Project:**
Using Github, clone this project to your computer or download the Zip package and open it up: https://github.com/rgeorgeoff/SnapQuestContentTest

You will also need Unity version: 2021.3.21f1 which can be found in unity hub or via Unity's website.

Make a new scene in the scene folder called "[YourName]_Submission" and that is the scene where you will make your level.

**Overview of The Project:**
The project you downloaded is based off of a 3d sample project from unity. We have added Playmaker (wiki), A* pathfinding (wiki), Probuilder, and a limited subset of scripts from our game that allow for rudimentary Camera picture taking, and simple touch interactions.

Going to the scene "ExampleScene" in the scenes folder is where you will see an example scene we set up for you.




**Goal:** The goal of this Example Scene puzzle level is to collect 2 keys, and open the purple door to touch the final Win diamond which will pull up a “GameWin” ui.

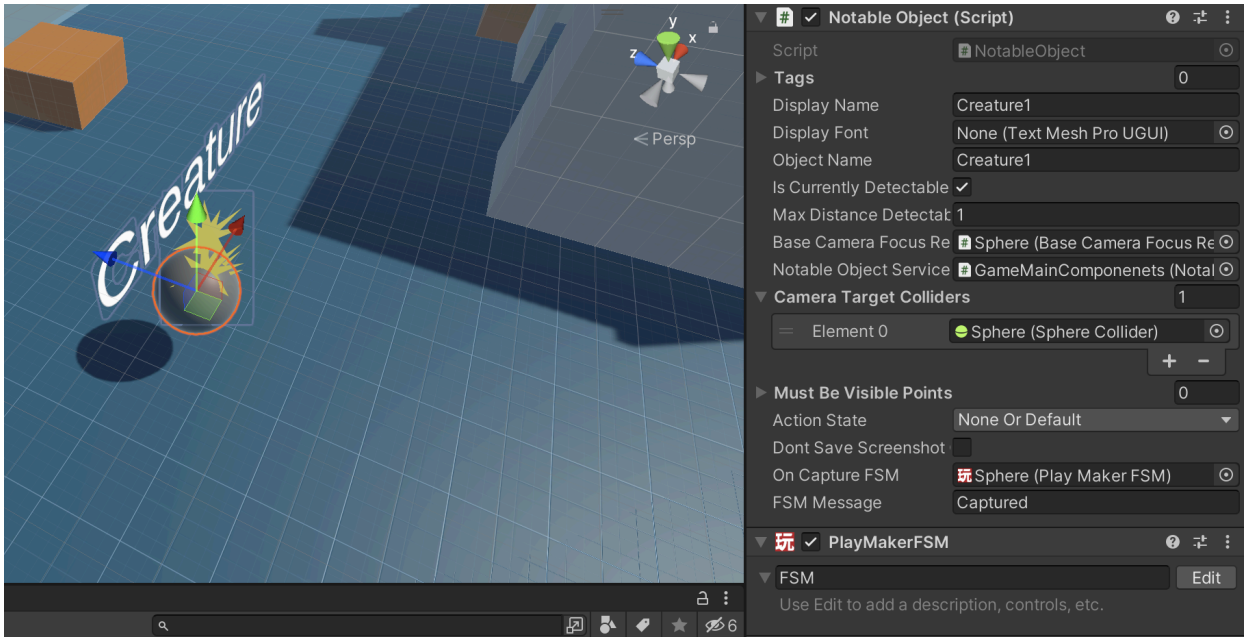
To collect the first key, you will walk (WASD) to the left, pull out your camera (C) and take a picture of the moving creature. Then continue down that path to the pushable blocks, interact (F) to push the big one so you can jump (SPACE) up to the second creature, and get Key2. After both keys are collected the purple door will lower after interacting with it, revealing the final win box. Interact with it to enable the GameWin UI.

This is a very simple example, and we would hope yours will be completely different and unique. Feel free to change all aspects of the scene, add/remove assets, nothing is required to stay as is, and in some regards the more different the better. The interactions and the camera trigger effects can do anything you can imagine and feel comfortable to implement.

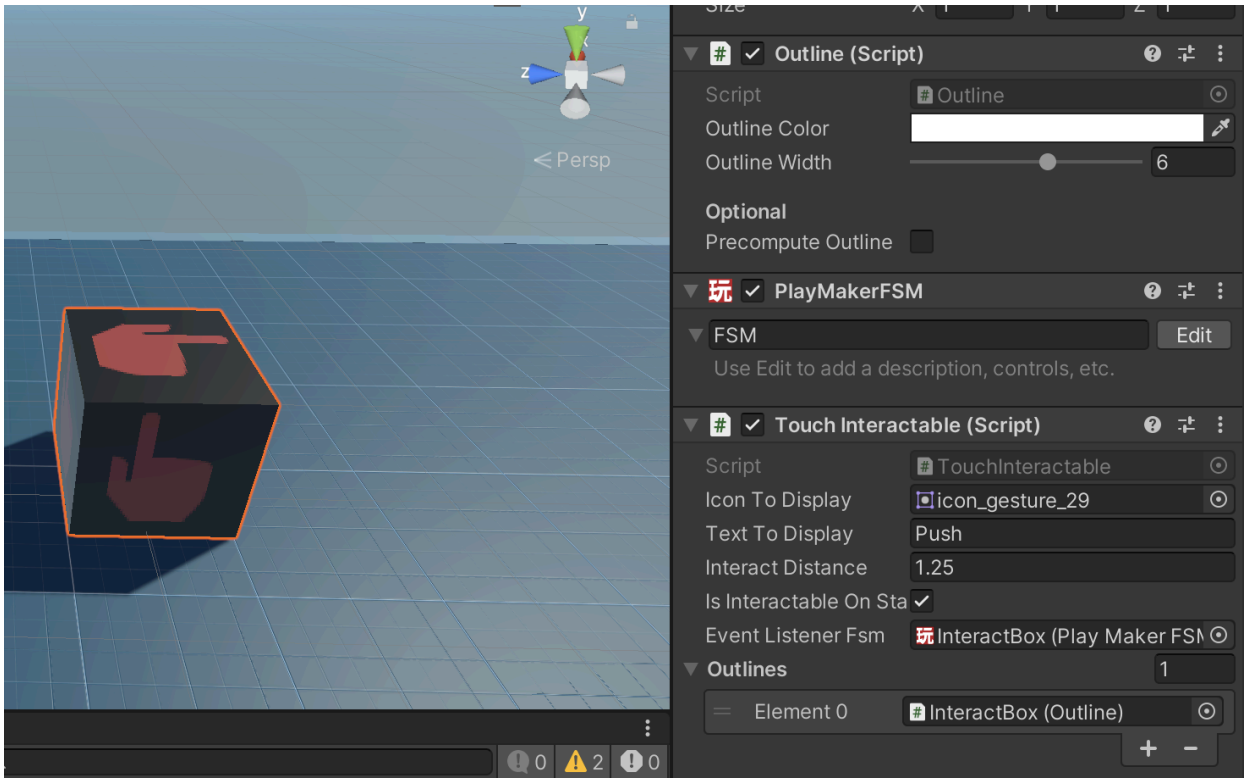
**Additional info for how to use our custom Camera and Interaction behaviours:**

**Camera:**
Pressing C will bring up the camera which can detect Notable Objects. When a photo is taken (by left clicking the mouse) on an object with the component NotableObject.cs on it, the playmaker function "Captured" will be called to the OnCaptureFSM that is attached to that NotableObject.cs component. In our example you can see by opening the FSM that this will trigger the playing of a particle effect, and the pooping out of Key1. 💩

Ensure to have a collider attached to the object you want to be hit by camera detection and assign it to the “Camera Target Colliders” field on the Notable Object component.




**Interactables:**
Pressing F near interactable objects will call the "StartInteract" event for the attached PlaymakerFSM. Adding the TouchInteractable.cs component, you can set the icon and text to display when the player is close, modify interact distance, and change if its interactable on start. Ensure to add a PlaymakerFSM and an Outline.cs component add assign that to the Interactable as well. After an interaction happens, Playmaker takes over. You can see another examples of Interactions in the example scene on the Key objects, the Purple Door, and the Win Diamond.




**The Juice:**
The purpose behind this part of the evaluation is to get a sense of how capable you are of polishing a project to a place where it can be released. It's often said that the last 90% of a game development is polish, and all games need a good amount of juice to feel good. Its all about game feel and feedback. I can always recommend this talk from 14 years ago which is still extremely relevant to todays scene.

Juice it or lose it - a talk by Martin Jonasson & Petri Purho

- Particles
- Tweening animations
- Use of colors on player feedback
- SFX/VFX

Just make your feature feel nice!~

**End State:**
Just to make a clear end condition we added a simple You Win! UI for the end of the level. You don't have to use this specific object, but as long as there is a clear end state, then that works for us!


**Extra Tips:**
Feel free to change mouse sensitivity on the player armature object in the ThirdPersonController.cs script and the camera sensitivity on the CameraToolService (located on GameMainComponents) since different systems have very different sensitivities.

It would be good to look at the Playmaker editor for any of the gameobjects with the Playmaker Icon but in particular a good example is located on the DoorToWin->Sphere. This is where most of the puzzle logic sits including the state of keys collected.


If you feel lost, play through the Example scene and look through the components on the objects within that scene. And again as a last resort feel free to reach out to us with any questions, we look forward to seeing your creativity!