Gathering Gold

Table of contents

Changelog

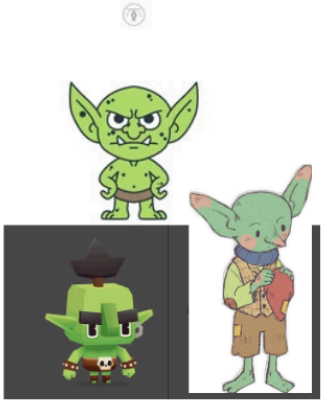
Who	What	When
Engie	Started the GDD and added the mechanic picking up info	18/12/2025
Peyton	Added remaining mechanics, edited existing mechanic sections	23/12/25
Peyton	Added sound manager section, header for shader section, images from miro to support mechanics, concept art section, level map section	23/12/25

Game Overview

Gathering Gold is a co-op puzzle game for two players in which two mushroom goblins are attempting to steal a sleeping dragon’s treasure hoard right out from under him! Solve puzzles and manipulate mechanisms as a team to load as much treasure from the cave into your minecart as possible without waking up the dragon.

References/Concept Art

Moodboard

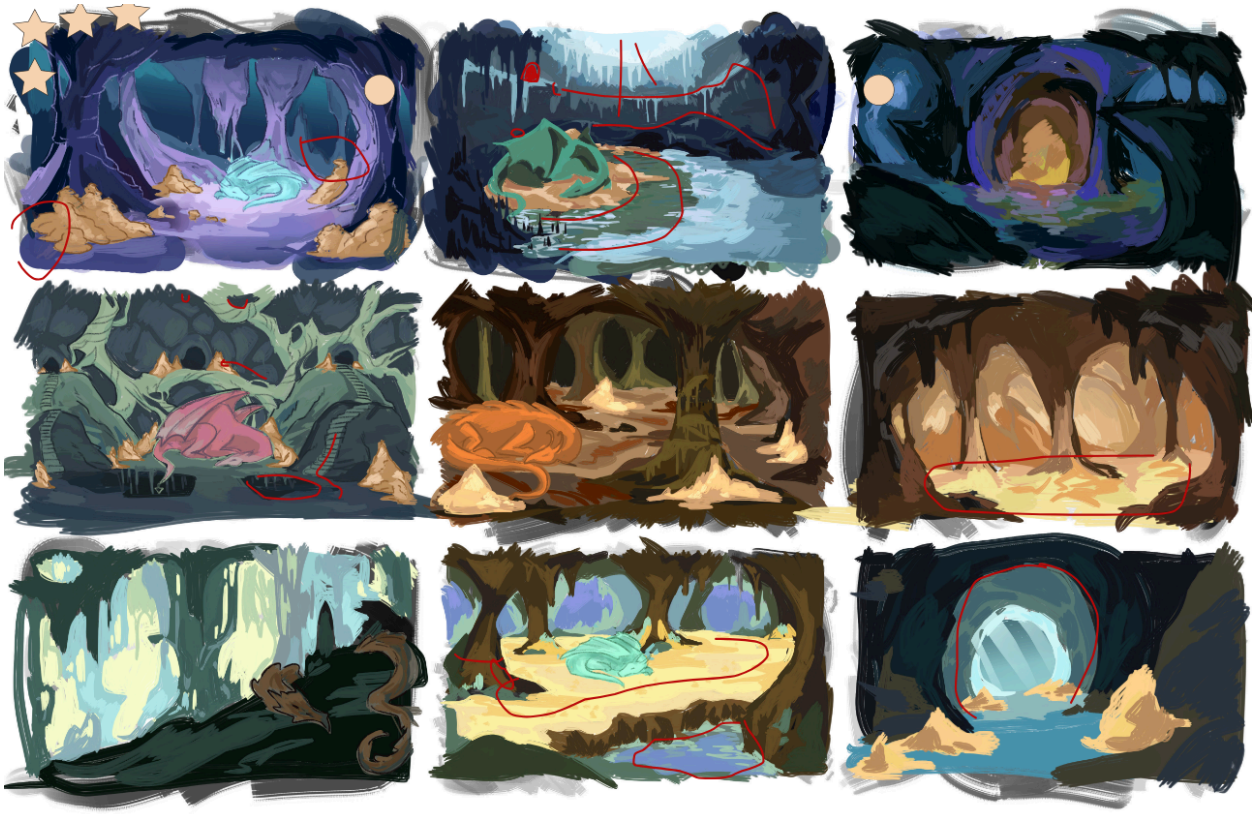


items

- coins
- bread
- jewels
- bowl of fruit?

Character Concepts

Environment Concepts



3D Environment References

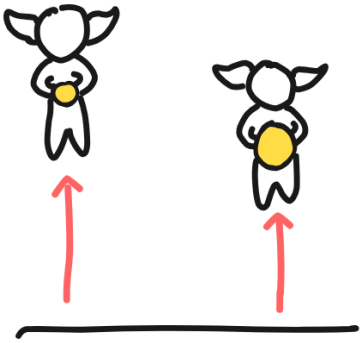
Mechanics

Movement

Each player can move independently, and the camera will adjust to keep both on screen. The game uses split keyboard controls for couch co-op.

- Character A: WASD to move, space to jump
- Character B: Arrow keys to move, right shift to jump

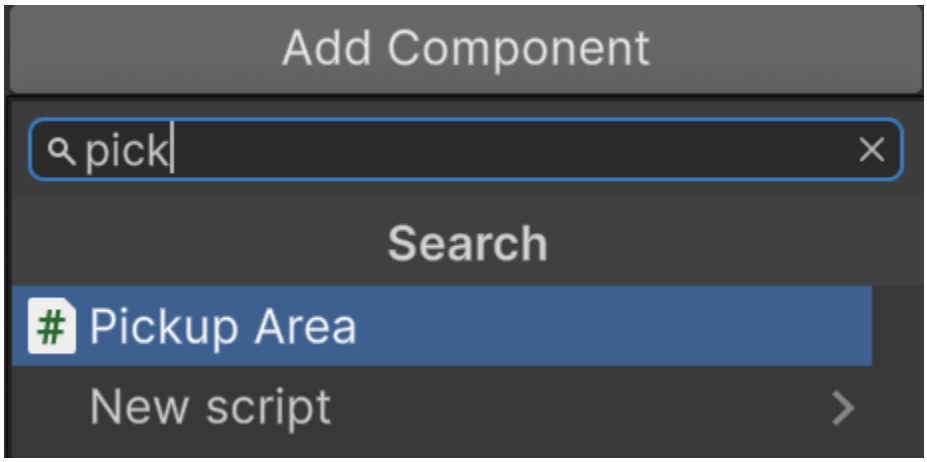
Movement is impacted by carrying treasure (jump height is halved)



Carrying Treasure

Items can be picked up and set down again by pressing left shift (Character A) or / (Character B). The player can walk and jump while carrying an item. While holding an object, jump height is reduced to half.

To make an item pickupable, add an empty gameobject to it and add the "PickupArea" script to it:

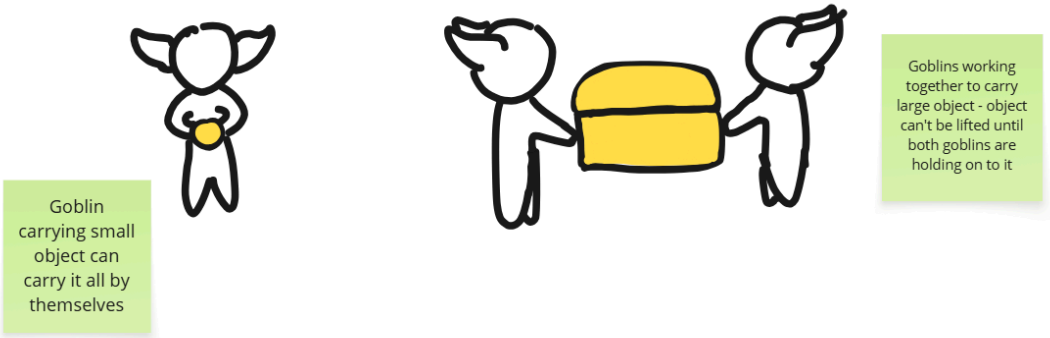


- it will add a spherecollider to the child, make sure you set it to trigger and change the radius so that it's slightly bigger than the item itself that collider is the area the player has to be in to pick it up
- to pick up items, use LShift for the left player and / for the right player
- player jump height is also halved while holding an object, and you can drop the item by pressing the same key again

How to give things weight,
If it's a Player:,
1) Change the GameObject layer to 'Player.'
2) Add the 'CarryWeight' script on the player. Set the desired weight in 'Object Weight' to HALF what you want it in the Inspector under the script; default weight set for them is 2.5. Current bug is from the fact that the player has a collider AND a character controller, so the plate is counting them twice.
If it's an Item:,
1) Change the GameObject layer to 'Loot.'
2) Add the 'CarryWeight' script on the object. Set the desired weight in 'Object Weight'; default weight is 5.

Treasure can be carried by two players if it is too heavy to be carried by one.

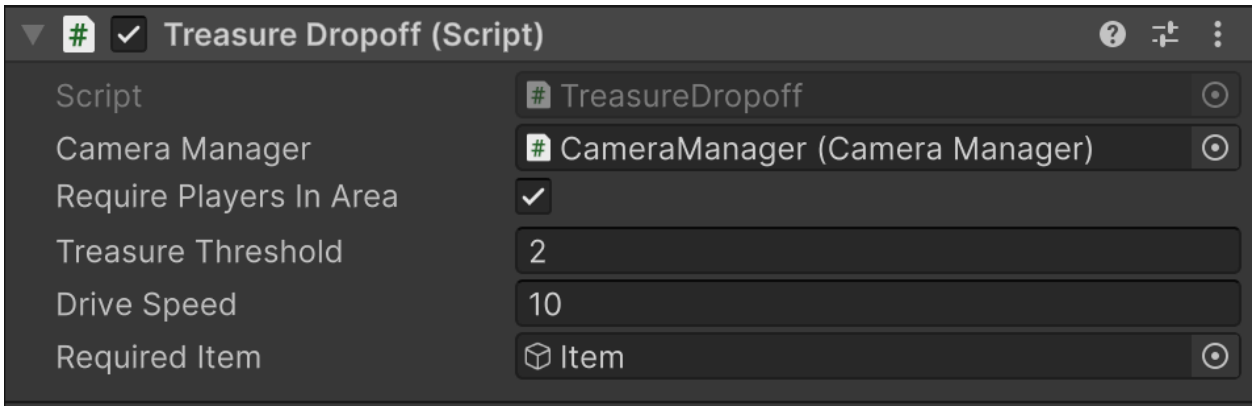
How to assign an object meant to be lifted by two players:
1) assign an object as a pick up.
2) in the Inspector, set the object to Heavy.
3) set the cutoff Distance float to something appropriate to the object



Treasure Dropoff

Players collect treasure by placing it in the minecart, designated as the Treasure Dropoff. The player must place the required treasure (either by amount or specific item) in the Treasure Dropoff to win the game.

- Treasure Threshold: The number of treasure items required in the Treasure Dropoff to win the game. Once the required number are placed, the game will end.
- Required Item: Set a required item to disregard the treasure threshold and set the win condition to that specific item instead. Once the Required Item is collected, the game will end.



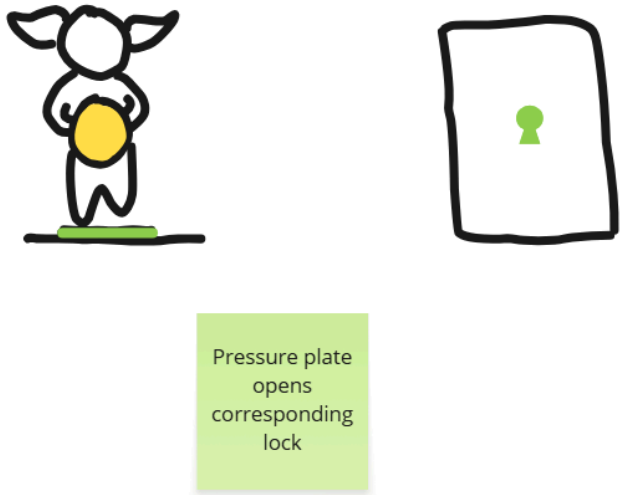
Pressure Plates

Pressure Plates are large buttons activated by placing something with the required weight on top of them. They can require the weight of a player, an object, or a combination of the two.

Pressure plates can be used to activate other mechanisms (e.g. doors, ramps, etc).

Doors

Doors can be used to block certain areas, and be unlocked with pressure plates.



How to assign a door to unlock with a plate:
1) pull a Door prefab to scene.
2) pull your preferred pressure plate prefab to scene.
3) expand your plate prefab in the Hierarchy so you can select the one suffix-ed by "Block."
4) Select the actual door object by going Door>Pivot Point>Door.
5) In the 'Door' script, drag and drop the "Block" object in the Door script, asking for the "Unlocking Plate" in the Inspector. Your door should now be linked to that plate.

See Saws

See Saws are boards balanced on a base that tilt when weight is applied to one side (either the weight of a player or an object). Rotation and movement is constrained on all

axes except for the x axis to allow for tilting in one direction. See Saws can be used as ramps or as bridges depending on how weight is applied.

Placing a heavy object on one side of the see saw turns it into a ramp

Placing a matching object on the other side balances it, creating a "bridge"

Sleeping Dragon

The antagonist and main obstacle of the level is the sleeping dragon. He is lying in the center of the cave. If the player stays within the dragon's detection range for too long, he will wake up and the players will lose.

What makes dragon wake up

- Noise
    - Running on a pile of gold
    - Area around the dragon the dragon is really sensitive to waking up through accumulation of points
    - (mine cart is close to the dragon)
    - If player is carrying something its louder
    - Time spent in danger zone dragon wakes up

Particle effect signifiers will identify if the player has woken the dragon:

- 'ZZZ' particle effect if the dragon is sleeping
- '!' effect if the dragon has noticed the players and woken up

Level Map

Sound Manager

The Sound Manager allows sound effects to be played based on animations, as well as called from other scripts.

To add a sound to the Sound Manager:

1. Add the name you would like the sound to be identified by in the inspector to the list in the Sound Manager Script

```
4 references
public enum SoundType
{
    LAND,
    WALK,
    RAMPDOWN,
    PLATEDOWN,
    PICKUP,
    DROP,
    KEY,
    FLAME,
    SEESAW
}
```

2. Drag the Audio Clips to the corresponding slots in the inspector (add another list item if you need to)

3. To time the sound with beginning/ending an action that has an animation: Select the animation state you'd like to have sound in, and click Add Behaviour in the

inspector. Choose PlaySoundEnter or PlaySoundExit depending on if you'd like the sound to play at the beginning or end of the animation, and choose the sound you want from the dropdown list.

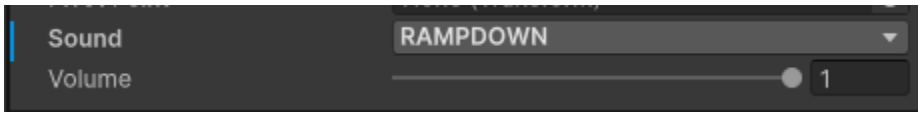
4. To use the sound manager from another script (This is not quite working right so edits may come later): Add these variables at the top of the script to reference the sounds

```
37      // stuff for sfx
38      [SerializeField] private SoundType sound;
39      [SerializeField, Range(0, 1)] private float volume = 1;
```

And add this line into whatever logic you're using to determine when to play the sound

```
103                 SoundManager.PlaySound(sound, volume);
```

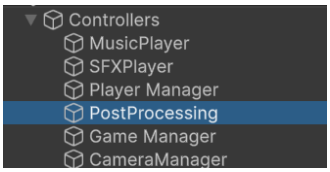
There will now be a sound dropdown and volume slider added to this script in the inspector, select the sound you want to play from the dropdown.



Shaders and Effects

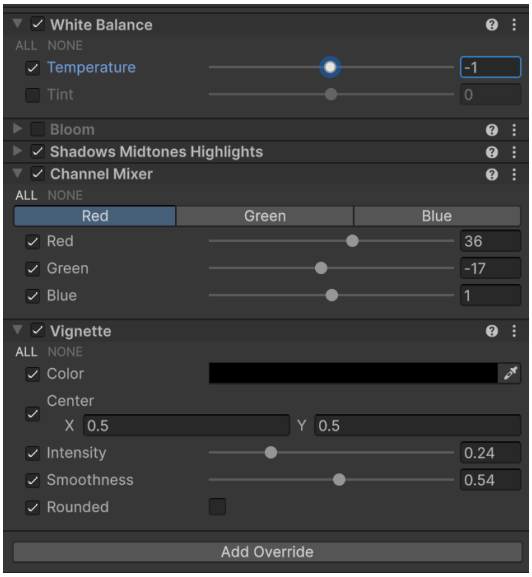
Shaders are graphical processes that render how a game looks. For Gathering Gold, the visuals come two-fold:

Post-Processing Effects



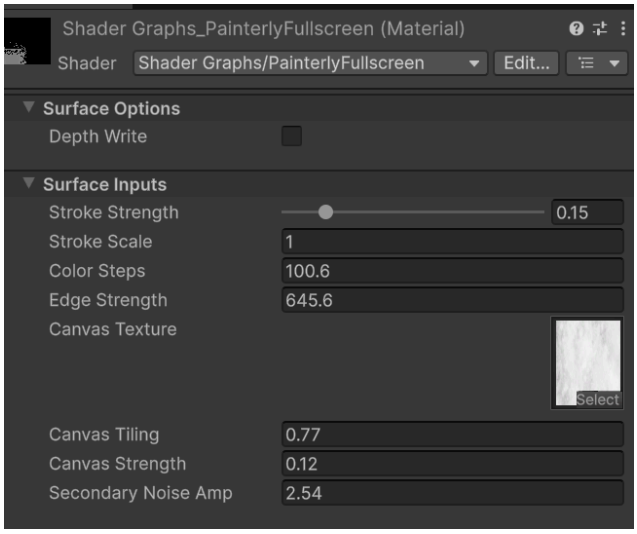
In the editor, you'll find the Post Processing object under the Controllers. In its inspector, you'll find how these various dials affect the aesthetics:

1. White Balance: Affects the temperature (how blue/orange the whites tend to be).
2. Shadow Midtones Highlights: These affect how strongly each of these value/saturation properties appear. Increasing shadows makes the darks darker, and the highlights make the bright brighter. Midtones are all the values between intensities.
3. Channel Mixer: These are the color balance effects which dictate how a hue appears through this filter.
4. Vignette: These are the darkened edges by the corners and sides of the camera.



Painterly Shader

The Shader can be modified through selecting the Inspector of the Shader Graph, found in the Shaders folder. Here is what changing these values do:



1. Stroke Strength: How defined each block of color/value is (as assessed by the logic found in the graph). Lower values create smudgier 'strokes' while higher values make it more uniform and akin to how it really is.
2. Stroke Scale: DO NOT TOUCH!! This is how large the shader stroke effect is. Its perfectly fit to the camera right now.
3. Canvas Texture: a 2D texture that masks the camera to provide a 'canvas-like' texture.
4. Canvas Tiling: Affects how much of the canvas texture to show. Values less than 1 show a part of the entire image with Values greater than 1 have them tile on repeat.
5. Canvas Strength: How strongly the texture appears on the shader. Higher values make the texture mask appear more strongly.