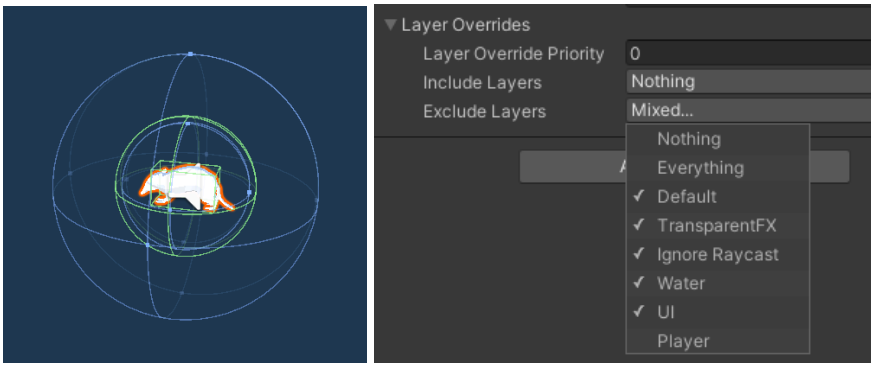
An Aussie Adventure - Update

Progress Update

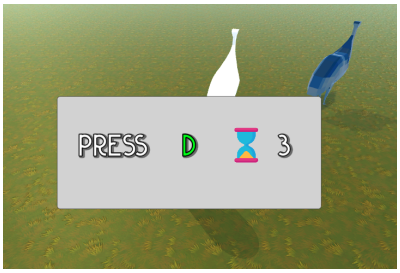
I eventually settled on using physics for the player's navigation of the world. This is done by using a rigidbody with continuous detection and interpolation to aid with smooth movement while hopefully minimising the jitteriness that can occur. This meant I could utilise the physics system's gravity and force pulses to manipulate the player. The player's collider is used in conjunction with other trigger colliders, such as that of the money prefab. It is not yet sufficiently implemented but I want some of the money to be accessible as a human, while each transformation would gain access to different areas of the map. Money is managed in a singleton progress script that also maintains whether the player has regained their transformative powers and which animals they've unlocked.

Interacting with the wild animals is integral to the game and manages to incorporate most of the learning topics. Naturally, each type of animal is prefabbed to allow for quick reproduction and a single place for updating all values simultaneously. As seen in the prefab image below, each of the wild animals includes multiple green colliders, the box collider representing a simplified collider for the animal's body with the outer sphere collider acting as the trigger collider for the interaction minigame. The animal's trigger collider is set to exclude interactions with all layers with the exception of the player, hopefully improving general performance. When the player is within the collider and presses the interact key, the transformation minigame commences. The blue spheres represent the distance the animal's audio source is audible. Drawing from a list of randomised audio clips that were prepared by cutting down a single longer clip in Audacity. These serve as the sounds of the animals moving and the audio sources were modified to appropriately reflect them. A small bandicoot would make much less noise when moving compared to the much heavier and less subtle movement of a kangaroo.



The initial action plan was outlined on the basis that the transformation and interaction systems were progressing well and so would be naturally finished first. It's perhaps appropriate that this 'triple A title' would get stuck as I struggled to implement the envisioned ideas. Through some trial and error, I eventually built the minigame that can be triggered to unlock the animal transformation. The minigame combines many elements of learning. Trigger colliders, a canvas that is set as active

for visual interaction, a one shot success sound effect, ideas raised about using a singleton to store player progress, and independent research on using coroutines. Though the visual aspect is currently *underwhelming*, the core idea could be expanded to include more button prompts or more stages requiring success to progress.



Further demonstration of the UI and Scenes can be seen in the transformation menu for swapping between forms and in the main menu. The main menu and the credits are both canvases sat in world space, as the camera will rotate to focus on them when selected. Using a log pile from an asset pack and a simple gradient particle effect, a campfire was created to demonstrate further research. This was then combined with particle light effect, so that the particles display the effect of a simple spotlight. The transformation menu needs reworking as it is hard coded rather than using more intuitive elements such as the Unity buttons. The menu draws from the player's progress and updates dynamically when they unlock the new transformations: changing the colour from black to white and replacing question marks with the animal's name.

Remaining Work

I feel audio is the most underutilised and underdeveloped in the prototype. Though there are audio cues for menu decisions and the animals movements, it feels like there's still very little life in the game. One thing I want to focus on is finding some ambient sounds played throughout the game, while also finding a few more effects to push assets a little further, such as a crackling fire sound effect. Further demonstrations of this could involve using more mixers and colliders to implement different ambient sounds for different areas. For example, I found a Kookaburra call that I could play infrequently in an area of densely populated trees.

Combining methods used in the transformation menu and the minigame, I could implement the rudimentary shop for the player, thus allowing them to gain access to the required potions to progress the game. The player progress can be reused to manage the availability of the items within the shop. As the items are tied to the story progression, they are effectively one time use and the UI should be updated to display this while the player should also be prevented from repurchasing them through code. This process could display snippets of story above the characters heads.

Currently, the transformations are instant, but I would like to try experimenting with using scale to fake a transition from one state to another. Once chosen, the player's transform could scale to nearly zero, swap to the chosen transformation, and then rescale back to the original size. During the transformation, their rigidbody could be set to freeze on all axes, their transform raised an arbitrary amount and then unfrozen. This should hopefully alleviate potential issues with clipping due to the size of the models, such as when transforming from a bandicoot to an emu.

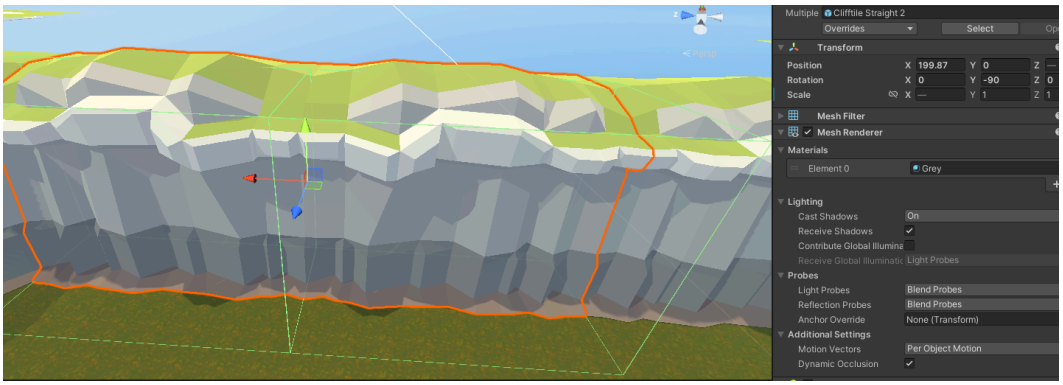
As of yet, there's not been much in the way of implementing any of the stretch goals, but I had planned to evaluate the viability of these in the coming days. As it covers a key learning objective, I could include the evil shaman's lair as a scene change to display understanding of Unity's DontDestroyOnLoad, even as a proof concept. Finally, though not part of the learning directly, I want to establish the reason behind the wildly inconsistent frame rate as it's likely something I've incorporated without fully understanding the ramifications of.

Changes

A superficial but important change was made when I discovered lots of low poly packs on the Unity Asset store from a now defunct studio. While I outlined using the Unity terrain pack and its included vegetation, BrokenVector's packs have a consistent art style and include swappable palettes, allowing me to mix and match to have a more consistent aesthetic throughout the game. Furthermore, the assets I created in Blender for the project fit into this style neatly when compared to the more realistic aesthetic of the Unity models. I may utilise the colours used in the packs when attempting simple texturing of the models as I sadly lack much intuition in the way of colour theory. These assets can be seen used in the previous main menu screenshot. Culminating in a much more coherent art style throughout the game while also being more in line with the target audience that I had outlined in the design document.

As the animal system was already nearly completed it was expected to be the first completed on the initial report. However, the system was inevitably redesigned to make them feel a little more alive. I researched coroutines and wrote a function to reflect this ambition. Animals now perform simple randomised exploration and will return home if they've strayed too far. Additionally, they now react to the player and try to plot the best direction of escape. This will require a bit of reworking as they can currently get soft-locked trying to return home while the player is in their way, thus making them become stuck in the escape routine.

With the animals now correctly exploring, I learned the hard way when utilising the cliff assets by setting the x scale to minus one, effectively flipping the model. It felt like a clever work around to create a different asset to avoid repetition, which proved shortsighted as it would confuse Unity's collider system and Unity would force bad collider geometry. This meant the animals could disappear off through the inverted cliffside and would often get stuck on their attempt to return. For now, I have simply reverted all of them back to the correct scale. If I feel the landscape becomes too repetitive, I could make new prefabs with the supplied models, inverting the scale of the model rather than the completed prefabs in Unity's scene editor.



Action Plan

Date	Task	Notes
4/4/24	MVP report due	
6/4/24	Implement the shaman interaction to unlock transformations and the end game potion.	Using interaction from the transformation minigame as a starting point.
8/4/24	Adding the other outlined areas of required work such as transformation 'animations' and audio	
10/4/24	Review and attempt to incorporate any of the viable stretch goals.	
14/4/24	Incorporate final topics of learning	
18/4/24	Stop active development and polish what's already in the game.	This will include throwing in more environmental assets which are already collected.
25/4/24	Game prototype and video presentation due	Date given as absolute deadline but would naturally aim for earlier