

SPY QUEST - COMS BC3997 Final Project

Aditi Dam, Luci Feinberg, Luiza Leschziner, Shivani Patel

Overview of the Game Mechanics

- Player Controls
 - Player Movement - right (←), left (→)
 - Player Jump - Space
 - Coyote Timing Jump: allows the player to easily jump & move through the playforms
- Players must avoid obstacles the wizard guards. Obstacles include: spikes and fly traps

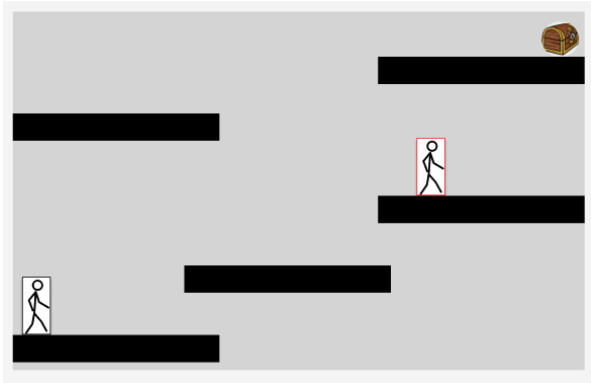
Objective of the game (**WIN**): collect all keys from each level to open the door to progress to the next level. At the last level, retrieve the crown and reach the door to win the game.

Avoid obstacles and wizards (**LOSE**): You should not let the guards spot you which utilizes an AI component through ray detection. If so, you are captured and must restart the level. You must also beware of all obstacles: fly traps and spikes. Collisions with these specific objects will result in loss.

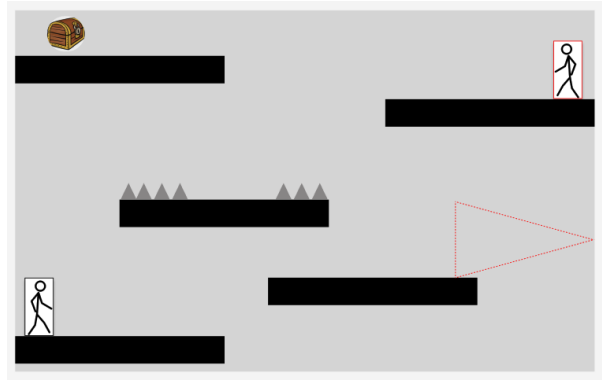
Design Process/Figma Mock-ups

We did a very simple mock-up of what we wanted our game to look like to ensure that we were on the same page before taking our game to Unity.

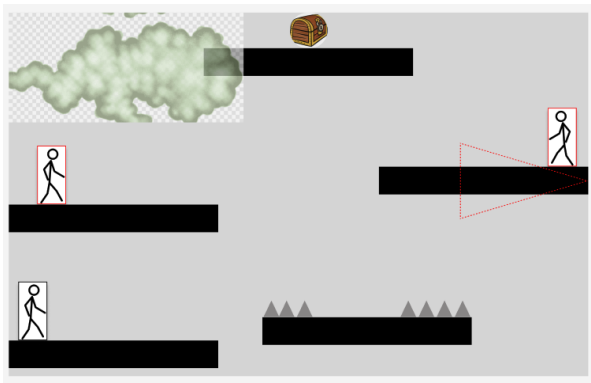
Level 1



Level 2



Level 3



Division of Work

Shivani and Aditi focused on movement mechanics, AI, and picked out all of the assets to add to the thematic feel of the game. Luiza used figma to outline levels and worked on the UI of the menus, buttons. Luci focused on implementing juice effects, animators, and UI with Luiza. The specific jobs done are as follows:

- ☐ Initial Level Design using Figma mock-ups - Lu
- ☐ Outline the game scenes - Shivani / Aditi
- ☐ Splash screen & instructions - Lu
- ☐ Win game screen + effects - Luci
- ☐ Collaboration environment using plastic SCM - Shivani
- ☐ Unity sprite assets selection & addition - Aditi
- ☐ Guards with AI sensing - Shivani
- ☐ Basic movement of the Spy - Aditi/Shivani
 - ☐ Coyote Movement - Luci
- ☐ Key collection - Aditi
- ☐ Opening/door mechanism - Aditi
- ☐ Collision detections for obstacles - Shivani/Aditi
- ☐ Choosing theme/sprites from Unity and positioning it on the scene - Aditi
- ☐ Restart level if Spy falls off platforms - Shivani
- ☐ 2D Semi-fixed screen Camera Movement - Shivani
- ☐ Juice Effects - Luci
 - ☐ Screen Shake on collision with obstacles
 - ☐ Player Feedback: cause and effect
 - ☐ (jump sound, Collect key sound, Door open sound, winning sound)
 - ☐ Door Open Animation
 - ☐ (transition flow)
 - ☐ Coyote Timing: Buffering when jumping
 - ☐ Game Music
 - ☐ Wizard Dialogue when spotting Spy - Aditi
- ☐ Home Button on Levels - Shivani
- ☐ Write up - Everyone
- ☐ Video demo - Lu

Description of Playtesting & Debugging

Debugging ranged from playtesting the game to utilizing console outputs. We also heavily relied on the scene view when the game play was 'on'. For example, when we were having issues with camera tracking, we referenced the scene view to track where in the canvas the main camera was on in respect to the entire map. This assured us that we were not tracking the wrong sprite, unintentionally modifying scope of view, or encountering issues with layering. The Unity console was also very helpful through the development process to provide us feedback on the states of our sprites, class variables, and object tags. The Unity console was crucial in order to nail down the single jump using coyote time for the movement of the spy. We used the debug log to signal that indeed collisions with "obstacles" indeed triggered before advancing to restart the scene. This provided feedback on whether the physical space was the problem or rather the C# scripts themselves.

Playtesting processes were used for debugging purposes and for improvements to the game. Group walkthroughs of the game ensured our creative visions were all realized. Since all of us played the levels, we adjusted how we, as players, would enjoy playing the game. This resulted in us to adjust the player speed, jump force, and coyote time to finetune the movement. This resulted in us altering juice effects, platforms, and thematic content. To gauge whether our instructions were clear enough, we referenced external peers to play the game. This feedback was used to alter the UI, instructions, and basic game mechanics.

Spy Quest Demo: <https://youtu.be/vWnyEDliqN8>

Reflection

This final project was a great way to end the semester. It reinforced the concepts we learned from the past prototypes and lectures. We enjoyed combining all topics (AI, juice effects, level design, etc.). This project also enforced collaboration in a cumulative way to end the course, which was really fun. Our group learned how to collaborate on Unity through Plastic SCM which was a little difficult to figure out at first, but at the end it provided an excellent way to work together. It was also really fun to combine both our creative and technical skills!

Resources:

Tutorials used and code adapted:

- Add sounds for different effects:
 - <https://www.youtube.com/watch?v=UVI6bwldPio>
- Animation: Code modified from there for screen-shake:
 - <https://www.youtube.com/watch?v=N24MhfeoUpE>
- Screen Zoom Feature/ Scope Zoom effect (Code modified):
 - <https://www.youtube.com/watch?v=9g2 qJv AnQI>

Sounds:

1. Pixabay <https://pixabay.com/sound-effects/>
2. Mix Kit: <https://mixkit.co/free-sound-effects/>