Qualitative Testing – Dungeon Escape

Functionality Checklist

- ✓ Player Movement: Player can move in any direction using w-a-s-d as expected
- ✓ Border Colliders: Borders are static and block player from leaving level boundaries as expected
- ✓ Levels and Transitions: Loading zones all redirect to the intended scenes, and it is not possible to skip to other scenes out of order unless they are played directly from the unity app. All 3 planned levels have been implemented
- ✓ Spike Instantiation: Spike prefabs continually generate in a grid pattern controlled by scene 1 GameState as expected. Spikes are destroyed when colliding with borders to make room for new ones
- ✓ Player Spike Collision: Player is always translated to the beginning of the stage when colliding with a spike
- ✓ Coin Collection: Coins are triggered by the player only, and increment the score when touched
- ✓ Ongoing Score: GameState is not destroyed between scenes and score transfers between each level until the end. The score does refresh at the beginning of the game, even in the same run
- ✓ Moving Objects: The objects in the stage 2 puzzle are RigidBodies that are moved by the player when collided
- ✓ Puzzle Triggers: Triggers and objects are tagged with matching keywords, and a trigger only returns true when the matching GameObjects interact, as tested in the inspector
- ✓ Puzzle Logic: When all matching triggers return true concurrently, only then will the overall puzzle be solved. No other combination will work
- ✓ Blockade Destruction: Blockade collider object is destroyed only when puzzle is solved/objective is met, and cannot be removed otherwise
- ✓ Projectile Shooting: Projectile Prefab is produced in front of player when spacebar is pressed, and force applied to the GameObject moves it across screen
- ✓ Projectile Collision: Projectile collision with enemy reduces its health, colliding with anything else will destroy it
- ✓ Enemy Defeat: Enemy takes a certain number of hits, and will be destroyed after its health reaches 0
- ✓ Menus and GUI: Score and Health GUI elements update accordingly to the live values, as tested by debug statements. All buttons navigate to the correct menu scenes. Volume GUIs also affect in-game volume settings live

Usability Testing

In the puzzle, users may find their objects getting stuck in the corners of the colliders, and the puzzle becomes unsolvable. A reset button to put them back in starting position is added to prevent ‘soft-locking’ the game.

One issue with user understanding is that the colour puzzle has no visual indication of an individual match. With more time, sprite changes could have been added to make the puzzle easier to understand, especially for beginner users.

When playing the game, users maybe be disturbed by the volume of SFX or music. A menu with sliders is added to allow the user to customise their experience.

When finishing the game, the player wants to try the game again, or check the high score without cancelling and running again. To fix this, a ‘back to main menu’ button has been added to contribute to replay-ability and create a gameplay loop.