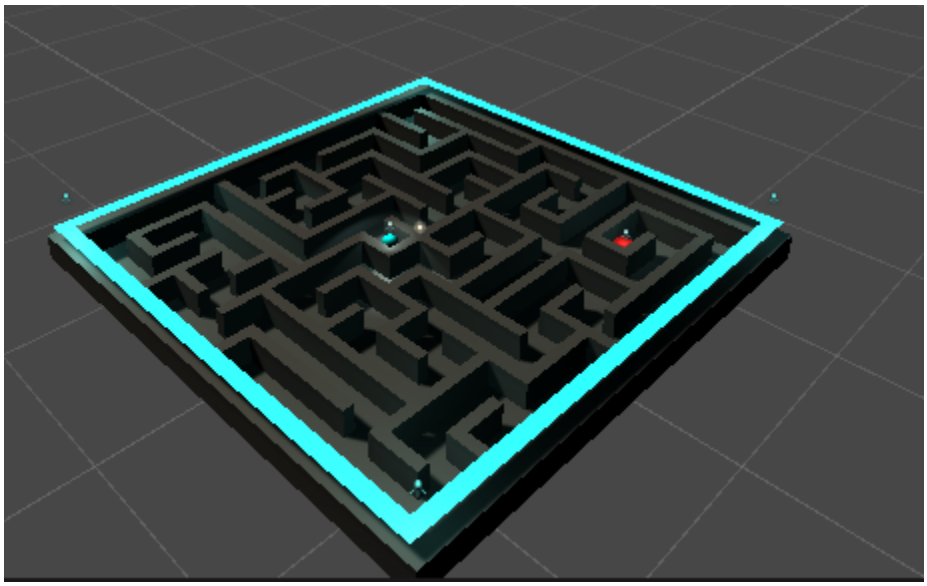
PROJECT PROPOSAL #5

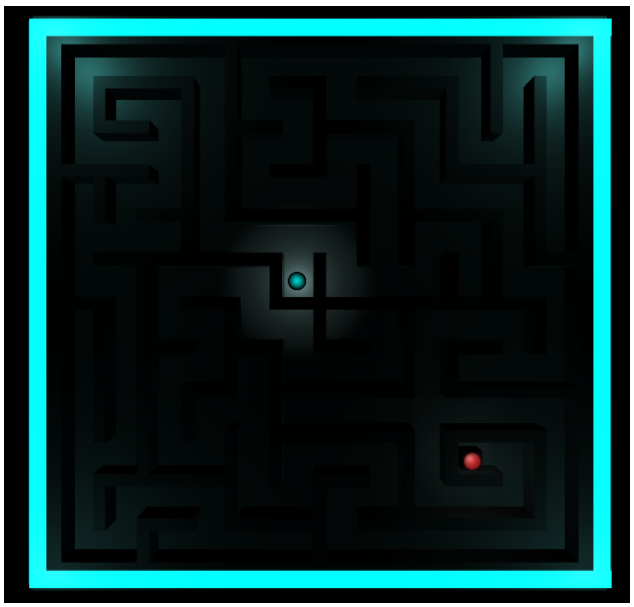
Austin and Sathwik

Feb 7th, 2022

Completed items

- Four mazes have been generated
- The basic lighting for the maze has been fully completed
    - This lighting may change later in the project to improve the aesthetic design of the game





- An arrow key proof of concept has been created, along with an actual setup related to how interaction with hand gestures will be handled.
- Tutorial levels have been created to teach how to move the player around

- Title screen has been created; we are working to also add title screen control with the kinect to make it fully immersive to the Kinect controls
- All three levels have been fully implemented with functioning ends of the levels
- An end of game screen has been created for when the final maze is completed
- The sensitivity of the movement has increased to prevent tiring from a large amount of movement.
- Using the two gameObjects which will represent the two hands of the player, all interactions which will rotate the board have been completed by Austin. A code snippet has been included below.
    - The code originally used collision detection with zones, but due to this causing infinite loops and crashing unity the code has just been set to track the location of the hands and the bounds of each zone manually (hard coded).
- A timer system is implemented
- Intro and exit scenesare created and implemented
- Fixed the unity freezing bug from the introduction up movement to introduction down movement scenes

```
void FixedUpdate(){
        if(leftHand.transform.position.z >= 2.8f &&
rightHand.transform.position.z >= 2.8f){ //both up
            float turnDegree = 5f * ((leftHand.transform.position.z +
rightHand.transform.position.z) / 2f);
            EntireMaze.transform.rotation = Quaternion.Euler(turnDegree-108,
0, 180);
            //Debug.Log("Tilt Forward");
        }
        else if(leftHand.transform.position.z <= -8.4f &&
rightHand.transform.position.z <= -8.4f){ //both down
            float turnDegree = (5f * (1f - ((leftHand.transform.position.z +
rightHand.transform.position.z) / 2f)));
            EntireMaze.transform.rotation = Quaternion.Euler(320-turnDegree,
0, 180);
            //Debug.Log("Tilt Backward");
        }
        else if(((leftHand.transform.position.z > -8.4f &&
leftHand.transform.position.z < 2.8f) && leftHand.transform.position.y <= 19f)
&& (rightHand.transform.position.y > .9f && (rightHand.transform.position.z >
-8.4f && rightHand.transform.position.z < 2.8f))){
            float turnDegree = 180f +
Mathf.Rad2Deg*Mathf.Atan((leftHand.transform.position.x -
rightHand.transform.position.x) / (leftHand.transform.position.y -
rightHand.transform.position.y));
            EntireMaze.transform.rotation = Quaternion.Euler(turnDegree, 90,
90);
            //Debug.Log("Tilt Left");
        }
        else if(((rightHand.transform.position.z > -8.4f &&
rightHand.transform.position.z < 2.8f) && rightHand.transform.position.y <=
19f) && (leftHand.transform.position.y > .9f && (leftHand.transform.position.z
> -8.4f && leftHand.transform.position.z < 2.8f))){
            float turnDegree =
```

```
Mathf.Rad2Deg*Mathf.Atan((leftHand.transform.position.x -
rightHand.transform.position.x) / (leftHand.transform.position.y -
rightHand.transform.position.y));
            EntireMaze.transform.rotation = Quaternion.Euler(turnDegree, 90,
90);
            //Debug.Log("Tilt Right");
        }
        else{
            EntireMaze.transform.rotation = Quaternion.Euler(-90, 90, 90);
            //Debug.Log("DeadZone - set rotation to zero");
        }
        EntireMaze.transform.position = new Vector3(0, 1, 0); //locks around
single point of rotation
    }
```

Items left to be completed

- We need to try to use a multi-camera system to see if the camera and board angles are easier to understand from an alternative view
    - After trying this, a top-down view is by far the best for creating a game such as this.
- We need to increase the camera's Y-level in order to fully fit the maze in the game.
- We need to implement the danger zones (environmental damage)

Challenges we have encountered

We are trying to find different mechanics and different ways to make the controls more fun for the player.