# PROJECT PROPOSAL #7

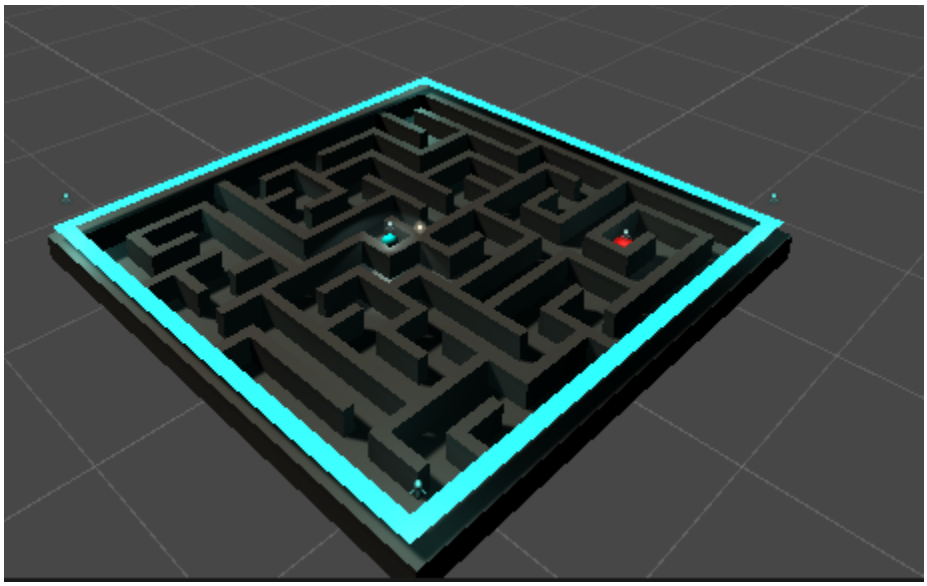
Austin and Sathwik

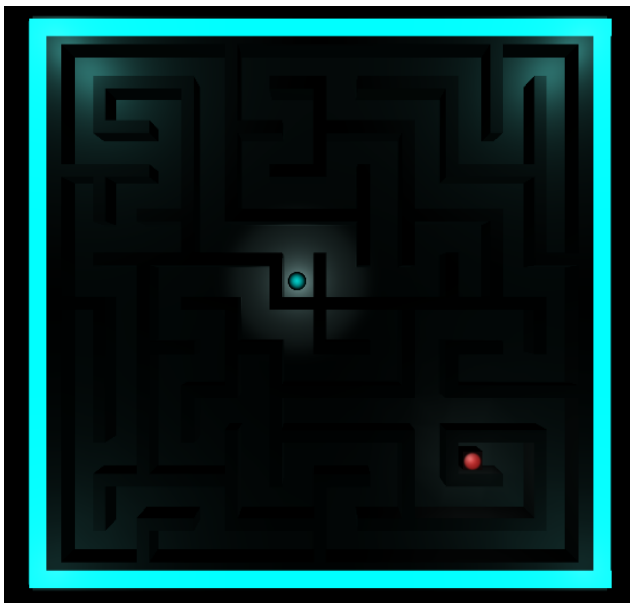
February 14th, 2022

---

**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).

```
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
    }
```

- A timer system has been put in place, forcing the player to move back a level whenever they are unable to complete a level.
- Intro and exit scenes have been fully created, along with a working button system using the Kinect when trying to select an option.
- The angle of the camera has been changed in order to give the player a perspective of looking down upon the actual board instead of having an isometric top-down view of the board.
- We added alternative controls within options, allowing the player to choose which way they wish to move their hands to change the up/down tilting method.
- We fixed the unity freezing bug from introduction up movement to introduction down movement scenes
    - We solved this problem by embedding the entire project into one scene and used GameObject.active to make switch between “scenes”
- We added background music to the game as well to give a techno futuristic vibe to the game.

**Items left to be completed**

- We have to implement individual sounds such as the rolling, hit, danger zone, and timer count down sound effects.

**Challenges we have encountered**

We have a small bug where the game object is not recognizing that it is not in a danger zone outside when switching scenes. This is our only bug right now and we are hoping to solve it by tomorrow and implement an out standing features of the game and polish up the game.