Tank Escape

Ryley Goodine, Zheyuan Xu

# Introduction

The goal of the game is for the player to move the blue cube tank to the end of the maze without touching the enemy cube tank.

Challenges faced with implementing the game were in the creation of the maze, how the player will be colliding with the maze walls, or the enemy cube tank and in initializing projectiles.

~~Challenges where faced with deciding on the size of the maze and its complexity, as the objects defining each wall had their own texture~~

~~Movement of the player is done through a standard browser game mapped to the WASD key arrangement. Rotation of the first-person perspective camera is given to the Q and E key and rotates left and right respectively. Changing the camera is done through the F key, which cycles through a top-down view and a first-person perspective view.~~

# Methods

## Playing Field / Terrain

* Summary of functionality
  + Plane object for floor and cubes for walls.
  + All made using Zacks game engine, with texture and colour added
* Link to theory
* Implementation details

## Main Player

* Summary of functionality
  + The player object is a basic cube shape with colour calculated using the Bling-Phong model and diffuse texturing of the alien.jpg image.
  + The player is moved using the standard browser WASD key configuration.

this.player.translate(vec3.fromValues(+0.x, +0.0, +0.0)); // move left

this.player.translate(vec3.fromValues(-0.x, +0.0, +0.0)); // move right

this.player.translate(vec3.fromValues(+0.0, +0.0, +0.z)); // move forward

this.player.translate(vec3.fromValues(+0.0, +0.0, -0.z)); // move back

* Link to theory:
  + Movement of the player object on the scene is done through constant translation of the player objects current location. The movement in the x-axis is considered flipped because
* Implementation details
  + This code was implemented using resources from the beginning labs and assignment in WebGL focusing on model movement.

## Interacting Objects

* Walls
  + Summary of functionality
  + Link to theory
  + Implementation details
* Finish line
  + Summary of functionality
  + Link to theory
  + Implementation details

## Non-Player Character

* Summary of functionality
  + The enemy red tank, it moves in a fixed loop around its position.
* Link to theory
  + Creation of the model with its texture, and movement around the scene at fixed locations.
* Implementation details
  + Texture image and colour values retrieved from the scene.js file. Its then linked into the fragment and vertice shader where the Blinn-Phong model equation will be used to calculate colour. Texture colour is calculated

## Change of View

* Summary of functionality
  + Two types of views included; top-down and first-person.
* Link to theory
* Implementation details

## Additional Functionality

* Adding sound from projectiles
  + Summary of functionality
  + Link to theory
  + Implementation details
* Adding basic timer
  + Summary of functionality
  + Link to theory
  + Implementation details
* Adding keymapping to html
  + Summary of functionality
  + Link to theory
  + Implementation details
* Music
  + Summary of functionality
  + Link to theory
  + Implementation details
* Changing Lights on a counter
  + Summary of functionality
  + Link to theory
  + Implementation details

# Analysis and discussion