* List of Assets Used
  + Had to disable back face culling to get these to render properly.
  + <https://www.cgtrader.com/free-3d-models/architectural-details/floor/hexagon-tile-3-types>
  + <http://www.turbosquid.com/FullPreview/Index.cfm/ID/540981>
  + <http://www.turbosquid.com/FullPreview/Index.cfm/ID/617081>

Enemy

Bullet

Gun

Terrain

Vehicle

GameData

Game

EnemyManager

TPSCam

Camera

CMOGO

GameObject

Key:

Derived class inherits from Base class

Child class is owned by Parent class

* GameObject – This is the basic class that all objects in the game world inherit from.
  + CMOGO – A type of GameObject that has a CMO 3D model associated with it.
    - Terrain – A basic CMOGO with no movement or tick capabilities. A static object in the world.
    - Vehicle – The player vehicle. It can move on a flat plane with either mouse control or the WASD keys, and owns a gun that is mounted on top like a turret.
    - Enemy – While identical to the player in looks, the logic of the enemy is different. It patrols the terrain randomly, firing its own gun at random intervals. Due to miscalculations, the enemies do not face the correct direction when they move.
    - Gun – A rotatable CMOGO that follows its parent object. It owns a set number of bullets that can be fired either by player control or by AI.
    - Bullet – A CMOGO that travels in a straight line along the direction it is fired, and can destroy objects it collides with.
  + EnemyManager – A parent class for all the enemies in a world. It handles their spawning and acts as the instigator for their tick and draw functions.
  + Camera – A GameObject that provides a viewport for the player, by having a position and a target to look at.
    - TPSCam – A specialised camera that will follow its target object as it moves and turns.
* GameData – This data-only object stores all the useful variables that need to be passed to multiple objects, such as delta time, the keyboard and mouse inputs, and the current camera state.
* Game – The main game loop, which sets up all the objects, reads in the input devices, checks for collisions, and calls the tick and draw functions of all the GameObjects.

## Gameplay Instructions

* Overhead camera
  + Mouse to move
  + Z to shoot
  + WASD to aim gun
* Third-person camera
  + WASD to move
  + Left click to shoot
  + Mouse to aim gun

## Other information

* As stated previously, a mistake in the maths causes the enemies to face in completely the wrong direction when they move from place to place. I have tried multiple different implementations, both using all three dimensions and just two of them, but none of them truly fix the issue. One method, detailed as “MovementType = 2” in GameData, does make the enemies face the correct direction, but it also causes them to stop short for some reason and never reach their first target. MovementType and RotationType should both be set to 1 for the most complete version.
* If it does not build, make sure to set the application as the startup project in the solution browser in visual studio.