1. Tank Combat by Oscar Solorzano
2. 

3. There are run and compile batches available. The server must be running before the client game. Also, the client has 3 arguments in its run.bat that may need to be edited. The first is the server IP, which is displayed on the server console. The second is a port which is hardcoded and shouldn't have to be changed. The third is a string saying either "true" or "false". False renders the game in windowed mode, and true or no argument will run the game in FSEM.
4. It requires a keyboard and mouse.
5. The objective is to fire at all other tanks, every tank killed adds to your score. First to 3 kills wins and the game starts over.
    a. AI opponents can be introduced at any time using the 9 button. However, they are not transported over the server and are for single player purposes only.
    b. Once the server is on, any player can join in at any time.
    c. The number near the crosshair indicates how long until the gun is ready to fire again.
    d. Killing opponents is the only way to replenish your health.
    e. The terrain prevents movement until you back away from it.
6. WASD are used the move and the mouse is used to aim the cannon.
7. Scripting is used for bullet and tank attributes. These include:
    a. Bullet lifetime, movement speed, and scale for size.
    b. Tank HP, movement speed, and rotation speed.

c. These are all intended to change the balance of the game ie, changing tank HP for shorter games, or increasing bullet speed to make it harder to dodge.

8. Requirements:
    a. There are 6 crafted external models I created; the top and bottom of both tanks, the tracks, and the bullet.
    b. The server can handle multiple players. It transports packets describing:
        i. When a player joins and spawns.
        ii. The translation and rotation of other players tanks pieces.
        iii. When a player fires a bullet.
        iv. When a player wins the game.
        v. When a player leaves.

        The server also keeps track of last spawned point used. Also players have a way to select their avatar at start up with is portrayed on other player's client.

    c. Scripting is used for tank and bullet attributes. Specifics are on point 7.
    d. There is a skybox created from Terragen and using the sage skybox class. The terrain is image based which is located in the textures folder.
    e. Event handling is used during collision detection to notify a tank when it has been hit.
    f. 3D sound locations include:
        i. Player and ghost tanks engines and firing sound.
        ii. The sound whenever a tank takes damage or dies.
        iii. There's some faint background sound of a gunfight located in the center of the arena.
        iv. Bullets contain a slight buzzing sound to give the impression of them whizzing by.
    g. There is a HUD enabled by default. It provides information for:
        i. A crosshair to show the center of the screen.
        ii. A reload timer, to let the player know if they can fire their gun.
        iii. An HP bar to visualize the tanks health.
        iv. A string to show the player their score.
    h. The tank objects are both sage groups and take advantage of a hierarchical scene graph. Each tank consists of two objs, for head and body parts, and two ogrexml objects for the tracks.
    i. The tracks are animated and respond to WASD buttons to give the appearance of movement through tracks.
    j. NPCs are implemented using the sage behavior tree. Located in the NPC package is the controller and classes for the nodes of the tree.
        i. NPC actions include: wandering, running away, chasing, and attacking.
    k. The sage physics engine is used to provide velocity to the tanks, and also a frictional slowdown for tanks.
9. All the requirements are implemented in some way to my knowledge.
10. Everything implemented was within the scope of the requirements.
11. I did everything.
12. This was tested on PONG and PAC-MAN