Technical Design Document (TDD)

Collision Detection

- They are used where there is the need to detect the collision between two or more objects.
- Collision detection will be needed for the following:
    - The player hits any enemy's assets.
    - Enemy fire hits the player.
    - The player touches a fuel station.
    - The player's fire hits a fuel station.
    - The player's fire hits an enemy.
    - The player's fire hits a bridge.

Physics

- Objects must react to player input and player decision (e.g. firing a cannon).
- Physics will be needed for the speed of the objects.

Implementation

The player's aircraft have the following variables controls: **fuel level** and **missile**. The fuel level will be consumed as the game goes on. If it gets to zero, the plane will crash, and it will be a game over. The aircraft has only two missiles on each level, and there is no other way to get more missiles during the game. At the beginning of each level, these controls are reset.

The Enemy's Choppers move horizontally and shoot horizontally. It will be generated randomly from the game's left or right border.

The enemy's Ships move horizontally inside the river and shoot down vertically. It will be generated randomly from the game's top border.

The enemy's Anti-aircraft artilleries are static on the ground and shoot in any direction. It will be generated randomly from the game's top border.

The enemy's Jets move vertically and shoot vertically. It will be generated randomly from the game's top border.

The enemy's bridge is static and is generated on the ground at the end of each level.

Shootings: it has one variable to control de speed. It terminates when it hits any object on the game or when it hits the borders of the game screen.

Spawners

- One spawner for the player aircraft with its specific prefab.
- One spawner for enemies that moves horizontally that will receive its designated prefabs (e.g. choppers, ships)
- One spawner for enemies that moves vertically that will receive its designated prefabs (e.g. jets)
- One spawner for static enemies that will receive its designated prefabs (e.g. Anti-aircraft artilleries)

Game Layers

- The background scrolls vertically (down), with ground, roads, rivers, and trees.
- The player and the enemy's objects will be on the same layer, so they can interact with each other.

Effects

- When an object is destroyed, there will be an explosion effect.
- When the player's aircraft move left or right, the sprite will change to give the impression the plane is tilting.

Artificial Intelligence

- Enemy's jets' path will use AI. They fly toward the player's aircraft.

Code and Controls

- The CSharp code will control the behaviour of the game regarding the collisions. When a player's missile hits a bridge, the bridge will be destroyed, but nothing happens if a bullet hits the bridge.
- The player's bullet can destroy the enemy's object and fuel stations.
- The player's missile can only destroy bridges.
- The enemy's bullets can only destroy the player's aircraft. It cannot destroy fuel stations.
- Code will also be used to control the game's score and the fuel level on the player's aircraft.