**Game Design Document for:**

# Return Fire Remake

Written by Arthur de Paula Bressan AKA HardDisk

Version # 0.01

Started on Table of Contents

Conteúdo

[Return Fire Remake 1](#_Toc370143275)

[About this document 5](#_Toc370143276)

[Game Overview 6](#_Toc370143277)

[Common Questions 6](#_Toc370143278)

[What is the game? 6](#_Toc370143279)

[Why create this game? 6](#_Toc370143280)

[Where does the game take place? 6](#_Toc370143281)

[What do I control? 6](#_Toc370143282)

[What is the main focus? 6](#_Toc370143283)

[Feature Set 7](#_Toc370143284)

[General Features 7](#_Toc370143285)

[Multiplayer Features 7](#_Toc370143286)

[Editor 7](#_Toc370143287)

[Gameplay 7](#_Toc370143288)

[The Game World 8](#_Toc370143289)

[Overview 8](#_Toc370143290)

[World Feature: Destructible objects 8](#_Toc370143291)

[The Physical World 8](#_Toc370143292)

[Overview 8](#_Toc370143293)

[Vehicles 8](#_Toc370143294)

[ Jeep: 8](#_Toc370143295)

[ Tank: 8](#_Toc370143296)

[ Missile Truck: 11](#_Toc370143297)

[ Helicopter: 13](#_Toc370143298)

[ Drone 15](#_Toc370143299)

[ Submarine 16](#_Toc370143300)

[Special Buildings: 16](#_Toc370143301)

[ The Base: 16](#_Toc370143302)

[ The Flag Tower: 16](#_Toc370143303)

[ The Flag: 16](#_Toc370143304)

[ Ammo station: 16](#_Toc370143305)

[ Fuel stations: 16](#_Toc370143306)

[ Walls: 17](#_Toc370143307)

[ Gates: 17](#_Toc370143308)

[ Turrets: 17](#_Toc370143309)

[ Miscellaneous Buildings and decorations: 17](#_Toc370143310)

[Scale 17](#_Toc370143311)

[Camera and Controls 17](#_Toc370143312)

[Overview 17](#_Toc370143313)

[Classic control scheme proposal for one player 17](#_Toc370143314)

[ Keyboard: 17](#_Toc370143315)

[ Controller: 18](#_Toc370143316)

[Modern control scheme proposal for one player 19](#_Toc370143317)

[ Keyboard and mouse: 19](#_Toc370143318)

[ Controller: 20](#_Toc370143319)

[Classic: Pros and Cons 21](#_Toc370143320)

[Pros: 21](#_Toc370143321)

[Cons 21](#_Toc370143322)

[Modern: Pros and cons 22](#_Toc370143323)

[Pros: 22](#_Toc370143324)

[Cons 22](#_Toc370143325)

[User Interface 23](#_Toc370143326)

[Overview 23](#_Toc370143327)

[Vehicle Selection 23](#_Toc370143328)

[Classic mode 23](#_Toc370143329)

[Modern mode 24](#_Toc370143330)

[Overview 25](#_Toc370143331)

[Sound 25](#_Toc370143332)

[Music 25](#_Toc370143333)

[Single-Player Game 26](#_Toc370143334)

[Overview 26](#_Toc370143335)

[Single Player Game Detail #1 26](#_Toc370143336)

[Single Player Game Detail #2 26](#_Toc370143337)

[Hours of Gameplay 26](#_Toc370143338)

[Victory Conditions 26](#_Toc370143339)

[Overview 27](#_Toc370143340)

[Max Players 27](#_Toc370143341)

[Servers 27](#_Toc370143342)

[Customization 27](#_Toc370143343)

[World Editing 28](#_Toc370143344)

[Overview 28](#_Toc370143345)

[World Editing Detail #1 28](#_Toc370143346)

[World Editing Detail #2 28](#_Toc370143347)

[Extra Miscellaneous Stuff 29](#_Toc370143348)

[Overview 29](#_Toc370143349)

[Junk I am working on… 29](#_Toc370143350)

# About this document

This document is being written with a little of a headway into the creation of the game itself. During the course of the project the implementation will certainly change, but this document will try to congeal the core of the project as much as it can. Ideas come and go as well, and this document will try to keep up with them.

This document won’t contain any of the implementation details, and at most it will give due consideration to the Unity game engine, which is what is being used to build this game. If there’s a need to include details of technology used, it will be done in an annex.

This will also not contain any business considerations mainly because this project is a portfolio piece. If there’s a need or want or opportunity to do so, it will be done in an annex as well.

Big thanks for the people that made the artwork that appears in this document. All of the artwork are © their respective owners.

Special thanks for Chris Taylor (from Total Annihilation and Dungeon Siege fame) for creating the template for this document. The original template may be found searching for “ctaylordesigntemplate.doc”.

# Game Overview

## Common Questions

### What is the game?

Return Fire at its core was a class-base CTF game where the player or players controlled vehicle representation of those classes. This remake aims to keep this core, while porting the game to modern times.

### Why create this game?

Because I spent many hours playing this game way back when, and it is a rather simple game that hasn’t been ported to modern platforms yet. Basically, the reason to create this game is a combination of its relative simplicity of the systems involved with the fact that I want to complete a project that I want to create.

### Where does the game take place?

The setting of the original Return Fire was a generic contemporary jungle-y collection of islands. This one aims to have a more far future look, but it doesn’t need to be tied down to a specific location setting.

### What do I control?

The player can control the various classes, represented as vehicles, and switch between them as the situations demands.

### What is the main focus?

The main focus in the Multiplayer section is for players to be pitted against each other in arena-style battles to capture the opposing team’s flag, and bring the flag to their base thus either scoring points or as the ultimate goal.

In the singleplayer section, the player is pitted against the clock and the enemy fortification itself, and the player will try to complete levels in the shortest amount of time possible. Online leaderboard support is a consideration in this mode.

# Feature Set

## General Features

* Quick matches
* Balanced classes
* Easy to pick up and play
* Generate new levels with each new game, or select from a list of a lot of premade levels
* Record single-player and multi-player matches

## Multiplayer Features

* Up to 4 players in each team, choosing from either Red, Blue, Green or Yellow
* Local splitscreen play for up to 4 players
* Online leaderboard for single-player maps
* Leaderboard can also feature your own created maps

## Editor

* Comes with map editor
* Download a map from the Internet and edit it.
* Upload a map to the Internet and see it featured on the Leaderboards
* Editor is super easy to use

## Gameplay

* Control the game with a controller, in addition to KB+M
* Choose between “Classical” and “Modern” controls
* Choose between types of level generation, choose from premade levels or create your own
* Almost everything in the level is destructible, even your own base!
* Fortresses featuring turrets, walls, gates, random buildings, supply depots or fuel tanks
* Ammo and Fuel are not infinite, making the game more tactical
* Destroy all of your opponents’ vehicles or capture their flag to win the match
* Score that keeps track of the destruction you’ve caused in the level. Your opponent won the game? WELL, YOUR BASE IS A PILE OF RUBBLE, HOW DO YOU LIKE THAT?

# The Game World

## Overview

The game will take place in matches set in arena-style maps resembling tropical islands, with fortresses built to resemble military bases, lined with walls and protected by turrets. Inside those walls, is the flag towers that, in of one of those, contain the flag of the opposing team. The player that captures the flag will be the victor of the match.

## World Feature: Destructible objects

Everything but the terrain and the vehicle pads are destructible! Surgically remove wall sections from your foe’s base or just level everything. The game keeps score of how much you’ve destroyed, including your opponent’s vehicles.

## The Physical World

### Overview

The world will consist of the many features that appear in the game map. Some of them have a distinct role, like the turrets and the flag tower, while some of them are just for decoration, like trees and other types of buildings.

The following sections describe the key components of the game world. Note that any stats (things like health points, damage dealt, etc) are just suggestions

### Vehicles

The names are just what the class of the vehicle maps to the original game; they don’t have to look like those from the original.

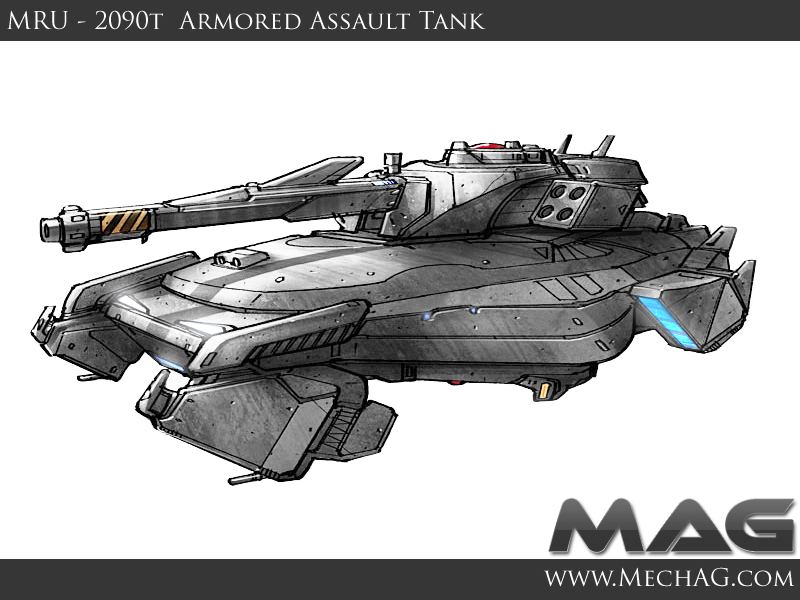
### Jeep:

Fast and fragile vehicle that attacks with a weak cannon or burst machinegun; doesn’t have anti-air. Can switch back and forth between regular mode and a mode that allows it to drive on water, but it is very slow in this state. The only vehicle able to carry the flag. Dies in one hit from anything.

### Tank:

Medium in all of the stats, and can rotate the turret independently from the body.



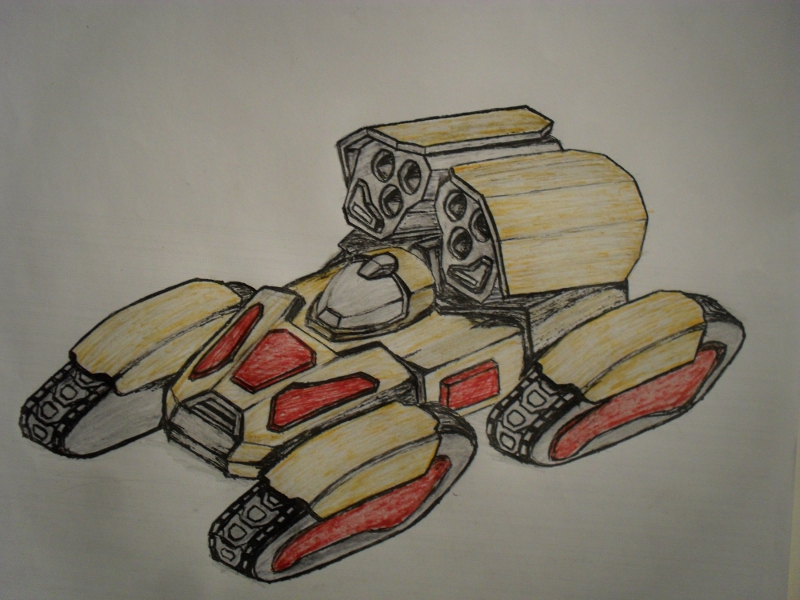


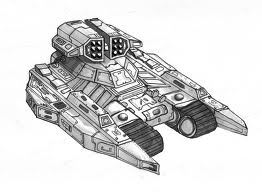




### Missile Truck:

Slower and better armored than the tank, but only shoots to the front of it. Higher range than the tank. Can lay mines.



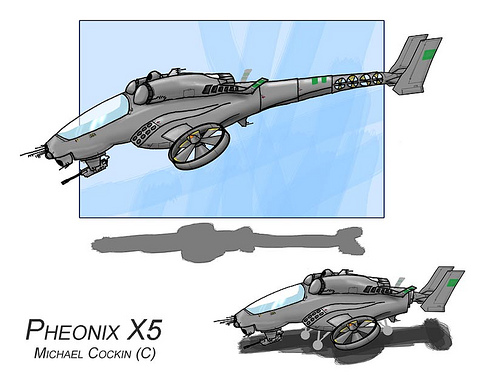


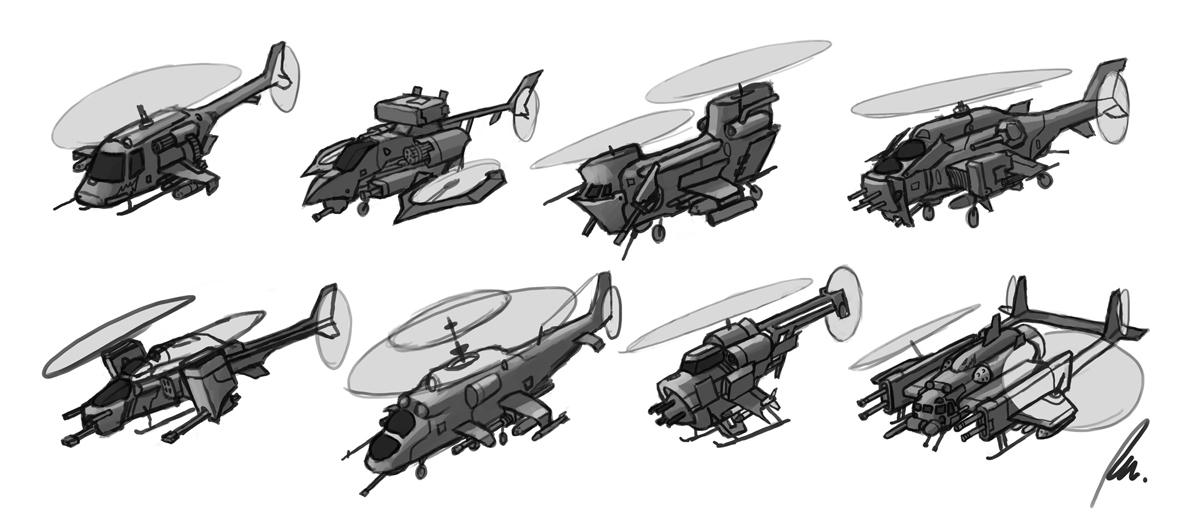


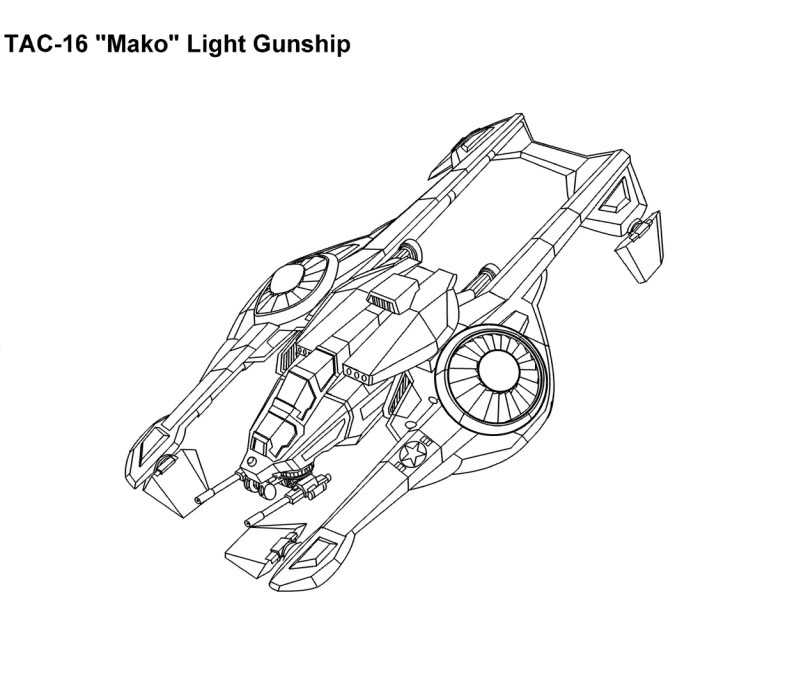
### Helicopter:

Very fast, but light and fragile. Has missiles and cannons. Because of its lightness, it is knocked around by hits and briefly loses control for a time. Consumes fuel even while not moving, contrary to the others. Has to unfold, rev up and gain altitude before being actually in the player control.



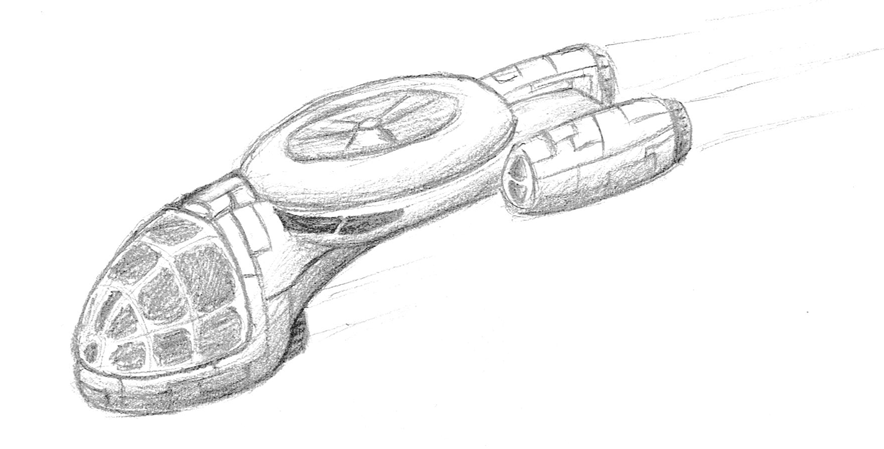






### Air Drone

NPC. Appears only in the Single-Player portion, in later levels. This drone will harass the player if the player keeps still for a long time. Armed with burst machine guns.





### Submarine

NPC. Appears in the deep sea if the players start leave the bounds of the map. Shoots a seeker missile that tracks the closest player regardless of team.

### Seeker missile

NPC. Shot from the submarine, and initially tracks whoever is trying to escape the island.

### Special Buildings:

### The Base:

This is the place where the player can change vehicles, and the place where the flag must be brought to score.

### The Flag Tower:

These buildings hold the flag the player needs to score. There may be multiple of those for each team. Even in these cases, only one flag per team may exist in the world.

### The Flag:

The McGuffin the players must capture in order to win the game. Can be anything appropriate for the world (say, a “power source” or “intelligence” or whatever).

### Ammo station:

These are buildings scattered throughout the bases that rearm vehicles stationed on it. Note that in the original game, they didn’t regard team while rearming, and didn’t rearm the helicopter.

### Fuel stations:

Refuels vehicles stationed on it. Same considerations as the ammo stations.

### Walls:

Keeps things inside, inside, and things outside, outside. Doesn’t work against flying things. Or things that go around them. They also don’t care about your feelings.

### Gates:

Opens up for vehicles from the same team, otherwise the same as walls. Especially the part about your feelings.

### Turrets:

Shoots at vehicles from other teams. They are used in conjunction with the walls to keep things out. The main enemy of the single-player portion. They appear where walls form 90º sections with each other, where the wall ends (such as in the side of gates) and they can appear by themselves.

### Miscellaneous Buildings and decorations:

Basically decorations like plazas, gardens, trees and assorted military-ish buildings. Some of them may generate soldiers that lob grenades at the attacker while running towards the sea/another building. Tents should be squishable.

### Misc. Stuff:

There are also things on the map that aren’t necessarily buildings:

### Trees:

They can be used as cover for the players, but they catch fire in one shot.

### Rocky terrain:

Rough terrain that the threaded vehicles can just pass over without any problem, but the wheeled vehicles slow down considerably

### Shallow water

Water (not necessarily) close to the shore that doesn’t prevent vehicles from traversing it, but they are slowed down considerably as well. The jeep can change modes to traverse faster, but still slower than going on land.

### Deep water

Water (not necessarily) further away from the shore that’s deep enough to fully submerge any vehicle. Any vehicle that isn’t the Jeep in the water mode is considered lost. If a player with a helicopter tries to fly away from the island, a sub appears and shoots a seeker missile.

### Paved roads

Makes wheeled vehicles go faster.

### Scale

The scale of the buildings and the vehicles should be believable, in the sense that, if a building is supposed to be a garage of sorts, then the vehicles should conceivably fit in there. The buildings should have doors that people can fit, and the vehicles must pass the idea of believable interior spaces.

## Camera and Controls

### Overview

There are two possible camera modes: Classic, which is a top-down view reminiscent of the original, and Modern, which is a third-person view behind the vehicle.

All the keys may be rebound at the player`s discretion.

### Classic control scheme proposal for one player

### Keyboard:

### C:\Users\Arthur\Pictures\QWERTY classic onep.png

Movement:

* W: Move forward
* S: Move backwards
* A: Turn wheel/Rotate counterclockwise
* D: Turn wheel/Rotate clockwise

Firing:

* Spacebar: Shoot to ground
* Shift: Shoot to air

R: Contextual button:

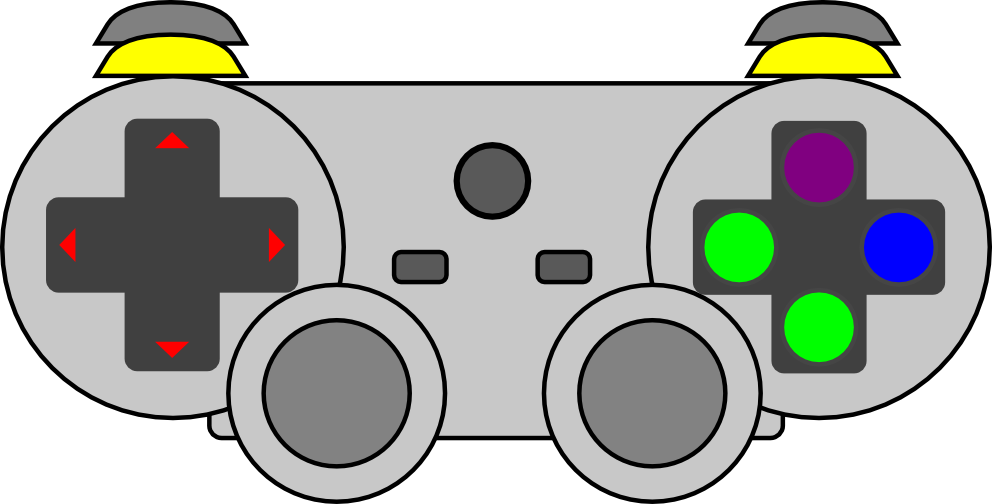
* In base screen (vehicle select): Show map
* Helicopter: Switch ammo type
* Jeep: Switch land/water mode
* Truck: Lay mine

Tank turret controls:

* Q: Turn turret counterclockwise
* E: Turn turret clockwise

X: Recover vehicle: When a player is over the base such that it becomes highlighted, the player can press this button to recover their vehicle and go to the vehicle selection screen

### Controller:



Movement:

* D-pad up: Move forward
* D-pad down: Move backwards
* D-pad left: Turn wheel/Rotate counterclockwise
* D-pad right: Turn wheel/Rotate clockwise

Firing:

* X: Shoot to ground
* □: Shoot to air

O: Contextual button:

* In base screen (vehicle select): Show map
* Helicopter: Switch ammo type
* Jeep: Switch land/water mode
* Truck: Lay mine

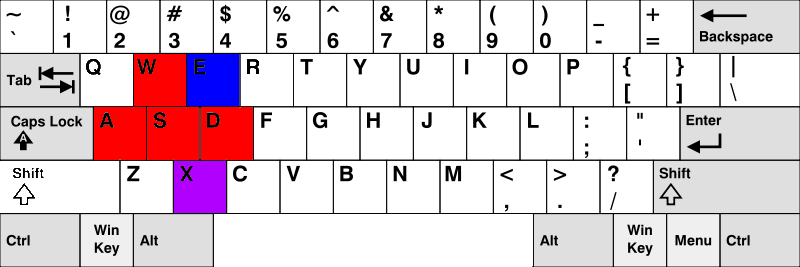
Δ: Recover vehicle: When a player is over the base such that it becomes highlighted, the player can press this button to recover their vehicle and go to the vehicle selection screen

Tank turret controls:

* R1: Turn turret counterclockwise
* L1: Turn turret clockwise

### Modern control scheme proposal for one player

### Keyboard and mouse:



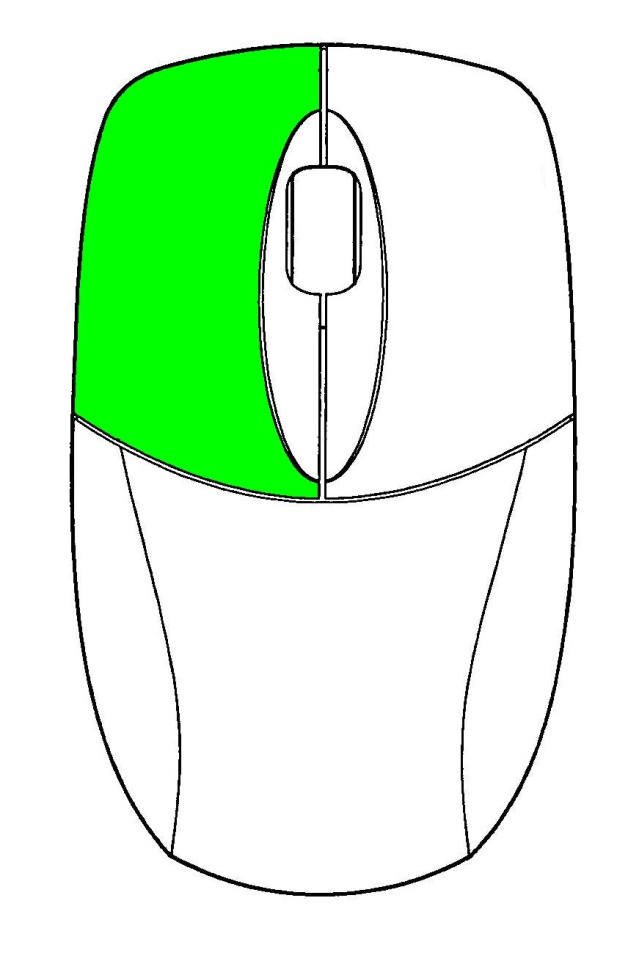
Movement:

* W: Move forward
* S: Move backwards
* A: Turn wheel left/Rotate counterclockwise
* D: Turn wheel right/Rotate clockwise

E: Contextual button:

* In base screen (vehicle select): Show map
* Helicopter: Switch ammo type
* Jeep: Switch land/water mode
* Truck: Lay mine

X: Recover vehicle: When a player is over the base such that it becomes highlighted, the player can press this button to recover their vehicle and go to the vehicle selection screen



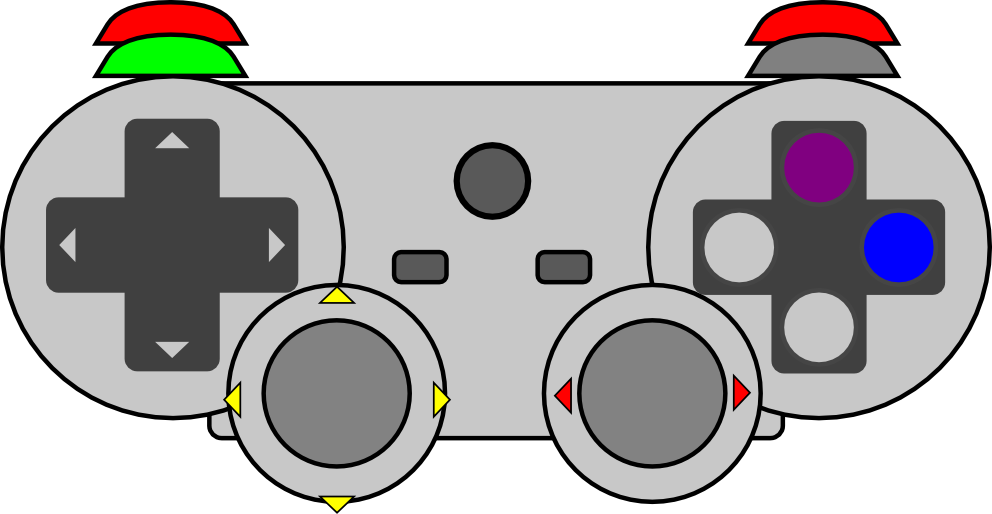
Left mouse button: Fire.

Mouse movement: Controls the camera and the aim cursor.

Some vehicles have wider or narrower arcs. While the camera can be controlled to rotate 180º, the guns’ aims are restricted to that arc.

The aim cursor follows the “cursor” pattern: Push mouse and cursor will go towards the top of the screen. There must be an option for reversing this movement.

### Controller:



Movement:

* R2: Accelerate/Move forward
* L2: Brake/Move backwards
* Right stick X+ : Turn wheel right/Rotate clockwise
* Right stick X-: Turn wheel left/Rotate counterclockwise

Firing:

* R1: Shoot

O: Contextual button:

* In base screen (vehicle select): Show map
* Helicopter: Switch ammo type
* Jeep: Switch land/water mode
* Truck: Lay mine

Δ: Recover vehicle: When a player is over the base such that it becomes highlighted, the player can press this button to recover their vehicle and go to the vehicle selection screen

Camera controls: Left stick controls the camera and the aim cursor

Some vehicles have wider or narrower arcs. While the camera can be controlled to rotate 180º, the guns’ aims are restricted to that arc.

The aim cursor follows the “cursor” pattern: Push mouse and cursor will go towards the top of the screen. There must be an option for reversing this movement.

### Classic: Pros and Cons

### Pros:

* Reminiscent of the original game.
* Easier to implement. Both in terms of the controls and the camera itself
* Easier to work on mobile screens: No need to implement mouselook
* Enables support for splitscreen multiplayer on the same keyboard

### Cons:

* Field of view is limited
* Hard to tell the place the helicopter is going to hit
* Newer desktop players aren’t used to this scheme.

### Modern: Pros and cons

### Pros:

* Doesn’t need a special key to fire anti-air

### Cons:

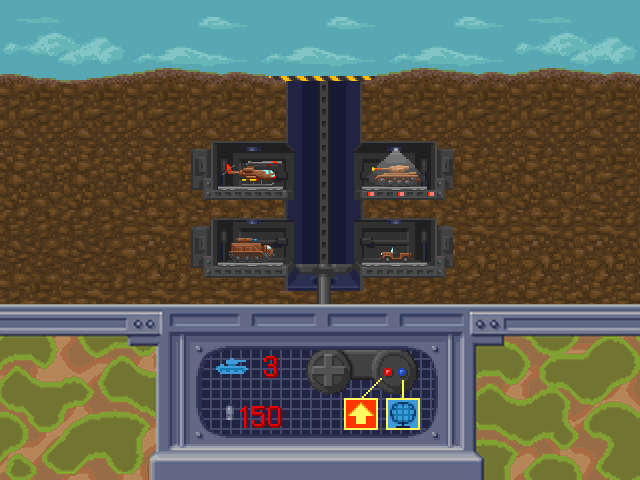
* Must implement firing arcs for the vehicles
* Splitscreen users must have at least one other gamepad

# User Interface and screens

## Vehicle Selection Screen

The vehicle selection screen shows a cutaway of the base, showing the four slots where the vehicles are parked. A lamp should be turned on over the selected vehicle, and any firing key should select the highlighted vehicle. Mouse control should also be enabled for the player`s convenience.

The number of the remaining vehicles of that type should also be shown. If there`s no more vehicles of that type left, then that vehicle must not show up.



Pressing the contextual button/key should toggle the map screen.

## Map Screen

## Classic mode

In this mode, the camera is centered on the player vehicle with some tilt to show perspective. How fast the player is moving corresponds to how far the camera is from the player vehicle.



## Modern mode

In this mode the camera is behind the player vehicle, with the cursor centered on the screen. The vehicle`s turrets will try to angle to match the player cursor, up to their maximum arcs.

-

Musical Scores and Sound Effects

## Sound Effects

The audio doesn’t need to be ultra-realistic, but they also can’t be cartoony. “Stereotypical” sounds are fine.

## Music

From Wikipedia:

“Also notable was the game soundtrack, composed exclusively by classical pieces such as [Richard Wagner](http://en.wikipedia.org/wiki/Richard_Wagner)'s [*Ride of the Valkyries*](http://en.wikipedia.org/wiki/Ride_of_the_Valkyries) when using the helicopter (in homage to [*Apocalypse Now*](http://en.wikipedia.org/wiki/Apocalypse_Now)), [Flight of the Bumblebee](http://en.wikipedia.org/wiki/Flight_of_the_Bumblebee) when driving the jeep, Holst's Mars from [The Planets](http://en.wikipedia.org/wiki/The_Planets) when driving the tank or the [*William Tell Overture*](http://en.wikipedia.org/wiki/William_Tell_Overture) by [Gioacchino Rossini](http://en.wikipedia.org/wiki/Gioacchino_Rossini) when riding with the flag. When driving in the Armored Support Vehicle, [Edvard Grieg](http://en.wikipedia.org/wiki/Edvard_Grieg)'s [*In the Hall of the Mountain King*](http://en.wikipedia.org/wiki/In_the_Hall_of_the_Mountain_King) plays.”

To expand:

* Tank: Holst`s Mars (The Planets)
* Jeep: Flight of the Bumblebee
* Truck: In the Hall of the Mountain King
* Helicopter: Ride of the Valkyries
* Found flag: That Hallelujah music
* Jeep with flag: Willian Tell Overture

# Single-Player Game

## Overview

The campaign is a series of progressively harder levels, but without a story (as in, characters, dialogues, cutscenes, cinematics and whatever else) linking them. This makes the game much easier, faster and cheaper to make, and retains the arena feeling for the multiplayer portion.

This doesn`t mean that the levels can`t be linked by references (as in, nods to each other), or have levels in the vein of “X but harder”. They must not veer too much from the same theme, though.

## Capture the flag

It`s the main mode of play. The victor is the one that brings the McGuffin hidden in the enemy base to their own base.

The main enemy throughout all of the levels are the turrets perched on the walls of the enemy`s fortress, and the labyrinth consisted of those walls. Players must manage their vehicle`s lives and their resources.

In the later levels, more enemy types such as flying and land drones show up, and AI-controlled vehicles are also a threat to the player.

Time is recorded, and that can be uploaded to an online leaderboard, where the players can share their times with their friends.

## Destruction score

Most of the elements of the game world are destructible, and those have a point score tied to them. The player that destroys an object from an opposing team gain that much points. Conversely, if a player destroys objects from the same team, those points are subtracted from that player`s total.

This score doesn`t decide who`s the winner, but it may grow into a mode in itself and might be another way the players have fun.

## Hours of Gameplay

For the base game in average, each level must last between 0:15 minutes (the easiest ones, with a newcomer player) to 10:00 minutes (the hardest ones, with a reasonably skilled player)

The total time can be close to infinite, because of the level-sharing feature built in the game.

Multiplayer Game

## Overview

The multiplayer matches resemble a lot of the single-player game. A single-player must be very familiar on how it should work: Two or more bases, and at least one player for every active team on the game.

The CTF part is the focus of the multiplayer game but the Destruction Score also comes back, to make another statistic in the matches.

## Max Players

At least two teams in the map, or at most four teams. Teams in the map don`t necessarily need to have a player.

From one player (a practice mode) to two players per team.

## Servers

The game should be built with an authoritative server, and this server is automatically created by the host of the match in the moment a lobby is created. This should also allow the distribution of a separate executable with only the server.

Note that for the singleplayer version, this server should be created as well, but at a local address to minimize architecture differences.

## Customization

* The maximum time, which if hit should be considered a draw. Also a checkbox to not mark time.
* Whether to tally destruction score
* Number of lives per vehicle
  + 1 – ∞, with a checkbox for each vehicle for marking infinite lives for that vehicle
  + A global checkbox for marking all vehicles infinite

# World Editing

## Overview

Every game map is internally represented by as a tile-based map, with each cell representing an object in the game.

A map editor is planned to be delivered together with the game, giving players graphical representations for the game elements.

## Basic editing

To create a map:

1. The user opens up the map editor; a box opens with a selection for map size. A list of common sizes should open up, and it also should allow for arbitrary input.
2. When the user confirms the size, the main window appears. It shows the map, initially covered in Deep Water tiles
3. The user can “paint” the terrain using “brushes”, much like users from paint programs.
4. From there, the user can place any buildings and decorations he wishes by first clicking the desired element in a palette and then, the cell the user wishes to place that object. This should be similar to a regular terrain brush, but limited as a 1x1 brush. Some buildings occupy more space than 1x1, and the editor must support those cases.
   1. Bases: Wherever a user places a base for a team, if the base for that team has already been placed, the old one must be removed and a new one is created in the place the user clicked.
5. For multiplayer, the map is considered valid as long as there one base for each team, and at least one flag tower for each team. For singleplayer, the map is considered valid if there is one base for the playing team and at least one flag tower for at least one non-playing team.

## Roads, walls, turrets and gates

**Turrets** can be placed either stand-alone or they can be converted from a straight wall section. They are automatically placed when a wall section meets another perpendicular section.

**Wall** and **Roads** are placed by selecting two points in a line. The editor then fills the between cells with either roads or walls. If a road passes an existing wall section, a gate will be created, and vice versa.

**Roads** must connect to:

* Perpendicular roads
* Ammo and fuel stores
* Flag towers