CS 447/547 Project 2 Design Document

MARSHAL

Solongo Munkhjargal, Hua Zhang, Kuei-Ti Lu

# Game Overview

## How will the game be played?

The game will be a real-time strategy game similar to *Command & Conquer* (see the image below). In this game, there will be two countries. Each country has its own headquarter, tank and defense wall factories, and tanks. Two players battle against each other, and each represents one of the two countries. A player will be able to command the tank to attack another’s entities. The goal is to take down the other player’s headquarter.



*image retrieved from* [*http://en.wikipedia.org/wiki/File:CnC\_Red\_Alert\_iPhone\_Gameplay.png*](http://en.wikipedia.org/wiki/File:CnC_Red_Alert_iPhone_Gameplay.png)

## What types of interactions are possible?

The players use mouse clicks to command a tank to attack or retreat, select an attack target, and perform other controls. The players will have coins, which they can spend on building defense walls and tank factories or upgrading tank factories. The player can choose where to build the defense walls and tank factories using mouse clicks.

For viewing, the players can move the cursor toward near the boundaries of the window to scroll the maps.

## What are the visual entities in the game?

In addition to the map, there are headquarters, tank factories, tanks, and defense walls.

The headquarters are critical; if a player’s headquarter is destroyed, the opponent wins the game. Tank factories produce tanks. Defense walls can be built to protect the headquarters.

The map will be a tiled map that has several layers as varied terrains, which can be obstacles for tanks.

## What will the player do?

There will be two players in each round of the game. The players use tanks, factories, and defense walls to battle each other to win the game. A player must destroy the opponent’s headquarter. The players command tanks to attack the opponent’s headquarter or other entities, including defense walls, factories, and tanks. They can also spend money on building defense walls or factories and upgrading factories to manufacture more powerful tanks. Players earn coins by destroying the opponent’s factories or tanks but not walls.

## What makes this idea interesting, or why do you think this will be fun?

As a multi-player game, friends and families can play the game together. There will, also, be a leaderboard to engage people to compete with one another. In the game, the players can choose where to build their own defense walls and factories. The players also have choice of whether a tank should attack, stand by, or retreat and the target of an attack. This flexibility makes the game interesting because players can try out different strategies.

# Development Strategy

We will start from scratch in general, but there are libraries or tools available for several tasks; for example, we will use the Tiled map editor to create the maps. Except networking, most code should be similar to what one or more of us have done in Project 1.

The hardest part of the project is the networking part. More detail is in the “Technical Showpieces” section.

The first sticking point is building the initial infrastructure. This includes building basic objects for the entities and creating a basic map.

The second sticking point is the user interface and state transitions from a stage to another. This affects both further development and testing.

The following table shows our subgoals and milestones’ plan. The technical showpiece is highlighted (see the “Technical Showpieces” section for more detail).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Milestone** | **Task** | **Main Member(s) for the Task** | **Planned Start Date** | **Planned End Date** | **In Alpha Release (Dec 2)?** |
| 1 | Build base objects | Kuei-Ti | Nov 10 | Nov 16 (temp, should be earlier) | tick.jpg |
| 1 | Create a basic map and collect artwork for maps | Kuei-Ti | Nov 10 | Nov 16 | tick.jpg |
| 1 | Create UI and  game stage transitions | Solongo, Hua, Kuei-Ti | Nov 10 | Nov 23 | tick.jpg |
| 2 | Pathfinding for tanks | Hua | Nov 16 | Nov 28 | tick.jpg |
| 2 | ***Networking*** | Solongo | Nov 16 | Dec 1 | tick.jpg |
| 2 | Scrolling | Solongo, Kuei-Ti | Nov 20 | Nov 21 | tick.jpg |
| 3 | Collect animations | Hua | Nov 26 | Dec 5 |  |
| 3 | Collect sounds | Kuei-Ti | Nov 24 | Dec 15 |  |

# Technical Showpieces

The game features multiplayer real-time networking play.

## Networking

Players will sit on different machines that are connected to the LAN. Since the game is real-time, synchronization is critical. Using client/server architecture and ensuring thread-safety would solve it. Server machine will process entire game logic. Clients will send their inputs to the server and server will respond with current state of the game. Since network latency is negligible, client side prediction algorithm is not necessary. Player should be able to see opponent’s territory by scrolling. Therefore, each player’s current state will be passed to server frequently. In order to utilize bandwidth efficiently, packet size will be minimized.

## Other Technical Features

### State-Based Behavior

The headquarter will have at least two states: active and inactive. This is used to determine whether the game ends and who wins/loses.

The factory will have at least three states: manufacture, deliver, and inactive. They represents making tanks, releasing tanks, and being destroyed respectively.

The tank will have at least five states: target, attack, move, standby, and inactive. In the target state, the tank waits for the user to select a target. If the target selected is an entity, the tank transitions to the attack state, moves toward the target, and attacks the selected target when the target is within the attack range; if the target selected is not an entity but just a tile on the map, the tank transitions to the move state, moves to the destination, and transitions to the standby state. In the standby state, the tank does nothing. The inactive state represents the tank’s being destroyed.

### Collision Detection

Different types of tanks have different attack ranges. Collision detection is used to check whether a target is within the tank’s attack range, which determines whether the tank moves or attacks in the attack state.

### Pathfinding

Layered maps will be used. A layer will contain obstacles through which the tank cannot pass. Also, tanks cannot pass through other entities. Pathfinding will be used so that tanks can move toward attack targets or destination automatically. Tanks can move in eight directions: left, right, up, down, upleft, upright, downleft, and downright. The A\* algorithm will be used for pathfinding.

### Scrolling Maps

The maps will be scrolling maps, both vertically and horizontally.

# High Bar

1. AI: Create AI to allow single-player mode, in which the player plays against the computer.
2. More types of entities: Create more factory or weapon entities.
3. More maps: Design more maps.
4. More control for tanks: Allow bridge building and other functions for tanks to get across obstacles such as river. These functions should cost coins.
5. Chat room: This allows the players to chat with each other. Players cannot view the opposing team’s chat.
6. Team play: Each team member controls different units and independently earns coins.
7. Timed play: Each round has a time limit.
8. More images, music, and sounds: There can be additional images, music, or sounds.

# Low Bar Checklist (On a separate page)

1. Leaderboard: Players are ranked in the leaderboard based on the coins left at the end of the game. The winner receives bonus coins at the end of the game.
2. Scrolling maps: Players can scroll around the map.
3. Headquarters: Each round of the game contain one headquarter for each of the two players.
4. Various types of tank factories: Different types of factories produce different types of tanks. Factory types change when upgraded using coins.
5. Various types of tanks: There should be at least two types of tanks, one of which is more powerful than another in terms of damage caused to targets, health, defense, or range of attack.
6. Health bars: Each entity has its own health bar. Headquarters, tanks and factories can be repaired (before they are fully destroyed) by spending coins.
7. Coin trackers: These track the players’ coins.
8. ***Networking***: Players play against each other through networking.
9. Pathfinding: Given a target, a tank should perform pathfinding to reach the target.
10. Entity state-based behavior: headquarter, factory, and tank behavior is state-based.