Tank Game - Design Document

0. Team Members

Bemen Girgis - 900213066
Ahmed Badr - 900202868

1. Application Overview

The Tank Game is a single-player combat game where the player controls a tank to defeat an enemy house while avoiding obstacles and mines. The game combines real-time movement, projectile physics, and multi-threaded game mechanics.

2. Code Structure

2.1 Package Organization

```
src/main/java/com/tankgame/
                - entities/   # Game objects and entities
                - ui/         # UI components and game panel
                - utils/      # Utility classes and enums
```

2.2 Classes

Entities Package

- `GameObject.java`: Abstract base class for all game objects
  - Provides common properties (position, dimensions)
  - Defines abstract methods for update and render
- `Tank.java`: Player-controlled tank
  - Handles movement and rotation
  - Manages health and missile firing
  - Uses smooth rotation animation
- `House.java`: Enemy building
  - Implements automated missile firing
  - Tracks player position for targeting
  - Manages health system
- `Mine.java`: Hazard object
  - Blinking animation (on/off)
  - Provides instant destruction on contact
- `Mountain.java`: Obstacle object
  - Immovable barrier
- `Missile.java`: Projectile object
  - Handles movement and collision
  - Distinguishes between player and house missiles

UI Package

- `GameWindow.java`: Main application window
  - Manages game layout
  - Handles window sizing and positioning
- `GamePanel.java`: Main game rendering panel
  - Pretty much handles everything
    - Implements game loop
    - Manages entity updates
    - Handles collision detection
    - Processes keyboard input
- `ControlPanel.java`: Game configuration panel
  - Provides difficulty sliders

Utils Package

- `Direction.java`: Enum for movement directions
  - Provides directional vectors
  - Handles rotation calculations

3. Threading Model

3.1 Main Game Thread

- Runs the game loop in `GamePanel`
- Manages fixed time step updates
- Handles entity movement and collision detection

3.2 Event Thread

- Manages UI updates and rendering
- Handles user input events
- Controls slider interactions

3.3 Thread Synchronization

- Used ConcurrentHashMap for key state tracking

4. Assumptions

- Screen resolution minimum of 800x600
- Missiles travel at constant speed
- Missiles cannot go through mountains
- Missiles can only move in eight directions