

# The Best Defense

Game Design Document



November 26, 2014

CIS 587

Eric Smith

# Table of Contents

- Design History ..... 1
- Development Team..... 1
- Section I - Game Overview..... 1
  - Game Concept..... 1
  - Target Audience ..... 1
  - Look and Feel ..... 1
  - Story Abstract ..... 1
  - Development Platform ..... 1
- Section II - Gameplay and Mechanics ..... 2
  - Gameplay ..... 2
    - Game Progression ..... 2
    - Mission/challenge Structure ..... 2
    - Objectives..... 2
    - Play Flow ..... 2
  - Mechanics ..... 2
    - Physics..... 2
    - Movement..... 3
    - Objects ..... 3
    - Actions ..... 3
    - Combat..... 3
  - Screen Flow ..... 4
    - Screen Flow Chart ..... 4
    - Screen Descriptions..... 4
- Section III – Story, Setting and Character ..... 5
  - Story and Narrative ..... 5
    - Back story ..... 5
    - Plot Elements ..... 5
    - Game Progression ..... 5
    - Story Board ..... 5
  - Game World ..... 6
    - General look and feel of world ..... 6
    - Area #1 ..... 6

Characters .....	6
Character #1 .....	6
Section IV – Levels .....	7
Level #1 .....	7
Section V - Interface .....	7
Visual System .....	7
HUD .....	7
Menus .....	8
Camera .....	8
Lighting Models .....	8
Control System – How does the game player control the game? What are the specific commands?... 8	
Sound Effects .....	9
Section VI - Artificial Intelligence .....	9
Opponent AI .....	9
Section VII – Technical .....	9
Target Hardware .....	9
Development hardware and software .....	10
Section VIII – Game Art .....	10
Characters .....	10
Environments .....	13

## Design History

Document Change History		
Date	Author	Description
11/21/2014	Eric Smith	Initial document creation
11/23/2014	Eric Smith	Updated overview
11/25/2014	Eric Smith	Updated gameplay
11/26/2014	Eric Smith	Updated story/levels

## Development Team

Design – Eric Smith

Programming – Eric Smith

Testing – Eric Smith

## Section I - Game Overview

### Game Concept

The best defense is a first person shooter (FPS) in which the player must defend a specific asset while it is undergoing repairs. Waves of enemies assault both the player and the asset. The player wins when the asset is fully repaired or if all enemy forces are defeated. The player loses if the player's health is depleted or the asset is destroyed.

### Target Audience

Casual gamers that enjoy FPS style games with brief gameplay sessions, ages 13 and up.

### Look and Feel

The game features a present day military themed rural environment. Imagery will be typical of combat themed FPS games. Terrain is relatively sparse and non-descript and could represent a number of geographic areas, such as a war torn Midwest United States or Southern California. Individual levels will feature a military themed asset that must be defended against enemy soldiers and equipment.

### Story Abstract

The unnamed protagonist is a common enlisted soldier assigned to the mechanics division of a modern army. The friendly army finds itself being overrun by enemy forces but must hold their position. To achieve this goal certain assets must be kept operational, such as generators, defense systems, and heavy equipment.

### Development Platform

The game is being developed in the pc version of Unity 4.5. Assets are being generated using Blender 2.72b. Testing will be performed on a Windows 7 64 bit pc with Direct X

11 and a Radeon HD6950 2GB graphics card, Intel I3 3.1GHz processor, and 8 GB of RAM.

## Section II - Gameplay and Mechanics

### Gameplay

#### Game Progression

The game progresses as the player successfully defends and repairs specific assets. Early levels require the protection of trivial assets against weak enemy forces. As the player advances the assets being protected become more valuable, for example capable of playing a strategic role in defeating the enemy army, and the attacking enemy forces grow more powerful.

#### Mission/challenge Structure

Each mission will consist of protecting a single asset against enemy attackers. The asset will be assigned and described at the beginning of the level. The player will need to locate the asset then either distract or kill enemies to protect the asset. Repairs to the asset occur automatically at a set rate, but the repair rate can be increased if the player actively engages in repairing the asset. While engaged in repairs the player is unable to attack and will have a limited view of the playfield.

#### Objectives

- Destroy enemy forces
- Defend and repair assets
- Gather and manage ammunition
- Survive enemy attacks

#### Play Flow

The player begins in a quiet area of the level. The player's first objective will be to locate the asset being defended. The player will begin with a small amount of ammunition. Enemy forces attack in waves of increasing number of enemies. The player will need to balance defeating enemies with repairing the asset and collecting ammunition. Ammunition is available through enemy drops and resources located in static positions throughout the level map. The player must survive and protect the asset until either the asset is fully repaired or all enemies are defeated.

### Mechanics

#### Physics

The game will utilize a simple model of typical Newtonian physics including gravity and elastic collisions. The game physics will be modeled after an earth based environment but implemented in a way that enhances the enjoyment of

gameplay. For example gravity might be deliberately weakened to enhance jump height and fall speed.

## Movement

### *General Movement*

Typical FPS movement will be used for player movement. The player will move in 3 dimensions at set speeds. Movement options include run forward, walk backward, strafe left, and strafe right. Each type of movement will have a constant speed.

### *Look Movement*

Typical FPS look actions will be implemented. The player will be able to look in 3 dimensions at variable speeds. Look options include rotate left, rotate right, look up, and look down. Each type of look will have a variable speed based on the movement of an input device by the user, for example mouse movement.

## Objects

### *Picking Up Objects*

Objects will be picked up by simple contact. Upon collision with an object, such as ammunition, the object will be added to the player's inventory.

## Actions

### *Switches and Buttons*

Switches and buttons will be activated by simple contact. Upon collision with a switch or button the action controlled by the switch or button will commence.

### *Picking Up, Carrying and Dropping*

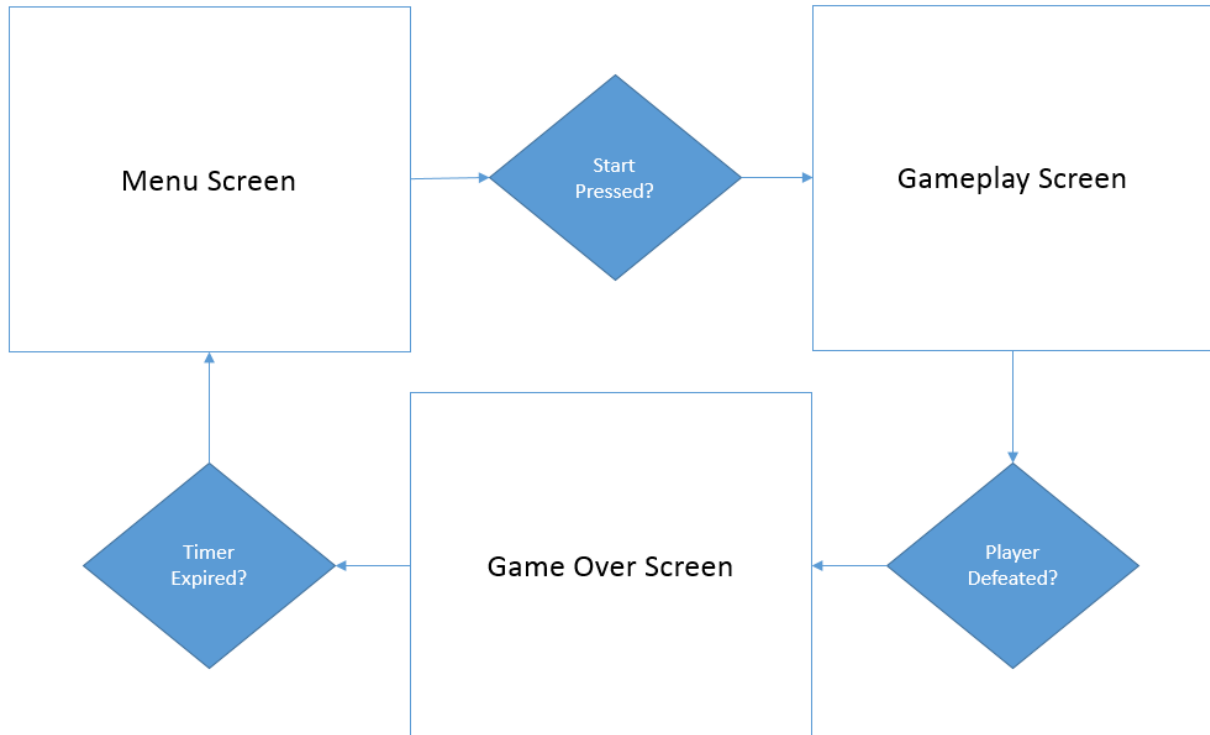
All objects are picked up by simple contact. Collision with an object results in the object being added to player inventory. The player is not able to drop collected items, except through their use.

## Combat

Combat involves aiming and shooting enemies. A basic hit, no-hit system is employed. If a shot contacts any portion of the enemy avatar a hit is registered. Enemies vary in strength with stronger enemies requiring more hits to defeat. Enemy fire that contacts the player also registers as a hit and depletes the player's health. The player's remaining health is displayed in the heads up display HUD. When the player's health is fully depleted the player is defeated. Enemy fire that contacts the asset being defended registers as a hit against the asset and depletes the asset's health. When the asset's health is fully depleted the player is defeated.

## Screen Flow

### Screen Flow Chart



### Screen Descriptions

#### *Menu Screen*

Used to start the game and display basic control instructions.

#### *Gameplay Screen*

Displays the HUD including player health, asset health, and remaining ammunition. Also displays all on screen real time gameplay.

#### *Game Over Screen*

Displays game over text and performance summary for a brief duration of time, for example 5 seconds.

## Section III – Story, Setting and Character

### Story and Narrative

#### Back story

Two nations are at war over scarce resources. The war has been ongoing for many years leaving a number of military assets in a state of disrepair. To defeat the enemy army and end the war, teams of mechanics have been dispatched to repair key assets.

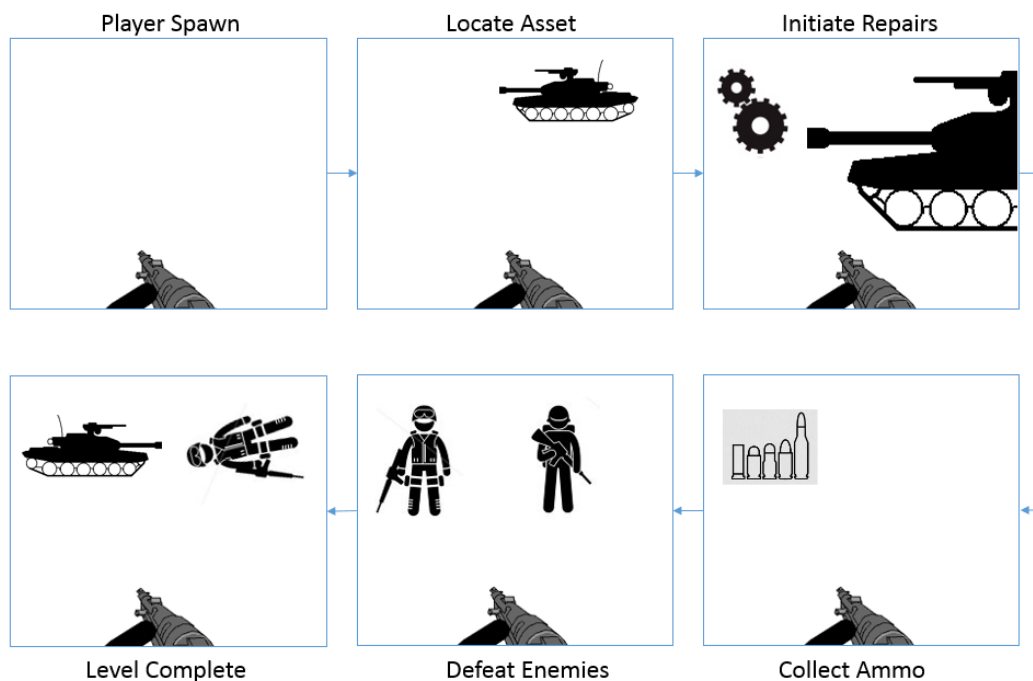
#### Plot Elements

The assets being protected drive the story plot and serve as the main elements describing the conflict between warring factions. Early assets include objects capable of defending a single base or defeating a single enemy unit, such as an electrical generator, machine gun turret, or elevated guard post. Advanced levels include assets which could affect the end of the conflict, such as prototype air vehicles, nuclear weapons, and satellite systems

#### Game Progression

The plot progresses as the player advances through the game levels protecting more strategic assets against stronger opponents. In this way the nature and level of conflict is alluded to, until the final level is defeated and the enemy surrenders.

#### Story Board





## Game World

### General look and feel of world

The game takes place in a rural war torn environment. The landscape is sparse and oppressive. Few trees line the horizon and those that do are leafless and thin. The sky is overcast and does not reveal the exact time of day. In general the world has a gloomy feel and is reminiscent of fall.

### Area #1

#### *General Description*

Outside flat area about 100 meters square. The area has a few trees and the terrain is a sparse grassy and rocky mix.

#### *Physical Characteristics*

The area is enclosed on all sides by terrain features such as steep hills.

#### *Levels that use area*

Mid-tier level used during first playable prototype.

#### *Connections to other areas*

All area connections are generated through successful completion of the level.

## Characters

### Character #1

Player's character, the protagonist.

#### *Back story*

An automobile mechanic until the war began, our hero has since been recruited to help win the war effort.

#### *Personality*

Sets out to complete each mission without complaint, focuses on the job at hand, not much for small talk.

#### *Look*

Represented on screen only by the barrel of a carried automatic assault rifle.

#### *Special Abilities*

Able to increase the repair rate of the asset being defended.

#### *Relevance to game story*

Main character controlled by player. Through this characters efforts the war is won.

## Section IV – Levels

### Level #1

Prototype level – Please see separate level document

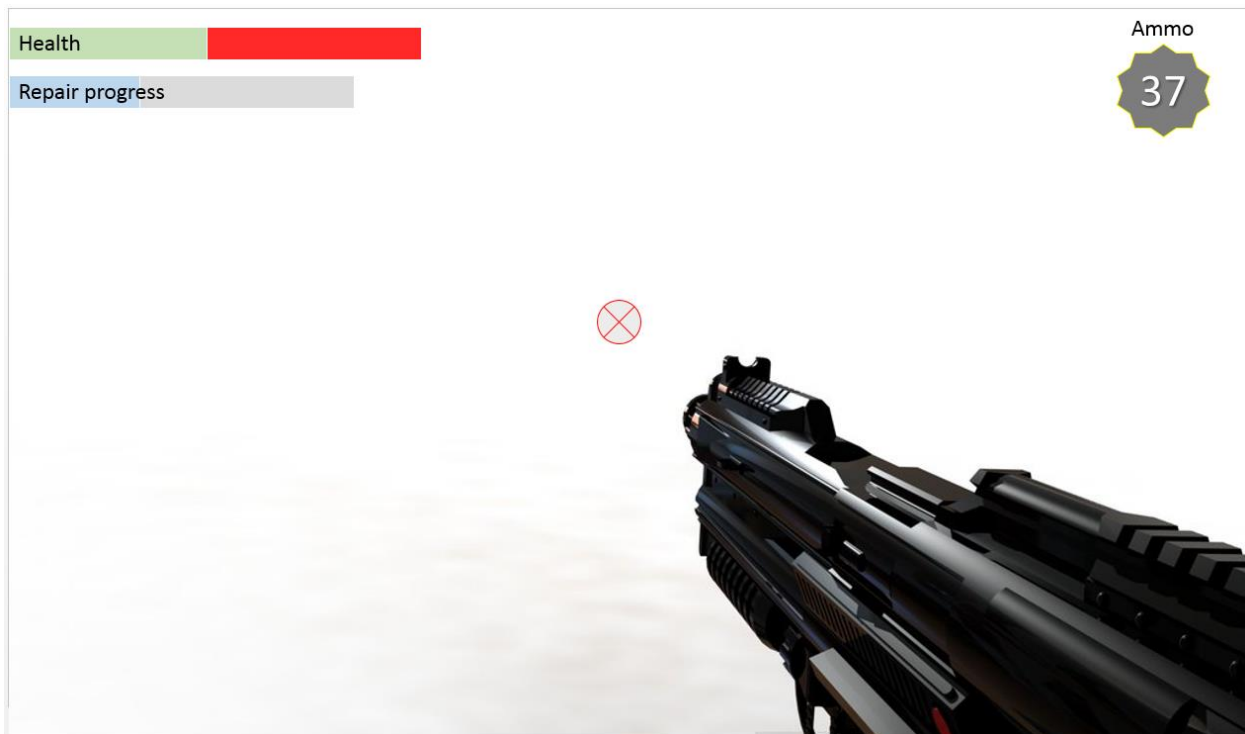
## Section V - Interface

### Visual System

#### HUD

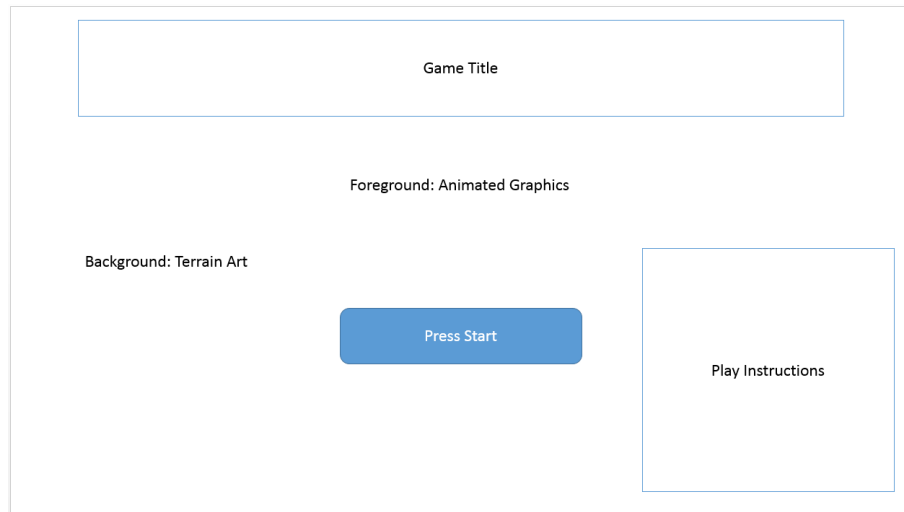
The HUD consists of 3 elements:

- Player health – A bar that decreases as the player takes damage
- Asset Repair Status – A bar that grows until the asset is fully repaired
- Ammunition Quantity – A counter indicating the number of rounds of ammunition the player has remaining.



## Menus

Only 1 menu screen exists in the game, the main menu. It consists of a start game button, game title, and game instructions.



## Camera

First person perspective, free look controlled by mouse interface device.

## Lighting Models

Outdoor lighting daytime lighting is modeled using an area light to represent the sun. Night scenes use spot lighting to represent lamps on building, headlights, flashlights, and other artificial light sources.

## Control System

The game is controlled using a keyboard and mouse.

- Mouse scroll down – Look down
- Mouse scroll up – Look up
- Mouse scroll left – Rotate Left
- Mouse scroll right – Rotate right
- Keyboard W – Forward
- Keyboard A – Strafe left
- Keyboard S – Backward
- Keyboard D – Strafe right
- Keyboard Space – Jump
- Mouse left button – Fire
- Mouse right button – Repair asset

The mouse is used to perform free look functions including look up, look down, rotate left, and rotate right. The keyboard controls movement using standard WASD keys for forward, backward, strafe left, and strafe right. Jump movement is controlled by the spacebar. Firing the weapon is controlled by the right mouse button. Initiating repairs is controlled by the left mouse button.

## Sound Effects

The game employs a number of sounds:

- Atmospheric wind and weather
- Footsteps – running
- Gunshots
- Gun empty
- Bullet impact
- Pickup ammunition
- Repairing asset
- Generator faulty
- Generator repaired

## Section VI - Artificial Intelligence

### Opponent AI

Opponent AI will be achieved through a number of algorithms. An algorithm allowing the selection of the nearest target will allow the enemy to identify either the player or asset as its target. Once a target has been established a path finding algorithm will allow the enemy to navigate toward the selected target. As the enemy approaches the target, a line of sight algorithm will trigger the enemy to begin attacking the target. Each enemy will have an unlimited amount of ammunition. The firing rate will be limited based on the specific enemies intended level of difficulty.

## Section VII – Technical

### Target Hardware

Recommended system requirements

- Windows 7 or newer
- 2GB of RAM
- Processor with SSE2 instruction set support
- nVidia or AMD graphics DX9 shared model 2.0

### Development hardware and software

The game is being developed in the pc version of Unity 4.5. Assets are being generated using Blender 2.72b. Testing will be performed on a Windows 7 64 bit pc with Direct X 11 and a Radeon HD6950 2GB graphics card, Intel I3 3.1GHz processor, and 8 GB of RAM.

## Section VIII – Game Art.

### Characters









Environments





