Abstract Defender
Technical Design Document

Created by: Traveen Weerasooriya

Changelog

V0.1 (23/05) - Creation of document and basic information added
V0.2 (31/05) - All headers laid out. More information added
V1.0 (9/06) - Changed software for developing art assets, added game loops and finalised technical risk assessment

Table of Contents

Development Environment

Software Requirements

Software	License	Version	Users	Purpose
Unity 2D URP	Free	2021.3.13f1	Designers, Programmers, Artists	Main Development of Game
Visual Studio 2022	Community	17.3.6	Programmers	IDE for Scripts
Paint.NET	Classic	5.0.3	Artists	Creation of art assets

Project Standards

File Naming

- Scripts - PascalCase without spaces (eg: EnemySpawner)
- Names in lowercase, with specific prefix and spaces are replaced with underscores:
    - Sprites - spr prefix (eg: spr_shrapnel_launcher)
    - Materials - mat prefix (eg: mat_yellow)
    - Textures - tex prefix (eg: tex_red)

Coding practices

- Naming conventions
    - Variables - camelCase without spaces (eg: enemyBaseSpeed)
        - Variable names must be noun phrases (describe what it is)
    - Functions including Coroutines - PascalCase without spaces (eg: GetMouseInput)
        - Function names must be verb phrases (describe what it does)
- Commenting
    - Always summarise the purpose of function before it is declared.
    - Comment over any more complex sections of functions to describe its usage, if needed
- Other
    - All parameters of custom scripts need to be accessible and modifiable - no magic numbers.
    - Limit modification of finished scripts unless absolutely necessary to make a certain action work.

Version Control

Repository and Contributors

https://github.com/TraveenW/td-tower-defender.git

Asset Reviewer and Main Contributor: Traveen Weerasooriya

Commit Message Format

All commit messages should follow this format , and all content in the commit must be reviewed before pushing.

ID : TYPE : SCOPE **Content**
ID: A unique identifying number for the commit. MUST be unique. Eg: #7, #25, #156, etc.
Type: The type of change. Adhere to these tags when listing the type. Their meanings can be usually inferred: FEATURE, TEST, REFACTOR, FIX
Scope: What part of the project is being changed. Although there aren't set rules for terms, it should still be clearly defined what's being affected. Eg: MENU, SANDBOX, SPRITE, etc.
Content: The content of the commit message. It should adhere to the following formatting: <symbol> <summary> + <added an item> - <removed an item> * <changed an item, or other message>
Example 1: #35 : FEATURE : MENU + Added "Quit Game" button to main menu
Example 2: #62 : REFACTOR : ENEMY * Simplified enemy splitting script
Example 3: #112 : FEATURE : SANDBOX + Added finished sprite for "Crossbow" - Removed test sprite for "Crossbow"

Game Overview

Genre	Tower Defense, Arcade
Target Platform	Windows PC (itch.io)
Point of View	2D Orthographic, Top Down

The player is a static tower in the middle of the screen that utilises a range of weapons to defend against a variety of incoming enemy shapes. The enemies come in random waves that get increasingly more difficult as time passes. These enemies are represented by basic regular shapes: triangles, squares and hexagons.

Windows PC Considerations

- Many PC gamers primarily use keyboard+mouse input. Controls must be prioritised for those inputs.
- Because this is releasing on itch.io, the game should be able to be zipped and uploaded; and then later downloaded and unzipped with no risk of substantial data loss or corruption.
- Windows PCs come with varying capabilities. This includes the hardware in the units, the screen aspect ratios of their monitors, and the ability to support several types and qualities of inputs.
  Features in the game may have unintended effects on certain systems, which may not be possible to detect during testing.
    - Although Windows 11 has been released, there are many people still using Windows 10 and below. The game should aim to be compatible for at least these systems.

Minimum Specs

- OS: Windows 7 or later
- Processor: Intel Core 2 Duo 2.1 Ghz or equivalent
- Memory: 2 GB RAM
- Graphics: 2nd Generation Intel Core HD Graphics (2000/3000), 512MB
- DirectX: Version 9.0
- Storage: 90 MB available space

Game Controls/Input

The game is entirely controlled with the mouse.

- Left Mouse Button
    - Confirms Main Menu options
    - Shoots active player weapon toward mouse cursor
- Right Mouse Button
    - Cycles between the player weapons (shortbow -> shrapnel launcher -> crossbow)

Feature List

- Player controls
    - Shoot where player aims the mouse cursor
    - Cycle between a variety of weapons
- Timer system - Tracks the elapsed time since the start of the game
- Hit counter system - Tracks how many times the player hit enemies in the game
- Enemy features
    - Random spawns using pools of enemies that change over time
    - Health determined by colour of enemy
    - Enemy is knocked back when hit by player projectile
    - Other properties (speed, ability to split) determined by type/shape of enemy
- Game Over screen
    - Displays Time attained and number of enemy hits in the game session
- Game Difficulty selection
    - Choose a specific game difficulty to change the speed of all enemies

Game Structure

Main Menu

- The main menu screen is overlaid on top of the main game scene. Enemies do not spawn, but the player is still able to use all their weapons.
- The top left of the screen has the title of the game
- The middle left of the screen has all the controls of the game listed
- In the main menu, player can select between 4 options
    - **Game difficulty options** - Choosing any of these options will start the game by instantiating the enemy spawn controller, and editing settings of the enemy's base spawn rate and speed
        - **Easy** - Top right of screen. Starts the game and decreases spawn rate and speed by 10%
        - **Normal** - Middle right of screen. Starts the game without any modifications.
        - **Hard** - Bottom right of screen. Starts the game and increases spawn rate and speed by 10%
    - **Quit Game** - Located at the bottom left of the screen. Closes game executable

Main Gameplay Mode

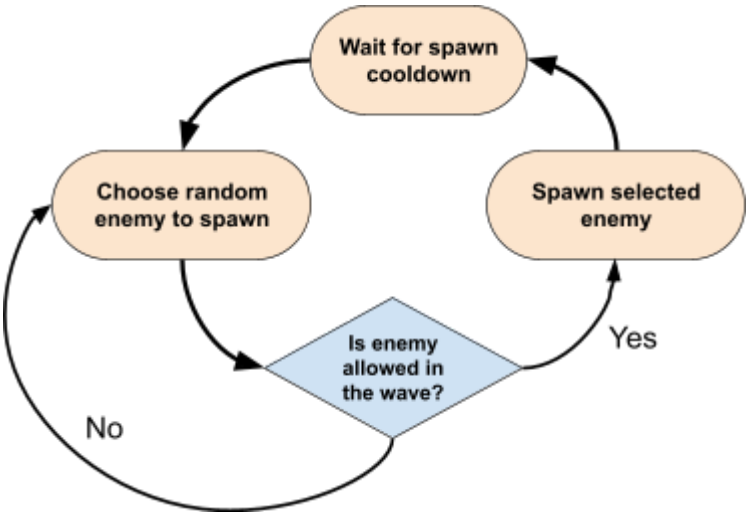
- Main Objective - The player must survive as long as possible
- Objective Trackers
    - Times how long the game has been running
        - Event manager relies on the timer to manage the pools of enemies that spawn over time
    - Hit tracker for number of times enemies were hit by player's projectiles
- Main Challenge - Enemies that spawn at a set radius around the player. These enemies have varying amounts of health and speed, and they get more difficult as time progresses.

Game Over screen

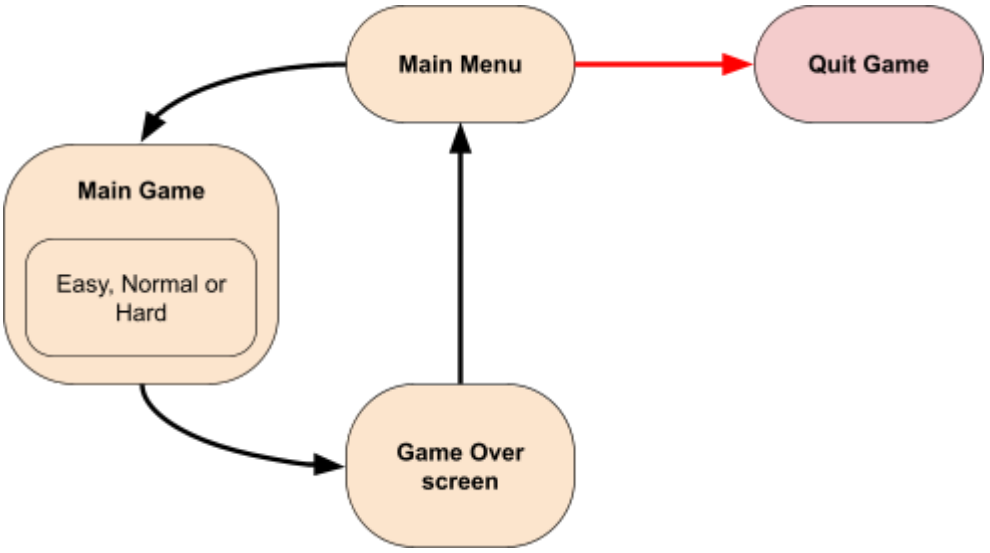
- Triggered when enemy collides with player hitbox in the Main Game
- Stops the game to display a screen with various elements:
    - Game over text
    - A display for the time tracker
    - A display for the hit tracker
    - A prompt for restarting the game
- When all elements are displayed, any mouse input can restart the game back to the Main Menu

Gameplay Loops

Enemy Spawning



Game Mode flow



Game Mechanics

Player Shooting

- Player character is in the middle of the playable area aiming towards the mouse cursor.
- Player shoots where the character aims with Left Mouse Button

Player's Arsenal of Weapons

Weapon features

There will be 3 weapons in the game: Shortbow, Crossbow and Shrapnel Launcher. Those weapons have different properties including:

- Number of projectiles
    - Shoot projectiles in an arc
- Fire delay
- Projectiles from specific weapons piercing through enemies
- Projectile range before disappearing

Eg: Shrapnel Launcher

- Fires six projectiles, evenly spaced at a 120 degree arc centred on mouse cursor
- Long fire delay (0.9s)
- Projectiles can only hit 1 enemy
- Projectiles have a short range before disappearing

Cycling between weapons

- Cycle between the three weapons with the Right Mouse Button
- When swapping to another weapon, they have to wait the fire delay before shooting
- Weapon cycle: Shortbow -> Crossbow -> Launcher

Time Tracker

- Time tracker shows player how long the game has run for
- Activates and become visible when player starts
- On Game Over, stop tracker, display time and then reset

Hit Counter Tracker

- Hit Counter tracker shows player how many times they hit enemies in the game
    - Increments everytime player weapon projectile hits enemy
- Similar behaviour to Time Tracker when player starts, and on Game Over

Enemy Health

- Health determined by colour of enemy
    - Changes colour when going down health
- Order of health from lowest to highest:
  Red - Green - Blue - Black

Enemy Movement

- Heads directly towards the player upon spawning
- All enemies have a base speed, with a certain speed modifier
- Speed modifier determined by shape of enemy
    - Enemy shapes ordered by speed from fastest to slowest:
      Triangle - Square - Hexagon

Enemy Splitting

- Enemy is knocked back when hit by player projectile
- Some enemy shapes have the ability to “split” when they are hit
    - Enemy turns into 2 of the lower grade enemy type with lower health
    - Those enemies get knocked in the opposite direction of the player, with a small deviation.
    - Does not apply if that enemy type already has 1 health
- Splitting shapes:
    - Hexagons -> Squares
    - Squares - Triangles

Enemy Spawning

- Enemies are organised into pools
    - The enemy that spawns is randomly determined by the types enabled in the pool
- Random spawns using pools of enemies to determine what enemy types are allowed to spawn
- Spawn at a random point that is a set distance from the player
- There is a spawn rate tied to each pool, determining how many enemies spawn in one second

Postgame Enemy Spawning

At a certain point in the game, there will be no more new enemy pools to pull from. This will be how that point is managed.

- A certain set enemy pool is used for this game section
- Spawn rate starts at a certain amount, and will increase over time

Game Systems

Enemy Wave System

Enemy pools and their corresponding spawn rates last for a certain amount of time. After that, a manager will switch to the next enemy pool and its corresponding spawn rate.

- Involved mechanics:
    - Time Tracker
    - Enemy Spawning

Game Difficulty System

In the Main Menu, a difficulty of the game can be chosen to play through: Easy, Normal or Hard. When chosen, it will modify the enemy's base speed, making them slightly slower on Easy and slightly faster on Hard.

- Involved mechanics:
    - Enemy Movement
- Utilises Main Menu UI

Game Content

Environment

- The entire game takes place in a singular scene at a 4:3 aspect ratio.
    - The arena of gameplay is hexagonal (pointed top) with a purple border to subtly indicate the parts unreachable with the shrapnel launcher.
- Everything in the game consists of sprites, basic polygons, and UI text elements.
- The environment in the scene is fixed, with the only moving parts being the Player character's weapons, their projectiles, the enemies, and small particle systems tied to various objects.

Technical Goals

- 60 FPS or more at the minimum PC specs
- Final game build being able to be zipped and later unzipped without significant data corruption/loss

Technical Risks and Avoidance

Risk	Managing risk
Frame drops from too many objects	Make the enemy actions as simple as possible: ● Only moving in a direction ● Only reorienting once after knockback ● Lowering spawn rates if needed
Loss of data during development	● Backups of data on GitHub (through Sourcetree) ● Backups of data on multiple storage devices

Appendix - Technical choice justifications

Development engine - Unity

Widely popular engine that is also easy to learn, especially for 2D projects. Godot and GameMaker are other 2D options, but would require time training to get used to.

Scripting language - C#

High-level programming language that can make scripts as clear as possible for designers, while granting many capabilities to programmers. It is also the default scripting language for Unity.

Art creation - Paint.NET

Simplified painting software that is also free to use and easy to learn. More lightweight compared to other art software.