IMGD 3000 Project 3 Design Document

Game Name

Too Many Bullets

Team

Brad Cosma - bcosma@wpi.edu
Benny Klaiman - bnklaiman@wpi.edu
Team Name: Danmaku-dan (弾幕団)

Genre

Bullet hell (a.k.a. 弾幕, *danmaku*) is a subgenre of 2D shooters where the screen is filled with moving projectiles, usually ranging from tens to low-hundreds of them onscreen at any given time.

Game Description

The player controls their ship in a two-dimensional space, with the ability to move in all eight directions. The player is constrained to a vertically oriented level of fixed width and height. The entire width of the level is on-screen during gameplay. The level can contain enemies and boss characters, which will appear above the player. The player has a choice of attacks that fire in different patterns with which to defeat the enemies. These attacks fire directly ahead or in fixed patterns, with the player able to toggle between a wide and narrow spread. The player is able to slow their motion for fine movement, which also changes the attack spread. The enemies also have their own attack patterns, which vary based on enemy type, with each enemy type having multiple attacks. Certain attacks may automatically adjust their target point based on the position of the player. The screen is usually filled with hostile projectiles (hereafter defined as “bullets”) and dodging is the main challenge. The game will have a point-based system where the player earns points by narrowly missing bullets (“Grazing”) or killing enemies, and feature “nukes” similar to saucer shoot. Gameplay is slightly random, enemies move and chose their attacks at random based on some parameters, similar to a decision tree or finite state machine model.

Technical Features

- “Slow-down” button: the ability to, when held down, reduce the movement speed of your character by half in order to more precisely navigate complex bullet patterns. This also focuses your own attacks to a narrow forward spread.
- “Graze” points: The player’s hitbox is smaller than their visual sprite. However, players may gain points by having bullets pass over their visual hitbox as a bonus for risky or skillful play.
- Targeted attacks: Attack types that move towards the point the Player occupied when the enemy attacked.
- Bomb: The “nukes” of bullet hell shooters, a type of special attack in limited supply that destroys all on-screen enemy attacks, intended for use when the action becomes too intense to handle.
- Circular calculations: Certain attack patterns may move outwards radially around the point from which they were fired. The bullets created by these patterns required special code to rotate and determine their positions.

Deviations from the Plan

Unfortunately a number of features were cut from the original plan. These include:

- We wanted to implement a difficulty system, but ended up not doing so.
- Enemies were intended to drop point pickups, but this was removed.
- The player was initially intended to be able to choose between the standard attack and a homing attack. This was not implemented, but the Boss does have an attack that targets the player’s position at the time of firing.
- Circular Motion: Having bullets be able to orbit a point was cut from development. Instead, a system where bullets spawn on a circle by rotating a point around another point was implemented. These bullets move linearly outwards.

Artistic Assets

Object Type	Size (width x height) / characteristics
Bullets	1x1, 2x1 up to 7x3
Player Character	5x7, df::RED, df::WHITE for hitbox
Enemy Characters (excluding boss)	3x2, 2 frames
Boss	11x7, 2 frames, red.
Main Screen & Game Over	Very large (full window)

Controls	One line, animated for flair (2-3 frames max)
Firing sounds	Various short clips
Explosion sound	Short explosion clip
Background Music	full BGM is 3:02, plan to trim in order to loop, mp3 format, 128kbps

Distribution of Work

We intended to distribute the work mostly evenly over the course of the assignment time. This did not end up panning out due to various circumstances. Sprites were evenly distributed, and all music and sounds used existed prior to the class or were taken from the internet. Code commit history should be visible in the git repository in the zip file.