Project Plan

for

Afterburner

Team Quebec

Adam Nelson & Kyle Strayer

25 February 2019

Table of Contents

Introduction

Afterburner is a 2D aerial shooter combat game. The player controls a fighter jet and must defeat waves of enemy aircraft by outmaneuvering and outshooting them. As the player progresses through the game and racks up more kills, they will unlock different power-ups that make their aircraft faster, smaller, and more lethal.

After the player completes four consecutive levels successfully, they will be eligible to record their score on the High Score table. However, each run of the game is different, i.e. no progress is recorded. If the player wants to get a higher score, they will have to do it all over again!

The main objective of the game is completing the four levels without being shot down. Once they have, they will be able to play a fifth endless-wave level to increase their score as much as they desire before being shot down/ending the game willingly.

Project Organization

Team

Kyle Strayer
Email: kstrayer@mymail.mines.edu. Phone: 303-803-3823
Kyle will be the lead developer and have the final say in decisions such as implementing new features to the game. He will delegate coding tasks among the team as needed.

Adam Nelson
Email: adamnelson@mymail.mines.edu. Phone: 303-257-3846
Adam will be the primary point of contact between the developer team and the publisher. He will be in charge of managing all logistics with the game and enforce deadlines.

Communication

GroupMe and SMS texting will be the primary methods of communication between team members. Email will be the primary method between the team and the publisher.

Logistics

Trello will be used to track all taskings and deliverables. The project “board” will be available and can be modified by both team members.

Structure

The team will follow the AGILE development process. Team members will update each other daily with what they completed yesterday, their tasks for today, and any issues preventing their progress. Team members will work individually but will meet and develop together on Mondays and Fridays. Tasks will be completed according to the schedule outlined in Task Dependencies/Schedule/Resource Estimation.

Project Technology

Version Control

The project will be hosted on a private Github repository.

Development Software

GameMaker Studio 2 for Windows

Gamemaker will be the primary software used to produce the game, including creating rooms, objects, interactions, and graphics.

- Engine: Direct3D Graphics API
- Languages: GameMaker Language
  - Other languages such as C++ and JavaScript may be required

Audacity

Audacity is a sound editing tool for modifying .mp3 and .wav files for creating in-game sound effects.

List of Tasks

High Priority	**Medium Priority**	**Low Priority**
Player Object	Sprites	Sounds
Enemy Object	Power-Up Objects	Pre/Post-mission screen
First Map	Animations	Boss Maps
Basic object interactions	Maps 2-5	
Title screen with Start/Quit		

Task Criteria

1. Player Object
    a. Player's aircraft can move up, down, left, and right
    b. Moving left or right changes image angle to give the appearance of "turning"
    c. Space bar/shift key fires bullets from aircraft
    d. Aircraft is destroyed upon losing all health; game over
2. Enemy Object
    a. Enemies move similar to Galaga enemies
    b. Sprite rotates depending on the direction of travel
3. First Map
4. Basic object interactions
    a. Player can shoot down Enemies
    b. Enemies can shoot down Player
    c. All objects stay on the map as it scrolls
5. Title screen with Start/Quit
    a. Start button begins the first level
    b. Quit button closes the game
6. Score Counter
    a. Text in the corner of the screen that shows the player's current score based on their number of kills and points gained from power-ups.
7. Sprites
    a. Player has several basic sprites to choose from to give their aircraft different appearances
    b. Enemy aircraft randomly select a sprite
    c. Sprites are designed for power-up objects
8. Power-Up Objects
    a. Missile Object: Allows the player to fire a limited number of missiles that instantly kill enemies, but have a slower fire rate
    b. Half-Health Object: halves the health of all current Enemy objects
    c. Nuke Object: destroys all current Enemy objects but damages player by 50% of total health
9. Animations
    a. Player and Enemy aircraft have flashing thrusters
    b. Player and Enemy aircraft flash when firing ammunition
    c. Power-Up Objects glow and "bounce" up and down
10. Maps 2-5
    a. Maps scrolls from the top down
    b. Map backgrounds are stitched together to give a seamless effect
    c. Additional maps are created with obstacles/ "no-fly" zones included as desired.
11. Boss Levels
    a. First level features heavy bomber

    b. Second level features two-phase laser enemy

12. Sounds
    a. Thrusters
    b. Machine Guns
    c. Missiles
    d. Aircraft destruction
    e. Kill confirmed
    f. Player aircraft destruction
    g. Game Over
    h. Power-Up pick-up
    i. Background music
13. Pre/Post-mission screen
    a. Pre-mission screen allows players to equip additional aircraft/upgrades
    b. Post-mission screen allows players to see their score

Task Dependencies/Schedule/Resource Estimation

Tasks	Timeline
Player Object, Enemy Object, First Map, Basic Object Interactions	25 Feb
Milestone Delivery #1	**25 Feb**
Title Screen with Start/Quit, Sprites for Aircraft, Score Counter	1 Mar
Power-up Objects	8 Mar
Milestone Delivery #2	**8 Mar**
Animations, Maps 2-4	15 Mar
Alpha Release	**22 Mar**
Sounds	29 Mar
Boss Levels	10 Apr
Beta Release	**10 Apr**
Pre/post mission screen	17 Apr
Final bugfixes	22 Apr
Final Release	**22 Apr**

Delivery Descriptions

Milestone Delivery #1

The player begins the game by controlling an aircraft, using the arrow keys to move and the spacebar to fire bullets at enemy aircraft. The enemy aircraft shoot back at the player and fly across the screen on a predetermined path. When an aircraft takes enough damage they are destroyed. If the player is shot down they lose the game.

Milestone Delivery #2

The main menu is implemented, with two options: start, and quit. The player's current score is shown on the screen. Basic aircraft sprites are added to the game, as well as power-up items that occasionally drop from enemies.

Alpha Delivery

Animations are implemented for most aircraft types. Levels 2-5 are added, with different "bosses" for each level.

Beta Delivery

Sound has been added to the game. The boss level maps have been added.

Final Delivery

Pre/Post-mission screens have been added between maps to give the player information and tell the story.

Risk Analysis

Risk	Effect	Probability	Plan
Platform cannot support Game	Player is unable to run game on their system	Medium-Low	Our Game will be developed for Windows, which is the most popular operating system for personal computers. A large percentage of our target audience will be able to successfully run the game.
Failure to manage project	Deadlines are not met, project gets behind in development	Medium-High	We will mitigate this risk as best as possible by being strict with our time management, enforcing deadlines and keeping each other accountable to our assigned tasks. If we don’t think we can meet a deadline, we will discuss how we will make up for lost time later in the project.
Lack of knowledge	Development takes longer than expected	High	To account for our novice experience in game development, we will both commit to independent research on game development, including but not limited to online video tutorials.
Too ambitious requirements	We are unable to implement all of the aforementioned features	Low	We have placed the most important tasks first. The later features are not critical to the overall gameplay, so if we are unable to implement them it will not have a significant effect.
Bad design	Misorganization causes coding delays.	Medium-Low	We will make our design modular and slowly add features, carefully tracking their development and integration through our daily meetings.

Test Plan

Internal Testing

Each week, the team will review the game by playing it and taking notes in a journal, recording any bugs that they may encounter. Any other thoughts, ideas, questions, or interesting events will be documented as well.

External Testing (Quality Assurance)

To ensure that gameplay is natural and easy to understand by players, we will periodically bring in outside parties (classmates, other students, friends, etc) to test the game and fill out a short feedback survey on their experience, including what they liked, didn’t like, and anything they would change.