

Project Write-up

[Analysis 2](#_Toc118973334)

[Problem spec 2](#_Toc118973335)

[Computational methods 2](#_Toc118973336)

[Research 2](#_Toc118973337)

[Stakeholders 3](#_Toc118973338)

[Developer 3](#_Toc118973339)

[Player 3](#_Toc118973340)

[Tester 3](#_Toc118973341)

[Parent 3](#_Toc118973342)

[Maintainer 3](#_Toc118973343)

[Essential features 3](#_Toc118973344)

[Limitations 3](#_Toc118973345)

[Requirements 4](#_Toc118973346)

[Success criteria 4](#_Toc118973347)

[Design 4](#_Toc118973348)

[Graphic Design 4](#_Toc118973349)

[Decomposition 5](#_Toc118973350)

[Structure 5](#_Toc118973351)

[Algorithms 5](#_Toc118973352)

[Usability 5](#_Toc118973353)

[Data Structures 6](#_Toc118973354)

[Test Data 6](#_Toc118973355)

[Further Data 6](#_Toc118973356)

[Iterative Development 6](#_Toc118973357)

[Iterative Stages 6](#_Toc118973358)

[Prototypes 6](#_Toc118973359)

[Modularity 6](#_Toc118973360)

[Annotations 6](#_Toc118973361)

[Naming 6](#_Toc118973362)

[Validation 6](#_Toc118973363)

[Review 6](#_Toc118973364)

[Evaluation 7](#_Toc118973365)

[Testing Evidence 7](#_Toc118973366)

[Failed Tests 7](#_Toc118973367)

[Alpha and Beta Testing 7](#_Toc118973368)

[Usability Testing 7](#_Toc118973369)

[Success Criteria vs Test Evidence 7](#_Toc118973370)

[Addressing Unmet Criteria 7](#_Toc118973371)

[Usability Features Evaluation 7](#_Toc118973372)

[Addressing Unmet Usability 7](#_Toc118973373)

[Maintenance/Limitations 7](#_Toc118973374)

[Future Development 7](#_Toc118973375)

[Quality 7](#_Toc118973376)

# Analysis

## Problem spec

* Computer Game
* Fun and engaging
* Skill based – quick reactions and quick assessment of moving objects
* Competitive

## Computational methods

These are the computational methods used in this game:

* Computer game
  + Graphics abstraction by using 2D instead of 3D
  + Aircraft abstracting by
  + GUI
  + Keyboard (or optional touch input)
  + Animations
* Modelling flight of projectiles
* Equations of constant acceleration

## Research

There are other similar games to this game such as the [Pygame-a-primer](https://realpython.com/pygame-a-primer/) game, which involves an aircraft that can be moved around using the arrow keys to dodge missiles coming towards it and when the aircraft is hit, the game ends. Some code (aircraft code, enemies code to help with the targets) from this game has been used in my game. A [scrolling background](https://coderslegacy.com/python/pygame-scrolling-background/) is harder to get working, but this website’s program does this very well. This has been very useful to get the scrolling background working properly.

List of Pygame websites and tutorials:

[Real Python](https://realpython.com/pygame-a-primer/)

## Stakeholders

### Developer

The developer – me – is the person who writes the code and thinks what code to write where in the code using the least amount of code possible to achieve the goal of writing the weather forecasting program

### Player

The player is the person who plays the computer game for many years, playing for many hours each day.

### Tester

The Tester is someone other than the developer one who rigorously tests the program, checking for any bugs that may occur when using the application for playing the game.

Parent

A parent is an older person who checks the user of the application

Maintainer

The person who maintains the code, checks the code

## Essential features

These essential features are essential for the game to work as it should. These are shown along with their relevant priorities in the table below:

|  |  |
| --- | --- |
| **Feature** | **Priority** |
| Animated aircraft | Must |
| Animated bomb | Must |
| Background with scenery | Must |
| Calculation of hits and misses on targets | Must |
| Anti-aircraft missiles | Should |
| Keyboard input | Must |
| Touch input | Could |
| Sound effects | Could |
| Scoring | Should |
| Bomb count | Should |
| High score recording | Could |
| Resizable window | Could |

## Limitations

Although there are many great features in the game, there are also unfortunately some limitations, which are shown below:

* Single player
* 2D graphics
* Python installation
* Pygame installation
* Supported operating system for Python/Pygame

## Requirements

The following are the functional requirements and non-functional requirements:

1. Load an animated background and show on screen
2. Load an aircraft and show on screen
3. Simple keyboard controls to control the aircraft
4. Simple keyboard controls to control the releasing of bombs
5. Simple touch controls to control the aircraft
6. Simple touch controls the releasing of bombs
7. Have a resizable window
8. Computer/tablet running Android/Mac OS/Windows
9. Python installation
10. Pygame installation
11. No fatal crashes
12. Program works as it should
13. Have an easy-to-use GUI rated by beta testers
14. Have >75% 5-star reviews by users
15. Look visually appealing to >75% of beta testers
16. Bright, contrasting colours for people with colour blindness
17. Sound effects to help the player know when they have hit a target

## Success criteria

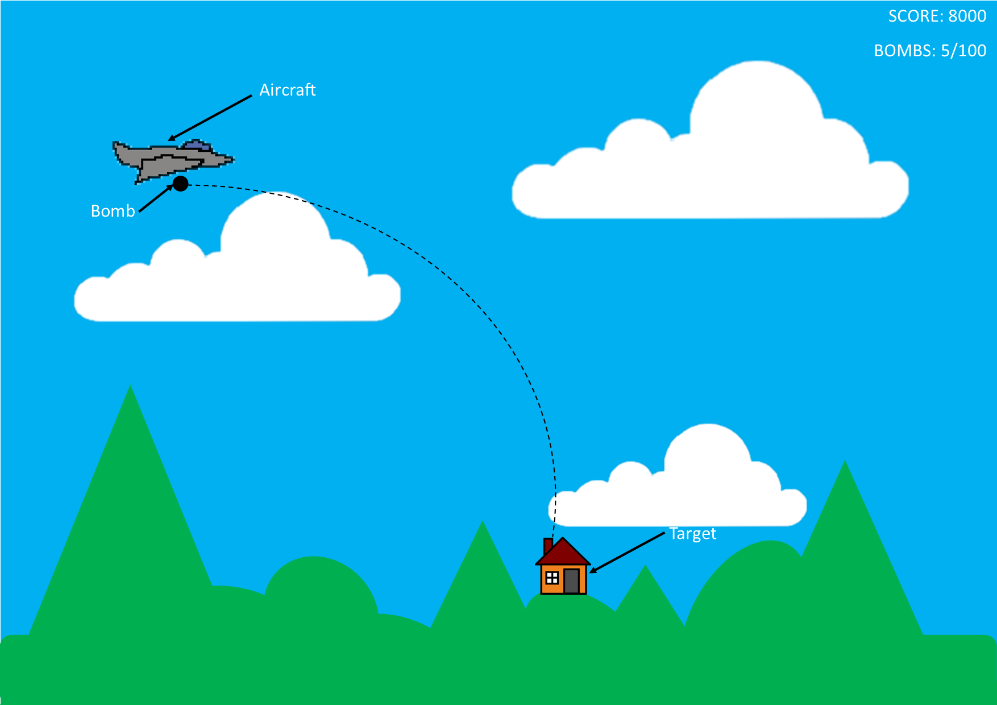
The following are the criteria that will make the game a success:

* An aircraft that can be controlled easily with the keyboard
* An aircraft that can be controlled easily with touch
* Scrolling background with targets placed on top
* Moving targets that explode when hit with a bomb
* Bomb released when spacebar is hit
* Bomb count that increases when a bomb is used
* Scoring – when a target is hit/not hit

# Design

## Graphic Design

Below is a mock-up of what the game will look like:



The background has been created with Microsoft Publisher, using shapes to created the mountains and hills.

## Decomposition

The game is broken down into these smaller sub procedures to make coding the game easier:

* Background
* Aircraft (keyboard input)
* Aircraft (touch input)
* Animation (simulated movement of aircraft)
* Bomb trajectory
* Targets
* Scoring
* Bomb count

## Structure

The game’s structure is shown below:

* Written in Python
* Object-oriented programming
* Pygame framework for games

## Algorithms

The algorithms used in the game are shown below:

* Using equations of constant acceleration, trajectories of bombs will be calculated
* Calculation of whether targets are hit or not based on trajectory
* Calculation of wind influence on projectiles

## Usability

The game has the following features to aid usability for users:

* Bright, contrasting colours for people with colour blindness
* Simple keyboard controls to control the aircraft and the releasing of bombs
* Simple touch controls to control the aircraft and the releasing of bombs
* Sound effects to help the player know when they have hit a target
* Visual explosion to help the player know when they have hit a target

## Data Structures

* Ordered list to store high scores/bombs used when score reached

## Test Data

* No data is required to initiate the game

## Further Data

????

# Iterative Development

## Iterative Stages

The game is broken down into stages in the “Decomposition” section, which is shown below:

* Background
* Aircraft (keyboard input)
* Animation (simulated movement of aircraft)
* Bomb trajectory
* Targets
* Scoring
* Bomb count

## Prototypes

* To be added later

## Modularity

* Object-orientated programming is used to decompose the problem into modular chunks

## Annotations

* Code is complemented with appropriate comments to help readability and usability

## Naming

* Variables are named with meaningful names to help readability and usability

## Validation

* To be added later

## Review

* Review to be added later

# Evaluation

## Testing Evidence

* To be added later

## Failed Tests

* To be added later

## Alpha and Beta Testing

* To be added later

## Usability Testing

* To be added later

## Success Criteria vs Test Evidence

* To be added later

## Addressing Unmet Criteria

* To be added later

## Usability Features Evaluation

* To be added later

## Addressing Unmet Usability

* To be added later

## Maintenance/Limitations

* To be added later

## Future Development

* To be added later

## Quality

* To be added later