CST8333

Assignment 2

Space Guy: Blockade

Andrew Gaal

March 23, 2022

All material prepared for this assignment was produced by the author. Material from all third parties has been cited and referenced.

Table of Contents

[1 Introduction 2](#_Toc100819842)

[1.1 Objectives 2](#_Toc100819843)

[1.2 Scope 2](#_Toc100819844)

[1.3 Timeline 3](#_Toc100819845)

[1.3.1 Milestones and Deliverables 3](#_Toc100819846)

[1.4 Risks 4](#_Toc100819847)

[1.5 Assumptions 4](#_Toc100819848)

[1.6 Technical Constraints 4](#_Toc100819849)

[1.7 Budget 5](#_Toc100819850)

[2 Status 6](#_Toc100819851)

[2.1 Introduction 6](#_Toc100819852)

[2.2 Accomplishments 6](#_Toc100819853)

[2.3 Goals 6](#_Toc100819854)

[2.4 Roadblocks 6](#_Toc100819855)

[2.5 Lessons Learned 6](#_Toc100819856)

[3 Tools and Technology 8](#_Toc100819857)

[3.1 Introduction 8](#_Toc100819858)

[3.2 Evaluation 9](#_Toc100819859)

[3.3 Selection 10](#_Toc100819860)

[3.4 Conclusion 10](#_Toc100819861)

[3.5 Requirements Traceability Matrix 10](#_Toc100819862)

[4 Source Code Implementation 12](#_Toc100819863)

[4.1 Introduction 12](#_Toc100819864)

[4.2 Coding Checklist 12](#_Toc100819865)

[4.3 Submit Code 12](#_Toc100819866)

[5 References 13](#_Toc100819867)

# Introduction

In this section background information on the project is provided, including the reasons for undertaking the project, specifically the business problem to be solved and how the proposed system will solve it, as well as the key stakeholders who will benefit from the project results.

## Objectives

In this section measurable project objectives, business outcomes to be derived from achieving the objectives, and the measurement criteria to be used to confirm that an objective and the outcome have been achieved are listed.

Table 1: Objectives and Business Outcomes

| **No.** | **OBJECTIVE** | **BUSINESS OUTCOME** | **MEASUREMENT CRITERIA** |
| --- | --- | --- | --- |
| 1 | Deliver a finished piece of software written using Python by the end of April 2022. | Developer learns foundational Python skills. | Increase in marketability and profitability in the job market. |
| 2 | Develop a top-down arcade style space shooter game. | Finished piece of software for the developer’s portfolio. | Increase in marketability and profitability in the job market. |
| 3 | Deliver fast-paced, challenging gameplay. | Cultivation of a dedicated player base. | Increase in number of downloads on game marketplaces (e.g., itch.io) |

## Scope

In this section the features and functions that characterize the product, service, or result to be delivered by the project are described. That is, the major activities that must be completed to complete the project are listed. Activities that are out of scope for the project also are listed to reduce ambiguity.

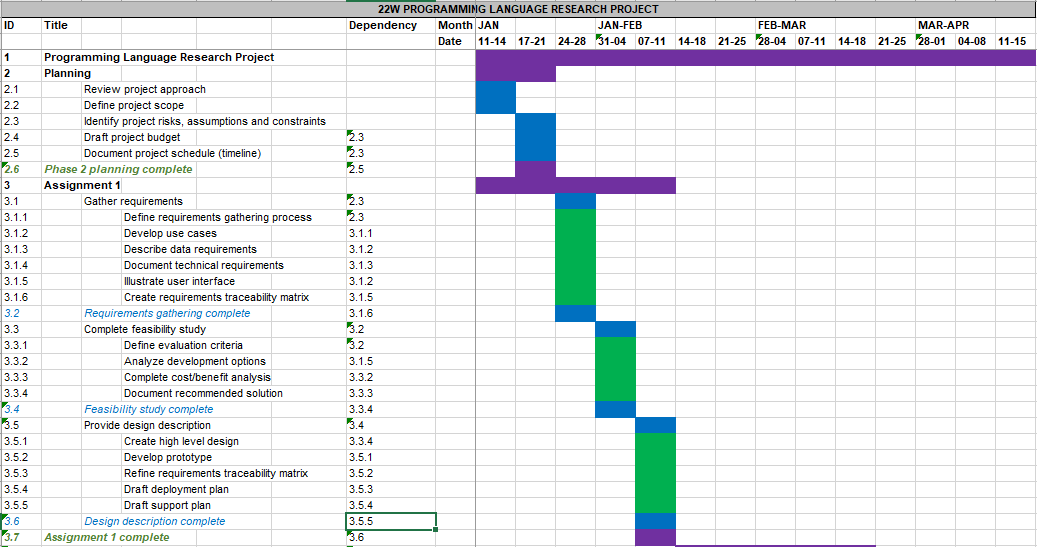
Table 2: Project Scope

| **ACTIVITIES IN SCOPE** | **ACTIVITIES OUT OF SCOPE** |
| --- | --- |
| 1. Requirement Collection and Analysis | 1. Apply for business financing |
| 2. Feasibility Study | 2. Direct requirement elicitation techniques, e.g., interviews, brainstorming, workshops |
| 3. Project Planning |  |
| 4. Testing |  |
| 5. Deployment |  |

## Timeline

In this section the project timeline is illustrated. The project duration is based on the CST8333 course calendar. Tasks included in the timeline are based on checklists included in CST8333 course modules. It is understood that unforeseen events and changes may result in revisions to the project timeline.

Table 3: Project Timeline



Legend

* Purple bars = Level 1 tasks
* Blue bars = Level 2 tasks
* Green bars = Level 3 tasks

[](Project%20Timeline.xlsx)

### Milestones and Deliverables

In this section significant events in the project and their associated deliverables are defined.

Table 4: Project Milestones and Deliverables

| **MILESTONE** | **DATE** | **DELIVERABLES** |
| --- | --- | --- |
| 1. Assignment 1 complete | February 10, 2022 | Written report and slide presentation |
| 2. Assignment 2 complete | March 17, 2022 | Written report and slide presentation, 60% of source code complete |
| 3. Assignment 3 complete | April 14, 2022 | Written report and slide presentation,  100% of source code complete |

## Risks

Project risks are uncertain events or conditions that, if they occur, have positive effects (opportunities) or negative effects (threats) on one or more project objectives, such as scope, schedule, cost, and/or quality.

In this section the principal project risk is identified the likelihood it will occur is estimated (high, medium, low), its impact if it occurs is estimated (high, medium, low), and mitigation strategies are described (how likelihood and impact will be minimized).

Table 5: Project Risks

| **No.** | **RISK DESCRIPTION** | **PROBABILITY (H/M/L)** | **IMPACT (H/M/L)** | **MITIGATION** |
| --- | --- | --- | --- | --- |
| 1. | Schedule slippage | M | H | Track project scope and timeline |

## Assumptions

Assumptions are factors that you believe to be true, although they are not confirmed to be true. Assumptions add risk to a project since it is possible that they will turn out to be false. Assumptions can impact any part of your project life cycle and resulting solution implementation, so it is important that they be documented.

In this section the principal project assumption is identified.

Table 6: Project Assumptions

| **No.** | **THE FOLLOWING IS ASSUMNED** |
| --- | --- |
| 1. | Fundamentals of new programming language will be learned and put to use, timely, to complete project |

## Technical Constraints

Constraints are fixed boundary conditions or limits on what you can do. They are the things you cannot change but that you need to be aware of and manage to. Technical constraints focus on architecture decisions that may limit your solution design. They tend to be inflexible and unchanging and may impact your solution implementation. They include areas such as development languages, hardware, other infrastructure, and software that must be used for your project.

In this section the principle technical constraint is identified.

Table 7: Technical Constraints

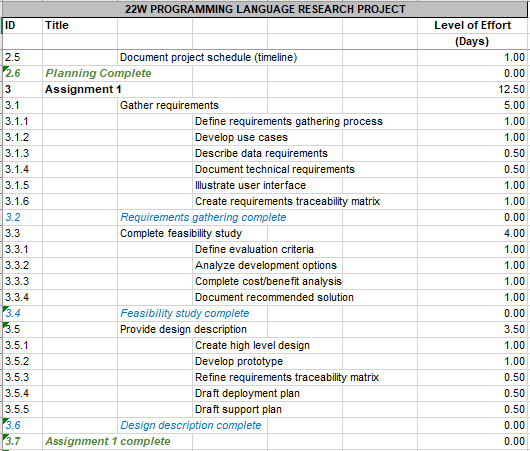
| **No.** | **TECHNICAL CONSTRAINTS** |
| --- | --- |
| 1. | Programming language selected does not accommodate all of the functionality desired in the solution |

## Budget

The project budget is a tool that is used to estimate all the costs that are likely to be incurred before the project is completed.

In this section a preliminary budget is estimated. Only in-scope items, as identified in section 1.3 above, are included. Out of scope items are excluded. It is understood that unforeseen events and changes may result in revisions to the project budget.

Table 8: Project Budget



[](Project%20Budget.xlsx)

# Status

## Introduction

This section details the progress made on the project since the last report. Included are the high-level goals that were set for this period of work, specific accomplishments, roadblocks encountered, and lessons learned during development.

## Accomplishments

The project is proceeding on schedule and is now in a playable state. Features implemented include, but are not limited to:

* Refactored class architecture, now far less tightly bound than the original design.
* Implemented Pygame Sprite class in place of Game Object class.
* Implemented Pygame Group class in place of Game Object Manager class.
* Integrated Scenes into main() method event loop (see Roadblocks section below).
* Implemented Menu Scene, navigable with the mouse, and including “Start Game” and “Quit” buttons.
* Implemented Game Scene, including a mouse-controlled player, lives counter, score display, and periodically spawning enemies with a variety of behaviours.

An updated list of requirements and their current state of completion can be found in Section 3.5: Requirements Traceability Matrix.

## Goals

* Learn the fundamentals of Python
* Learn the fundamentals of the Pygame Library
* Adapt the architectural design from Assignment 1 to work with Python and Pygame
* Implement enough source code to bring the game to a playable state

## Roadblocks

Significant modifications to the architecture proposed in Assignment 1 had to be made due to the constraints imposed by Python and the Pygame library. For example, interfaces and abstract classes are not present in Python, so design patterns and architectural choices had to be modified in order to be implemented. Additionally, the architecture of Pygame demands that many functions must be implemented directly in the main() method. This made encapsulating certain functions difficult, particularly because the classes in the original design were fairly tightly bound. As a result, there is some code duplication present in the main() method, which could lead to problems with maintainability down the road.

## Lessons Learned

Altogether, the project is proceeding on schedule. The goals that were set at the beginning of this work period were accomplished, although not entirely according to plan.

While some modifications had to be made to the planned architecture due to incompatibilities with the tools chosen for the job, the biggest changes to be made were due to the fact that the same architecture was fundamentally flawed. In particular, the original design called for a series of nested classes, each with fairly complicated behaviour, creating layers upon layers of dependencies to wrangle with each time a Game Object’s methods needed to be called. This problem was resolved by separating those components from one another and instead having them communicate via an event system. This solution also had the benefit of improving maintainability drastically, since changes to one class no longer cascade into failures in other classes.

In the future, care will be taken to ensure that classes are designed to function more or less independently of one another rather than being so tightly bound.

Table 9: Project Status

| **Activity** | **Completed** | **If “No” Please Explain** |
| --- | --- | --- |
| Completion of Requirement Analysis phase. | Yes |  |
| Completion of project related Feasibility Study. | Yes |  |
| Completion of Project Design along with your Project Title and definition. | Yes |  |
| Submission of Power Point Presentation includes Requirement Analysis, Feasibility Study and Project Design phases of your project. | Yes |  |
| Submission of Partially done Project Report [word file], having detailed information about the first three modules of your project. | Yes |  |
| Feedback taken from facilitator regarding all your submissions [presentation and report]. | Yes |  |
| Discussion between Facilitator and Developer regarding any updates suggested by the facilitator. | Yes |  |
| Submit the final presentation and final project report to the facilitator within the given deadline. | Yes |  |
| Facilitator approved the final submission-1 and allow you to start next phase of your development. | Yes |  |

# Tools and Technology

## Introduction

There is no “best” programming language. Programming languages are tools that are used to solve problems *(What Programming Language Should I Learn in 2020? Beginner’s Guide, 2019).* As such, as is shown below, a determining factor in choosing a programming language is the type solution that is to be built.

**Front-end Web Development:** Front-end development entails creating a user-friendly interface for a given website or web application. Programming languages used for front-end development include HTML, CSS, and JavaScript.

**Back-end Web Development:** Back-end, or server-side development entails creating the features that allow users to interact with data stored on a server. Programming languages used for back-end development include PHP, Python, Ruby, JavaScript (and Node.js), Java, and C#.

**Mobile Development:** Mobile development entails creating mobile applications. Programming languages used for mobile development include Swift and Objective-C (for iOS), Java (for Android), and C# (for Windows).

**Game Development:** Game development generally entails creating stand-alone applications, although there is often some overlap with other categories such as mobile or front-end web development. Many programming languages can be used for game development. Popular languages include Java, C# and C++, as well as scripting languages such as Lua and Python.

**Data Analysis:** In this field, development entails creating solutions for data analysis, machine learning, and artificial intelligence (AI). Languages used include Python and R.

As discussed in section one of this document the current project involves game development. As such, the following languages were evaluated against their abilities to enable the fulfilment of the project requirements.

* Python
* C#
* Java

A set of objective selection criteria was used as an aid in identifying the programming language of choice for the current project. “Popularity” is one such factor, as it is a potential indicator of ease of troubleshooting, as well as demand in the job market. However, this quality is not, by necessity, analogous with popular developer specializations as the job market for developers evolves continuously. For instance, at a given point in time Java may be ranked as one of the most popular programming languages. However, as is shown, below, in “The Importance of Being Earnest (TIOBE) Community Programming Index” (*TIOBE Index | TIOBE - The Software Quality Company,* 2018) the popularity of programming specializations, such as Java, fluctuates over time.

Figure 1: TIOBE Programming Community Index

A close up of a map

Description automatically generated

In addition to showing increases and decreases in popularity, the trendlines in the TIOBE index may indicate those languages whose status is trending down over time, such as C++, versus those that are trending up, such as Python. In summary, in addition to programming language popularity, trends in programming specialization over time is a factor to be considered in the selection of the programming language for the current project.

## Evaluation

In this section alternative programming languages are evaluated against the following criteria.

* Solution: Language supports the planned application and architecture
* Ease of program verification: Verifying that program correctly performs its required function
* Support and community: Community support includes wikis, forums, tutorials, and additional libraries, all of which help the language grow
* Availability of an IDE: A powerful Integrated Development Environment goes a long way in increasing the productivity of a programmer. The language with an IDE of well supported development, debugging and compilation tools should be selected.
* Popularity: Commercial demand for developers with experience in the language is stable or trending upwards.

Table 10: Selection Matrix

| Criteria | Python | C# | Java |
| --- | --- | --- | --- |
| Solution | √ | √ | √ |
| Ease of Program Verification | √ |  |  |
| Support and Community | √ | √ | √ |
| Availability of an IDE | √ | √ | √ |
| Popularity | √ | √ |  |

## Selection

In this section recommendations are prioritized according to the results of the evaluation of the alternatives, beginning with the preferred alternative.

My first recommendation is Python. Python is steadily increasing in popularity compared with the other solutions and frequently appears in job postings as a required or preferred skill. It is also the language that I am least familiar with, so dedicating time to learning it will result in the greatest benefit. While Python is generally used for back-end web development, there are tools available for packaging Python programs as stand-alone applications, as well as a robust game development community. Additionally, since Python is a scripting language, programs written with it do not need to be compiled before they are run. This makes it ideal for rapid iteration and testing. Finally, Python can be developed in Microsoft Visual Studio Code, which is a well-supported and lightweight IDE.

My second recommendation is C#. C# is not growing as rapidly as Python, but is fairly stable in terms of popularity, particularly in game development. C# is fairly good in terms of program verification, but not as instant as Python. There is also a fairly active community and many good resources available, including thorough documentation and IDE support. The deciding factor is that I am already familiar with C#. Because of this, developing a project in C# would not be as useful as a learning exercise as it would be with Python.

My third recommendation is Java. Java is well supported and documented, suitable for game development, and enjoys popularity with employers. However, this popularity is trending downwards and may soon be overtaken. Additionally, like with C#, I am already familiar with Java so opportunities for personal growth are limited in this context.

## Conclusion

The purpose of this tools and technology evaluation was to identify the programming language that is best suited to building the solution to the problem of developing this game. This report offered three alternative languages: Python, C# and Java. In addition, the evaluation measured each alternative against objective criteria. Python was found to be the preferred alternative. C# was the second ranked recommendation. Java was the third ranked recommendation.

## Requirements Traceability Matrix

This revised version of that requirements traceability matrix is the second step in the process identifying the tests that will be performed to validate whether documented requirements have been achieved.

Table 11: Requirements Traceability Matrix

| NUMBER | CATEGORY | REQ’T | TEST | EXPECTED RESULT | ACTUAL RESULT | PASS/FAIL | COMMENTS |
| --- | --- | --- | --- | --- | --- | --- | --- |
| B-03 | Functional | New Game | “New Game” main menu option is selected | Program transitions to Game scene and a new game begins | Program transitioned to Game scene and a new game began | PASS |  |
| B-07 | Functional | Player hitbox | Mouse is moved over player hitbox | “Player collision” message is written to console | “Player Collision” message written to console | PASS | Debug message since removed from source code |
| B-08 | Functional | Enemy hitbox | Mouse is moved over enemy hitbox | “Enemy collision” message is written to console | “Enemy Collision” message written to console | PASS | Debug message since removed from source code |
| B-09 | Functional | Bullet hitbox | Mouse is moved over bullet hitbox | “Bullet collision” message is written to console | “Bullet Collision” message written to console | PASS | Debug message since removed from source code |
| B-10 | Functional | Player-Enemy collision | Enemy hitbox collides with Player hitbox | Player is destroyed | Player destroyed | PASS |  |
| B-11 | Functional | Enemy-Bullet collision | Bullet hitbox collides with Enemy hitbox | Enemy is destroyed | Enemy Destroyed | PASS | Enemy hit points to be implemented later |
| B-12 | Functional | Enemy destruction | Enemy hit points reduced to 0 | Enemy is destroyed | TBD | TBD | Enemy hit points to be implemented later |
| B-13 | Functional | Losing Lives | Player is hit and destroyed | Player lives reduced by one | Player lives reduced by one | PASS |  |
| B-14 | Functional | Player respawn | Player is destroyed with lives remaining | Player respawns | Player respawns | PASS |  |
| B-15 | Functional | Game over | Player is destroyed with no lives remaining | Program transitions to High Score Screen | TBD | TBD | High Score scene not yet implemented |
| B-16 | Functional | New High Score | Player gets new high score | Score is inserted at the right spot | TBD | TBD | High Score scene not yet implemented |

# Source Code Implementation

## Introduction

During the implementation phase of the software development life cycle (SDLC) the development team creates the agreed upon solution to the business problem identified at the outset of the project. Project developers begin building the software in the staging or development environment using the programming language that was chosen in the previous phase of the project. Also known as building or coding, this stage is where the planning and prototyping come together.

## Coding Checklist

Table 12: Coding Checklist

| **Description** | **Completed** | **If “No” Please Explain** |
| --- | --- | --- |
| Complete: Everything that is in the requirements and design is implemented | No | At this stage in the project 60% of the source code has been written |
| Consistent: No mismatched interfaces and consistent with the design | Yes |  |
| Stylistic: Exhibits good programming style. | Yes |  |
| Modifiable: Code can be modified in keeping with required changes | Yes |  |

## Submit Code

The source code is accessible below, and has also been included with this project’s submission as a zip archive. Additionally, a playable demo has been included which can be played by extracting the Space Guy.zip archive to a convenient directory and launching Space Guy.exe.



# References

*14 Different Programming Languages and Their Uses Explained (in 2019).* (2017, July 5). Mikke Goes Coding. <https://mikkegoes.com/14-programming-languages-explained/#:~:text=14%20Different%20Programming%20Languages%20and%20Their%20Uses%20Explained,Objective-C%209%20Perl%2010%20PHP%20More%20items%20>

*Code Review Checklist*. (n.d.). Www.Codeproject.Com. Retrieved September 23, 2020, from <https://www.codeproject.com/Articles/1156196/Code-Review-Checklist>

*Design Criteria for Programming Languages.* (n.d.). Jcsites.Juniata.Edu. <http://jcsites.juniata.edu/faculty/rhodes/lt/plcriteria.htm>

Dhruv, S. (2019, February 15). *Factors to Consider When Choosing a Programming Language.* Aalpha. <https://www.aalpha.net/blog/factors-to-consider-when-choosing-a-programming-language/>

*OWASP Secure Coding Practices Quick Reference Guide.* (2010). <https://owasp.org/www-pdf-archive/OWASP_SCP_Quick_Reference_Guide_v2.pdf>

*Programming Language Selection.* (n.d.). AcqNotes. Retrieved September 23, 2020, from <http://acqnotes.com/acqnote/careerfields/programming-language-selection>

*TIOBE Index | TIOBE - The Software Quality Company.* (2018). Tiobe.Com. <https://www.tiobe.com/tiobe-index/>

*What Programming Language Should I Learn in 2020? Beginner’s Guide. (2019, February 27).* Mikke Goes Coding. [https://mikkegoes.com/what-programming-language-should-i-learn/#:~:text=%20How%20to%20know%20which%20programming%20language%20to,just%20one%20programming%20language.%20When%20you%E2%80%99re...%20More%20#:~:text=%20How%20to%20know%20which%20programming%20language%20to,just%20one%20programming%20language.%20When%20you%E2%80%99re...%20More](https://mikkegoes.com/what-programming-language-should-i-learn/#:~:text=%20How%20to%20know%20which%20programming%20language%20to,just%20one%20programming%20language.%20When%20you%E2%80%99re...%20More%20)

Wikipedia Contributors. (2020, January 17). *OWASP.* Wikipedia; Wikimedia Foundation. <https://en.wikipedia.org/wiki/OWASP>