|  |  |
| --- | --- |
| Learn Java - Apps on Google Play  Learning Java  PROJECT CHARTER | Abstract  This project aims to teach beginners the basics of the Java coding language. Through the construction of a web application, we hope to provide easy access for beginners to learn the basics of Java programming.  Destiny Deleon, Anthony Ephault, Max Flannick |

Contents

[2 Project Summary 2](#_Toc30145110)

[3 Team 2](#_Toc30145111)

[4 Scope 2](#_Toc30145112)

[4.1 Goals and Objectives 2](#_Toc30145113)

[4.2 Deliverables 2](#_Toc30145114)

[4.3 Stakeholders 2](#_Toc30145115)

[4.4 Out - of - Scope 3](#_Toc30145116)

[4.5 Risks, Constraints, Assumptions 3](#_Toc30145117)

[5 Success Measurements 3](#_Toc30145118)

[6 Signatures 4](#_Toc30145119)

[7 Appendix A – Glossary 5](#_Toc30145120)

[8 NOTES 5](#_Toc30145121)

# Project Summary

*Java is a versatile, object-oriented programming language developed by Sun Microsystems in the mid-1990s. In this project we aim to teach those who do not understand the fundamentals of Java through a fun matching desktop game. Through this game, we will teach the basics of the coding language in the simplest way possible without losing the focus of the one being taught.*

# Team

|  |  |
| --- | --- |
| NAME | ROLE (TEAM LEADER, PROGRAMMER, DESIGNER, ETC) |
| Max | Team Leader (Month 1); Designer (Month 2); Programmer (Month 3) |
| Anthony | Designer (Month 1); Programmer (Month 2); Team Leader (Month 3) |
| Destiny | Programmer (Month 1); Team Leader (Month 2); Designer (Month 3) |

# Scope

## Goals and Objectives

* *Teach students/beginners the course concepts of the Java Programming language*
* *Teach core topics starting with the basics of programming (printing, if statements, etc.)*
* *Move into more complex topics (functions/arguments, classes and constructors, etc.)*
* *Use different study strategies to teach the topics (quizzes, matching game, etc.)*
* *Have an interactive learning for each topic*
* *Organize each topic with a chapter number & section*
* *Have a few animations that keep the user engaged while not distracting them during the learning process*

## Deliverables

* *End Result: A desktop application that will take a user through the Java Programming language starting with basic topics moving to more complex ones.*
  + *This application will be coded through Java and the user interface will be created using JavaFX or another tool within the language*
* *Components within each chapter of the application:*
  + *Information/Reading Portions*
  + *Interactive Learning*
  + *Quiz Style Practice*
  + *Matching or another similar minigame*
* *Chapters in the application include:*
  + *The Basics (printing, variable types, etc.)*
  + *If/Else Statements*
  + *Loops (While, For, Do, etc.)*
  + *Functions, Arguments, and Parameters*
  + *Classes (Constructors, Parent/Child Classes, etc.)*
* *Note: each chapter will have sections within it. For example, Chapter 3: Loops will include a section that goes “Section 1: While Loops” to organize the material better.*
* *Order of Deliverables:*
  + *Phase 1:* 
    - *Interactive title screen presented*
    - *Organize information needed to teach a user each section*
    - *Get information put into a presentable manner*
  + *Phase 2:*
    - *Quizzes for each chapter will be created (~10-15 questions)*
    - *Interactive learning for each chapter will be created (practice questions, images, animations, etc.)*
    - *Everything will be working error free and ready to be put together*
  + *Phase 3:*
    - *Matching and other minigames will be created*
    - *Sound effects will be included*
    - *Application will be tidied up and all put together*
    - *Application complete*

## Stakeholders

|  |  |
| --- | --- |
| **Role** | **Interest/Impact** |
| *Board of Directors (High School)* | *Interest to incorporate a course for their associated high school involving teaching students, especially those with little-to-no knowledge on programming, how to code within the Java programming language.* |
| *Skillshare* | *Companies that present online courses may have a huge interest in this product. This application can help beginner programmers or people wishing to learn something new about the Java programming language. The end result will give the person a brief overview of the language which they can use to start their journey into the technology field or just boost their knowledge about computers.* |

## Out - of - Scope

* *Animations will not be included for every chapter or section as it is not necessary for every chapter and may be distracting.*
* *More complex topics like recursive processes will not be included as this application just covers the basics of the language. However, if there was less of a time-constraint these chapters could easily be added.*
* *There will be no multiplayer minigames included as the application strictly focuses on just a single user. However, the minigames supplied should still give a good amount of entertainment for the user.*
* *There will not be leaderboards for the minigames and it will only include your high score. This is due to a time constraint and a lack of storage in memory.*

## Risks, Constraints, Assumptions

|  |  |
| --- | --- |
| **Risk/Constraint/Assumption Title** | **Explanation** |
| *Time constraints* | *Time constraints may play a role during the duration of the project. Some sections may be longer than others so it will be important to properly organize what is required for each section to complete everything on time.* |
| *Error Processing* | *Error processing is important during the testing of the program. Some errors may be longer to debug than others so we will test each chapter multiple times to make sure everything functions properly before moving on.* |
| *Feedback* | *Feedback from peers will play a role in trying to make this product as informative and intriguing as it can be. This means some parts of the application will be added or dropped depending on the feedback received.* |
| *The user’s computer will be able to handle the application* | *The user’s processor within their computer should be able to handle the execution of the application. However, this application will not demand too much processing power so the user will most likely be able to run the application with no problems.* |

# Success Measurements

* *Create a desktop application to help teach a beginner or student in coding on how to use Java, hopefully leading to a greater understanding of coding languages.*
* *We also wish to do this at a quicker and faster rate than competitors by roughly 10 percent.*
* *We also wish to do this while making chapters less condensed (~5-10 percent) to keep the user engaged while still giving the ability for them to learn each topic and its main components.*

# Signatures

|  |  |  |
| --- | --- | --- |
| **Customer:** |  |  |
| **Name** | **Signature** | **Date** |
| **Chris P. Bacon** | Chris P. Bacon | 1/22/2024 |
| **Matt Cooper** | Matt Cooper | 1/22/2024 |
| **Project Manager:** |  |  |
| **Name** | **Signature** | **Date** |
| **Max Flannick** | Max Flannick | 1/24/2024 |
|  |  |  |
| **Team Members:** |  |  |
| **Name** | **Signature** | **Date** |
| **Anthony Ephault** | Anthony Ephault | 1/24/2024 |
| **Destiny Deleon** | Destiny Deleon | 1/24/2024 |
|  |  |  |

# Appendix A – Glossary

* Java - A high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible.
* Web Application - Software that runs in your web browser
* Function - A block of code used to do a certain task, such as adding two values for example, which could help reduce the amount of lines of code within a program file. It is helpful for dealing with repetitive tasks.
* Arguments - Values passed to a function
* Parameters - Values defined and used within a function
* Recursive Processes - A process used in programming where the a function calls itself within the function
* Class - A common feature within Java used in program files. A class gives the ability to create objects to call functions within the class.
* Parent/Child Class - A type of relationship between two (or more) classes where the child class has its own functions and is able to access functions and data within the parent class as well. The parent class, however, can access functions and data from the child(ren) classes.

# NOTES

* Java is an evolving language but this application focuses on core topics within the language that every beginner needs to know.
* The chapters will increase in length as they go along since they are ordered from least complex, the basics, to the most complex.