|  |
| --- |
| **Contents** |

Introduction………………………………………………………………………………………………………….1

Team Members...............................................................................................................1

Hardware and Software Requirements.…………………………………………………………………….2

Risk Analysis………………………………………………………………………………………………………...2

Work Breakdown..………………………………………………………………………………………………….3

Project Schedule............................................................................................................5

|  |
| --- |
| **Introduction** |

Our client has requested a recreation of the classic game *Pong™* utilizing Python and the Tkinter GUI library. *Pong™* was originally developed in 1972 and was a common staple of early arcades. The game was based on table tennis and functions quite similarly to the sport, in which two players utilize paddles and attempt to hit a ball back and forth to one another without missing. The project will showcase Tkinter’s broad functionality and demonstrate how many products of early technology truly are timeless.

|  |
| --- |
| **Team Members** |

|  |  |
| --- | --- |
| **Role** | **Name** |
| *Project Manager* | Darcy Ralstin |
| *Documentation Specialist* | Calli Church |
| *Programmer* | Brenden Melody |

|  |
| --- |
| **Hardware & Software** |

**Hardware Requirements:**

* Windows Computer
* Monitor
* Keyboard and mouse

**Software Requirements:**

* Windows OS
* Python 3.13.0
  + Tkinter Library

Severity

|  |
| --- |
| **Risk Assessment** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Very Low Severity | Low Severity | Medium Severity | High Severity | Very High Severity |
| Very High Frequency |  |  |  | 4 |  |
| High Frequency |  | 1,2 |  |  |  |
| Medium Frequency |  |  | (4) |  |  |
| Low Frequency |  |  | 3,5,8 | 7 |  |
| Very Low Frequency |  |  |  | 6 |  |
| Extremely Low Frequency |  |  |  |  |  |

Probability

|  |
| --- |
| **Work Breakdown** |

**WBS Chart:**

**1. Project Setup**

* 1.1 Create project folder and file structure
* 1.2 Set up virtual environment (if needed)
* 1.4 Initialize Git repository (version control)

**2. UI Initialization**

* 2.1 Create Tkinter window
* 2.2 Set canvas size and background color
* 2.3 Define game loop setup

**3. Game Objects**

* 3.1 Ball
  + 3.1.1 Create ball shape on canvas
  + 3.1.2 Define ball movement logic
* 3.2 Paddles
  + 3.2.1 Create left and right paddles
  + 3.2.2 Implement paddle movement functions
  + 3.2.3 Bind keys to paddle controls

**4. Game Logic**

* 4.1 Ball movement and bouncing logic
* 4.2 Collision detection
  + 4.2.1 Ball and paddle collisions
  + 4.2.2 Ball and wall collisions
* 4.3 Scoring system
  + 4.3.1 Track player scores
  + 4.3.2 Display scores on screen
  + 4.3.3 Reset ball after score

**5. Game Loop**

* 5.1 Main loop to update ball and paddle positions
* 5.2 Check collisions and update score
* 5.3 Refresh canvas periodically

**6. User Interface Enhancements**

* 6.1 Add score labels
* 6.2 Add start/restart button
* 6.3 Display win/lose messages

**7. Main Menu Window**

* 7.1 Design main menu layout (title, buttons, etc.)
* 7.2 Add navigation to start game, customize character, exit

**8. Character Customization Window**

* 8.1 Design customization UI (colors, paddle shapes, etc.)
* 8.2 Store customization selections
* 8.3 Apply customization to game objects

A graph with green and black text

AI-generated content may be incorrect.

|  |
| --- |
| **Project Schedule** |