Software Requirements Specification: Chrome Dino Runner

SFWRENG 3XA3 Project

Team Number: L03 Group 1
Team Name: "Team Rex"

Chelsea Maramot
maramotc

Anjola Adewale
adewaa1

Sheridan Fong
fongs7

February 11 2022

1 Project Drivers

1.1 The Purpose of the Project

The current pandemic has abruptly disrupted the entertainment sector, and now people are seeking entertainment within the comfort of their homes. An example of an at-home entertainment solution is video games. We aim to provide home entertainment by redesigning the classic Chrome T-Rex dinosaur game. We plan to improve its user-friendliness and interactivity while maintaining the game's basic functionality. The game's development will follow the software development process and be implemented in python using the PyGame library. The entire process will be documented and tested using the unit test framework.

1.2 The Stakeholders

1.2.1 The Client

The clients for this project are the course instructor of SFWRENG 3XA3, Dr. Ashgar Bokhari, and the teaching assistants (TAs), Stephanie Koehl, Veersah Palanichamy and Abdul Rab Mohammed. The clients will provide project requirements, deliverables and deadlines. They will also provide guidance when necessary and evaluate the project with respect to the requirements in the SRS document.

1.2.2 The Customers

The customers for this project are individuals who are interested in playing Chrome-Dino. The project does not explicitly target a demographic but is rather designed as a general-purpose entertainment source. The project is designed for anyone with the game's required software, such as Python and the PyGame library.

1.2.3 Other Stakeholders

All members of group 1 are stakeholders of this project. We are all responsible for the development process, such as implementing, testing, and documenting the project. Group 1 members all care for the project's success and are responsible for maintaining the repository. Developers that fork the repository and continue the project's development are also stakeholders as they will be continuing the project and are interested in the project's success

2 Project Constraints

are restrictions on the product due to the budget or the time available to build the product

2.1 Mandated Constraints

2.1.1 Solution Design Constraints

Description: The game (an executable file) must be able to run on any machine running on Windows 7 or newer, macOS 10.12 or newer or Linux Ubuntu or newer Rationale: Most computer users already use systems with these specification and so the users will not need to purchase a new system. Fit criterion: The game will be developed into an executable file that will be made to run on Windows or newer, macOS Sierra 10.12 or newer or Linux Ubuntu or newer

2.1.2 Implementation Environment of the Current System

N/A

2.1.3 Partner of a Collaborative Application

N/A

2.1.4 Off-the-Shelf Software

N/A

2.1.5 Anticipated Workable Environment

N/A

2.1.6 Schedule Constraints

Description: This project must follow the project schedule outlined in the Gantt chart and the Task Section. Rationale: This is because, the project must be completed prior to the end of the course a strict pre-defined plan must be followed. This is also so to ensure that all deliverable are submitted by their due dates Fit Criterion: All deliverable contained within this project must completed and submitted by April 12, 2022. Due to the time constraints on this project(3 months), this project needs to follow a predefined plan in order to ensure it's completion in an organized manner.

2.1.7 Budget Constraints

N/A

2.2 Naming Conventions and Terminology

2.3 Relevant Facts and Assumptions

2.3.1 Facts

The original repository contains 350 lines of code and is developed in Python using the Pygame library.

Table 1: Table of Naming Conventions and Terminology.

Term	Definition
Chrome Dino Runner	for all xy iht if $xy \in R$ and $yx \in R$, then $x = y$.
High Score	Highest score achieved by the player
Pygame	a cross-platform set of Python modules designed for writing video games.
Python	The programming language used in developing Chrome Dino Runner .
Score	A numerical value which quantifies the player's performance the game.
SRS	Acronym for Software Requirements Specification; A document that describes what the system will do and the expected performance.
User	The Individual who will be playing our game.
WASD keys	Four keyboard keys that are used to interact with video games in lieu of Arrow keys or a controller. W and S control forward and backward movement, while A and D are left and right
Basic English Proficiency	R Knowledge of vocabulary words, ability to speak simple phrases or sentences

2.3.2 Assumptions

Users possess a computer running Windows 7 or newer, macOS 10.12 or newer or Linux Ubuntu or newer User have basic English proficiency Users know how to operate a PC Users have the visual and physical capabilities to play the game

3 Functional Requirements

3.1 The Scope of the Work and the Product

3.1.1 The Context of the Work

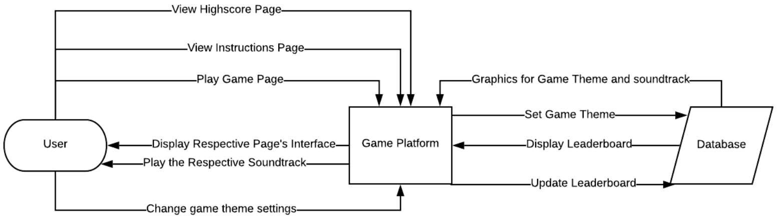

Figure 1: Caption

3.1.2 Work Partitioning

Table 2: Work Partitioning Event Details

ID	Event Name	Event Description	Input	Expected Output
1	Viewing the Instructions Page	Displays the instruction page.	Keyboard/Mouse	Instructions page
2	Viewing the Leaderboard	Displays the leaderboard	Mouse	Leaderboard page
3	Change Game Settings	The user can select different themes for the game, changes will be reflected in the UI. They can also alter the audio output.	Theme name and sound settings	Interface display and audio
4	Playing the Game	The user plays the dino game	Keyboard/Mouse	Interface display, theme music and final display

3.1.3 Individual Product Use Cases

The use case diagram above shows the different ways a user can interact with the system. The user can start a game and the game must include displaying the score and allowing the user to jump over obstacle and displaying the final score. Game play is not impacted on whether a user goes to the main menu or if the score is saved to the database. Therefore, these action extend the use case. The use cases for viewing the instructions page, leaderboard page and settings menu are all self-explanatory.

3.2 Functional Requirements

E1: Viewing the Instructions Page
FR. The system must display instructions on how to play the game to the user.
FR. The system must provide the user with a way to navigate the main menu.

E2: Viewing the Leaderboard

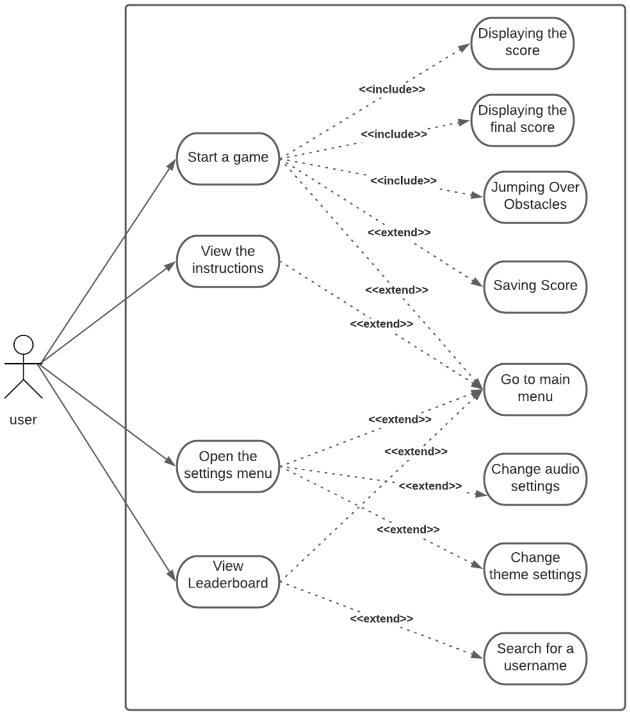

Figure 2: A use case diagram that displays the functionality of the application.

FR. The system must display the user with a list of the players with top scores and the respective score.
FR. The system must provide the user with a way to navigate back to the main menu.
FR. The system must be able to filter the high scores by player.

E3: Change Game Settings

FR. The system must provide the user with a way to exit from the game settings to the main menu.

FR. The system must allow the user to change the audio settings on the game. Such as having no audio or audio.

FR. The system must display the available themes and allow the user to select a theme.

E4: Playing the Game

FR. The system must display the game to the screen with the appropriate theme.

FR. The system must initialize the score to 0 when starting the game.

FR. The system must output sound based on the audio settings.

FR. The system must start gameplay when the user inputs a keyboard command.

FR. The system must allow keyboard inputs to control the user interface.

FR. The system must award points to the user after each dino step.

FR. The system must display the user's current score to the screen.

FR. The system must allow the user to replay the game.

FR. The system must allow the user to stop the game and return to the main menu.

FR. The system must allow the user to enter their username when they are done playing the game.

FR. After the game is finished, the user score must be entered into the database.

4 Non-Functional Requirements

4.1 Look and Feel Requirements

4.1.1 Appearance Requirements

LF1. The user interface consists of essential components relevant to the game.

4.1.2 Style Requirements

LF2. The product shall maintain the 90's arcade appearance.
LF3. The product shall incorporate different themes.
LF4. The game colours must not distract the user's game play.

4.2 Usability and Humanity Requirements

4.2.1 Ease-Of-Use Requirements

UH1. Game shall be controlled using two methods: arrow keys or WASD.
UH2. The game shall have a simple menu page where the game can be accessed easily.
UH3. The game shall be used by people intuitively, no training needed and at a minimum basic English level.

4.2.2 Personalization and Internationalization Requirements

UH4. The product shall only be used in English.
UH5. The user can adjust the theme of the game based on their preferences.

4.2.3 Learning Requirements

UH6. The user shall not require training before playing the game.
UH7. The game must contain basic instructions within the main menu page.

4.2.4 Understandability and Politeness Requirements

UH8. The game shall encompass a level of abstraction from the user.
UH9. The game shall use common control keys to play the game.
UH10. The game will include universal symbols and words that are naturally understood by the user community.

4.2.5 Accessibility Requirements

UH11. The game shall be playable for users with colour blindness.
UH12. The game shall use appropriate font style and size, making it readable to the general public.

4.3 Performance Requirements

4.3.1 Speed and Latency Requirements

PE1. Scores shall be uploaded to the leader board in less than 10 seconds.
PE2. The interface must have a maximum response time of 2 seconds to avoid disrupting the user's game play.

PE3. The game shall update the new status parameters within 5 seconds of user input.

4.3.2 Safety-Critical Requirements

N/A

4.3.3 Precision or Accuracy Requirements

PE4. Integer whole number scores shall be uploaded and displayed on the screen within 100 milliseconds of changing.
PE5. The Leader board shall be updated according to the top integer score.
PE6. The game shall have a maximum latency of 100 milliseconds.

4.3.4 Reliability and Availability Requirements

N/A

4.3.5 Robustness or Fault-Tolerance Requirements

N/A

4.3.6 Capacity Requirements

PE7. The game shall only be available to a single player at a time.
PE8. The game shall allow a maximum of 5 users stored in the leader board.

4.3.7 Scalability or Extensibility Requirements

PE9. Developers shall be able to add new features, such as themes, without compromising the basic functionalities of the game.

4.3.8 Longevity Requirements

PE10. The game shall maintain functionality with existing software until Spring 2022.

4.4 Operational and Environmental Requirements

4.4.1 Expected Physical Environment

PE11. The game shall be played with full functionality without an internet connection.
PE12. The game shall be played within any computer operating system (i.e. Linux, Windows).

4.5 Requirements for Interfacing with Adjacent Systems

4.5.1 Productization Requirements

PE13. The game shall be distributed to user computers as a .EXE file.
PE14. The game shall be installed without the use of separately printed instructions.

4.6 Release Requirements

PE15. The game shall be released by April 5, 2021.

4.7 Maintainability and Support Requirements

4.7.1 Maintenance Requirements

MA1. Source code shall be updated with the latest changes, within one working week of the scheduled agreement date.
MA2. Source code shall be appropriately documented.

4.7.2 Supportability Requirements

MA3. The full project source code shall be made available to Git users in order to raise issues.

4.7.3 Adaptability Requirements

MA4. The product shall run under Windows 10, Linux Ubuntu 16.04, or newer versions of these operating systems.

4.8 Security Requirements

4.8.1 Access Requirements

SR1. The game users shall have read-only access to the leader board and user scores.
SR2. Users are permitted to change game settings through user interface.

4.8.2 Integrity Requirements

SR3. The product shall protect itself from unauthorized user modifications.
SR4. The user shall not be allowed to modify scores.

4.8.3 Privacy Requirements

SR5. Each user shall only be able to view other user names and their corresponding high scores within the leader board.

4.8.4 Audit Requirements

N/A

4.8.5 Immunity Requirements

SR6. The product must not be vulnerable to unauthorized or undesirable software programs.

4.9 Cultural and Political Requirements

4.9.1 Cultural Requirements

CP1. The game themes shall not be offensive to any cultures or religions.
CP2. The game shall not allow users to input offensive English user names or from other languages.

4.9.2 Political Requirements

N/A

4.10 Legal Requirements

4.10.1 Compliance Requirements

N/A

4.10.2 Standards Requirements

N/A

4.11 Health and Safety Requirements

N/A

5 Project Issues

5.1 Open Issues

There are currently no open Issues in the Chrome Dino Runner repository. However we found that the required pygame version 2.0.1 does not exists but the most current version of pygame seems to work fine with the code. In addition, the code writes into a score.txt file that does not exist in the initial code repository.

5.2 Off-the-Shelf Solutions

A version of the game we are reimplementing is available on Chrome's offline mode and multiple websites including https://chromedino.com/, https://dino-chrome.com/en and more Finally, during the 2021 Olympic Games in Tokyo Google released a limited edition themed version of the Dino game which is no longer available https://chromeunboxed.com/chromes-dino-game-tokyo-olympic-theme/

5.3 New Problems

This new implementation of the game will not change the basic rules of the game. Players who are accustomed to the orginal version of the game should be able to play the our version without the need for instructions

5.4 Tasks

Tasks are scheduled and delegated as per the project blueGantt Chart.

5.5 Migration to the New Product

FILL IN WITH GROUP

5.6 Risks

This project has minimal risk as it is designed for entertainment purposes. The significant risk of this project will be in terms of testing. The quality of our game will depend on the graphical user interface. Testing GUI's poses challenges as the only testing method available would be system testing. Compared to unit testing, system testing increases the change of bugs and issues going undiscovered. Our game must also be compatible with different hardware configurations and on multiple operating systems.

5.7 Costs

This project does not have any costs associated with it as it is being developed with open-source software and resources. The only cost is time resources; these resources will be used for the development, documentation and testing of the project.

5.8 User Documentation and Training

5.8.1 Documentation

An instructions page will be available in the main menu of the game. This page will contain instruction on how the game is to be played as well as the keys that can be used as controls. Information regarding scoring will also be present in this page. Finally, the main repository of the game will include a README file, which will detail and installation and set-up instructions.

5.8.2 Training

A user can play the **Chrome Dino Runner** with no necessary training. The game's controls (up and down arrow) should be intuitive when jumping over obstacles. The player may practice the game to improve skill, but no prior training is needed to succeed.

5.9 Waiting Room

- Adding background music to the game to match the speed of the dinosaur

- Developing a themed version of the game
  - For example: McMaster student theme
- Inserting a play again button
- Creating additional pages such as a home and instruction page
- Implementing new button control and keyboard binding such as A,W,S,D keys
- Changing module design and incorporating software engineering principles

5.10 Ideas for Solutions

Manually create and populate the score.txt file to successfully compile and run the game.