

1. Design Documentation	2
1.1 Design Documentation Introduction	3
1.2 Class Diagrams	4
1.3 User Interface Mockup	6
1.4 File Formats	13
1.5 Development Environment	14
1.6 Patterns	15
1.7 Design Documentation Summary	16

Design Documentation

Formula Fast

Version History

Version	Date	Author(s)	Summary of Changes
0.1	12 Feb 2024	Ashton David Ryan Franklin	Created design documentation pages.
0.2	15 Feb 2024	Gabriel Azhari	Reordered the design documentation pages.
0.3	29 Feb 2024	Gabriel Azhari	Added the overview, objectives, and references information to the Design Documentation Introduction page.
0.4	01 Mar 2024	Gabriel Azhari	Added a reference to the design documentation introduction.
0.5	03 Mar 2024	Gabriel Azhari	Completed the development environment page.
0.6	03 Mar 2024	Que Hung Dang	Completed pattern design research to use for Formula Fast development.
0.7	03 Mar 2024	Ashton David Ryan Franklin	Updated the file format page under design documentation.
0.8	03 Mar 2024	Gabriel Azhari	Added information about our Bitbucket Git repository to the development environment page.
0.9	03 Mar 2024	Ashton David Ryan Franklin	Added class diagram to the confluence.
1.0	04 Mar 2024	Mahmoud Sherif Mohamed Radwan	Added UI mockup for login page and main menu.
1.1	04 Mar 2024	Matthew Marbina	Added UI mockups for gameplay and game completion screens.
1.2	04 Mar 2024	Matthew Marbina	Completed and uploaded design documentation summary.
1.3	04 Mar 2024	Ashton David Ryan Franklin	Uploaded UI mockups for leaderboard and instructor mode screens.
1.4	04 Mar 2024	Gabriel Azhari	Added two definitions to the design documentation summary.
1.5	04 Mar 2024	Que Hung Dang	Added reference to patterns
1.6	04 Mar 2024	Mahmoud Sherif Mohamed Radwan	Edited UI mockup for the login page to include team info
1.7	04 Mar 2024	Que Hung Dang	Added tutorial page

Design Documentation Introduction

Overview:

Education is crucial in understanding the complex world today, and many people may find it hard to have any enthusiasm when it comes to learning. However, educational games successfully blend the excitement of gaming with the benefits of learning. This is proven by studies that show that gamification in education can lead to higher levels of motivation, increased attention spans, and stronger retention of the learned material compared to conventional teaching methods.

Taking this into account, we aim to create an educational game that manages to enhance the learning experience for all. The software we create will be called "Formula Fast". Formula Fast will be a racing game where the user races against other cars that are computer-controlled. The user can only move their car forward by answering math equations and the user will have to cross the finish line before the timer countdown reaches 0. Also, the equations will become progressively more and more difficult as the user completes various levels. To achieve this we will be using Java as our coding language and VS Code as our IDE. We will also be using JavaFX and Scene Builder for the GUI.

Specifically, for this design portion of the project we will be using a class diagram to illustrate the initial design concept for our project, paired with a textual description of the diagram. Furthermore, we will be creating an interactive mockup of the user interface for our software using Balsamiq, paired with a textual description of each wireframe. We will also cover how we will be storing data involved in the project (CSV files), the aforementioned development environment we will be using, and what patterns we will be using in regards to our software.

Objectives:

The objectives of the project and software are to:

- apply the principles of software engineering towards a real-world problem
- work with, interpret, and follow a detailed specification
- create models of requirements and design from the given specification
- implement our design in Java and deal with decisions made earlier in the design process
- create graphical, user-facing content and applications
- write robust and efficient code
- write good, clean, well-documented Java code that adheres to best practices
- reflect on good/bad design decisions made over the course of the project

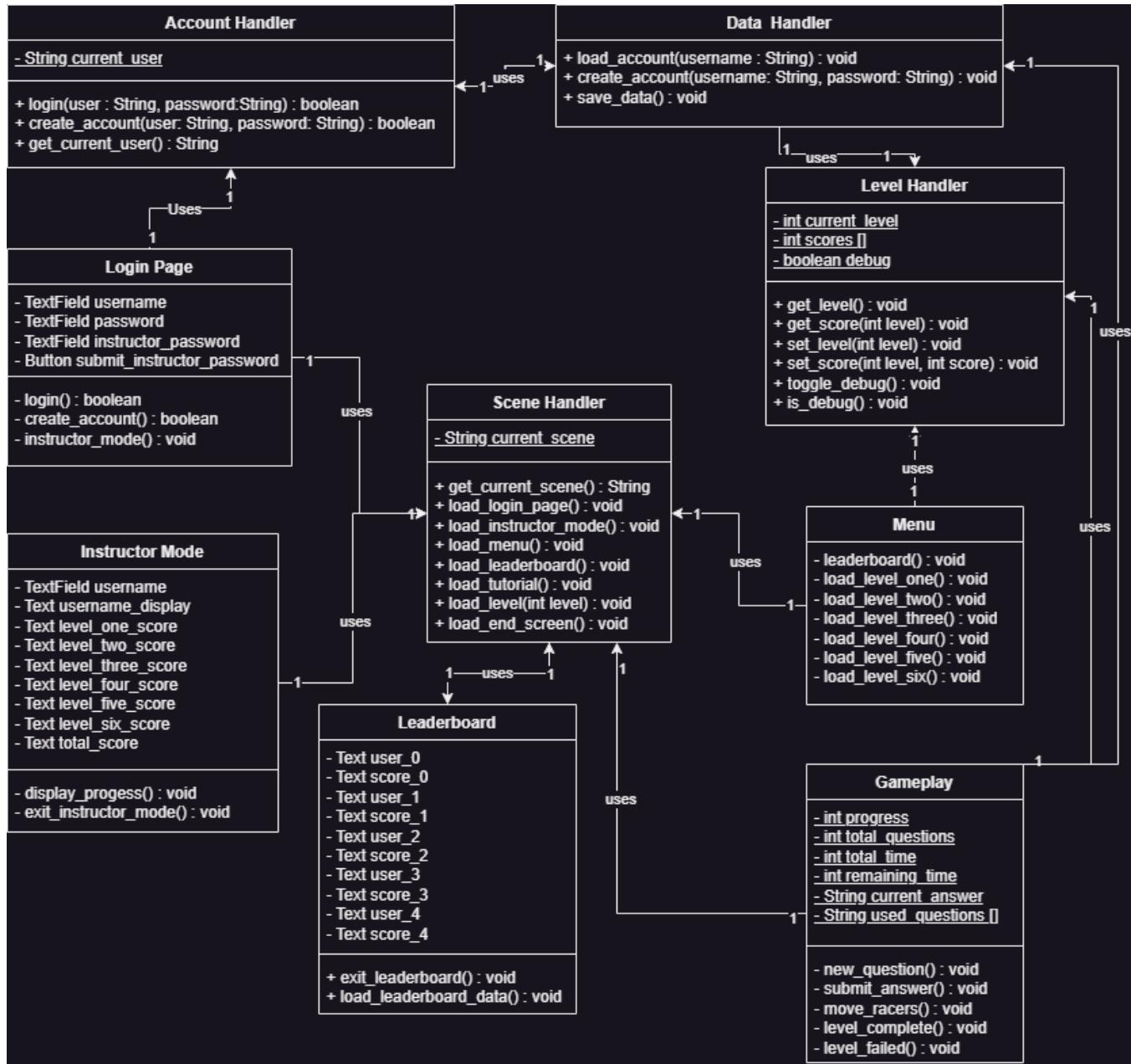
With regards to educational objectives, the main objective of Formula Fast is to create a game that helps teach arithmetic to young children in elementary school while giving them an enjoyable and memorable experience. Specifically, Formula Fast will help the player learn addition, subtraction, multiplication, and division by starting with easy equations and progressively making the given equations more and more difficult as the user completes levels and progresses through the game.

References:

CS2212 Group Project Specification. UWO OWL. Version 1.0. <https://owl.uwo.ca/access/content/group/aa2311c9-4cef-497f-8d9c-b502023be21c/project/CS2212%20Group%20Project%20Specification.pdf>. Accessed 15 Jan. 2024.

Formula Fast Requirements Documentation. Version 2.2. <https://wiki.csd.uwo.ca/display/COMPSCI2212W2024GROUP65/Formula+Fast?src=contextnavpagetreemode>. Accessed 29 Feb. 2024.

Class Diagrams



Account Handler: This class will handle anything to do with the users account, including logging into an account, creating a new account, or checking which user is currently logged into the system.

Data Handler: This class will handle interactions with the data csv file, such as loading a users data to the level handler, saving data to the csv file from the level handler, and creating new accounts in the data file.

Level Handler: This class will contain all information regarding levels for the currently logged in user, such as their currently unlocked level, their scores for each level, and methods for debugging.

Login Page: The login page class will be the controller for the FXML file displaying the login screen. It will have methods to handle the different button interactions available through the UI.

Instructor Mode: This class will be the controller for the FXML display of the instructor mode UI, it contains methods to display progress for a searched for student and interaction methods for the UI buttons.

Scene Handler: This class will handle all JavaFX scene changes, this will control the changing between all different screens, i.e. menu to gameplay, login to menu, etc...

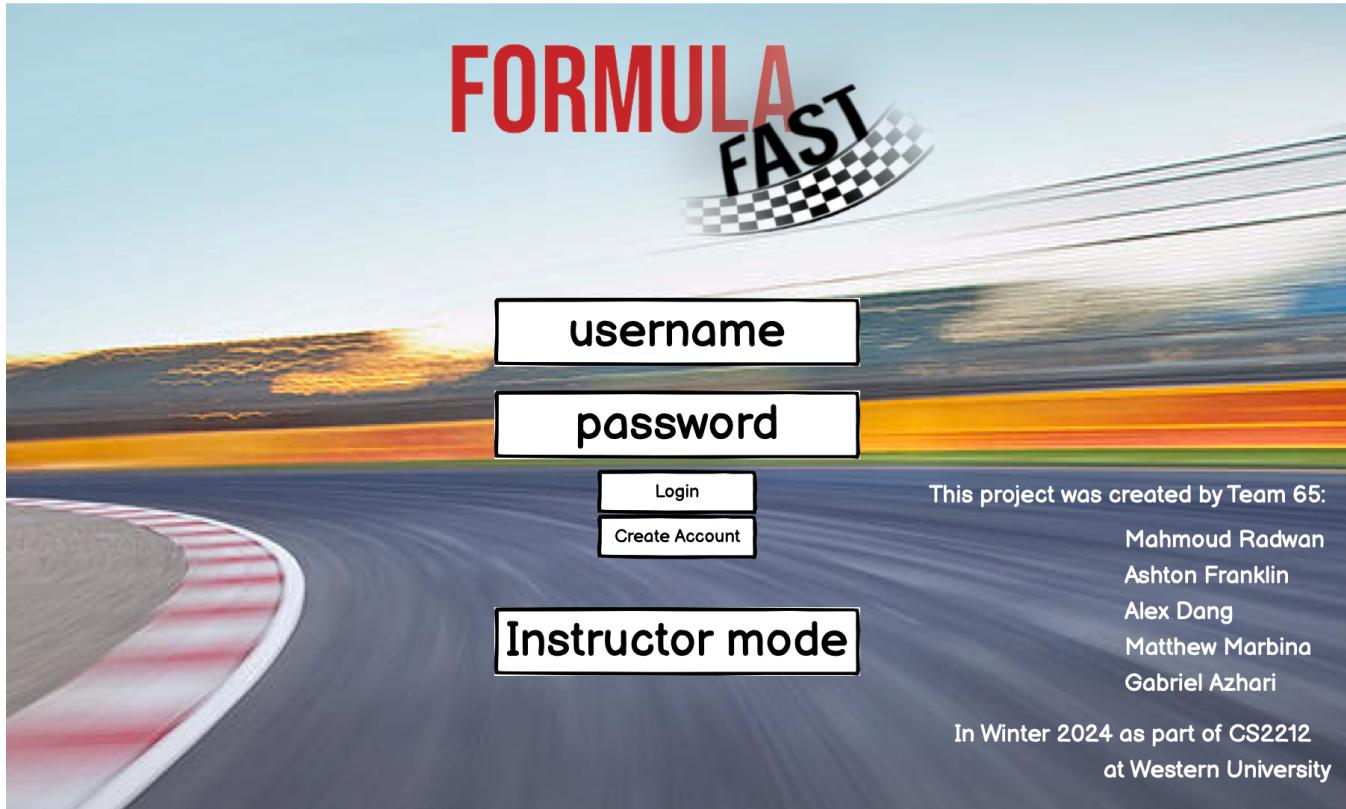
Leaderboard: This class will be the controller for the FXML display of the leaderboard page. The class will contain methods to load the top 5 users onto the UI as well as exit the leaderboard via UI interactions.

Gameplay: This class will handle the actual gameplay when a player enters a level, this includes generating questions, checking answer submissions, moving ai cars, etc...

Menu: This class will be the controller for the FXML display of the main menu, it will contain methods for all UI interaction within the main menu.

User Interface Mockup

Login Page

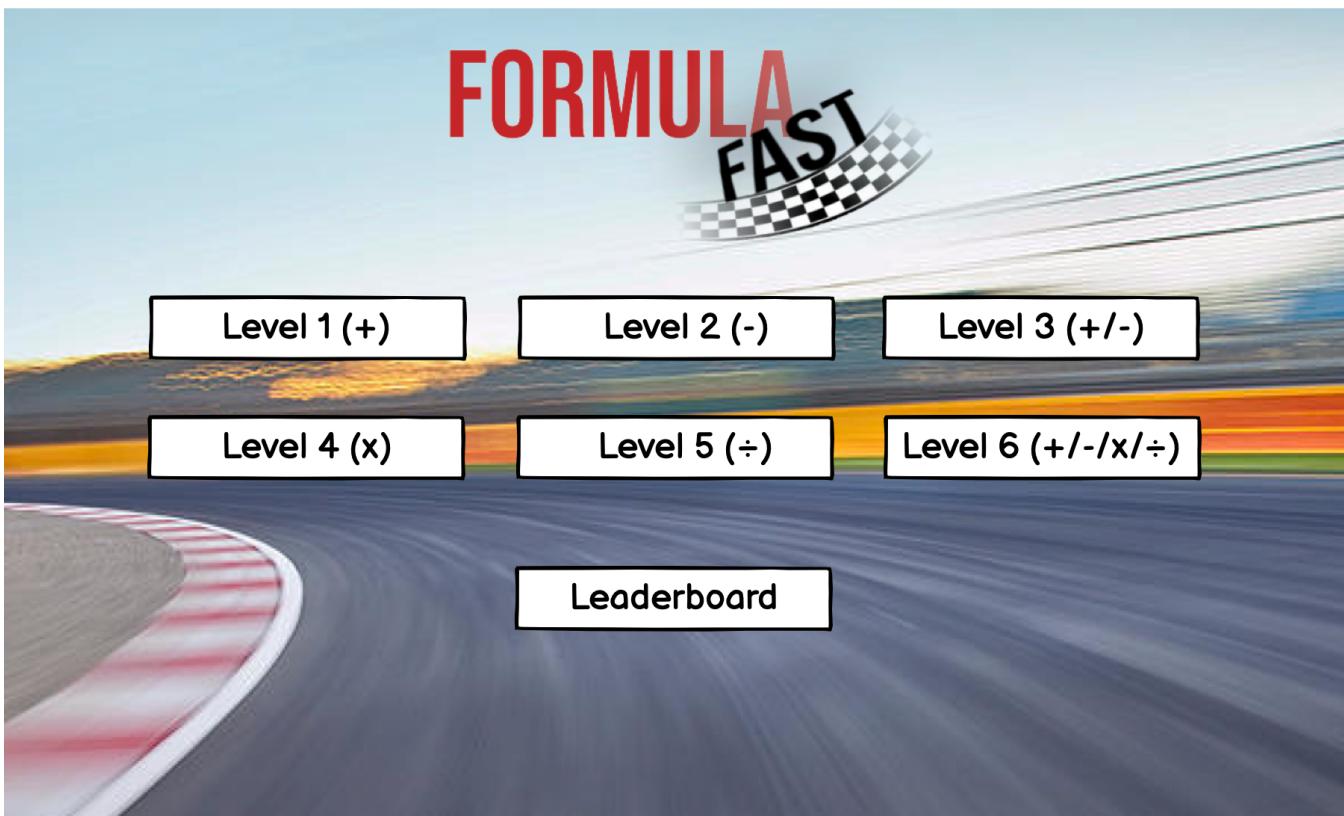


Link to Source File: [FF-loginpage.png](#)

Textual Description:

- For the Logic Page, we decided to keep it simple with a plain background image that is not too busy nor distracting. We included two big text boxes for username, password, and instructor mode since they are the focal point of this page. After the user inputs their details, they can either login to their existing account or create a new account using their respective buttons. Once the player creates their account, their screen automatically changes to the tutorial screen, where users learn the basics of the game before playing the first level with an interactive tutorial. We chose to make these buttons smaller than the rest as to draw the user's attention to the action of logging in. Our logo "Formula Fast" is placed at the top since it is the name of our game and we want to make it visible to the user as to make it more recognizable to them.

Main Menu

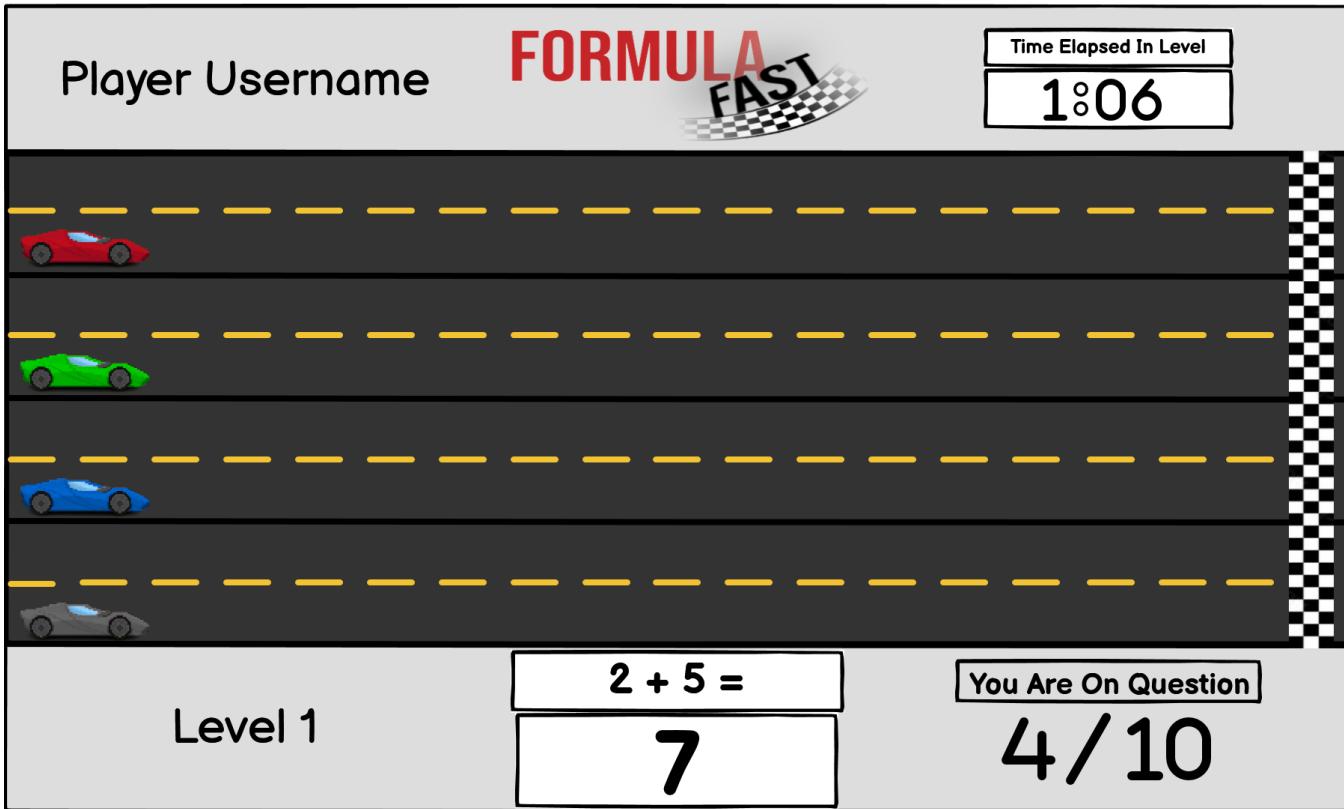


Link to Source File: [FF-mainmenu.png](#)

Textual Description:

- For the Main Menu, we still kept it simple as we want this UI to be user friendly and easy to use. The background stays the same as to make the switches from page to page more fluid (preventing the user from being "flashed" with multiple different background images after every button click). We present each level along with what the level will consist of -(+) for addition, (-) for subtraction, etc. We also placed the leaderboard in the centre and spaced it further down than the levels as to prevent the user from misclicking and to further show the difference of this button's function in a subtle and user friendly manner.

Gameplay Screen

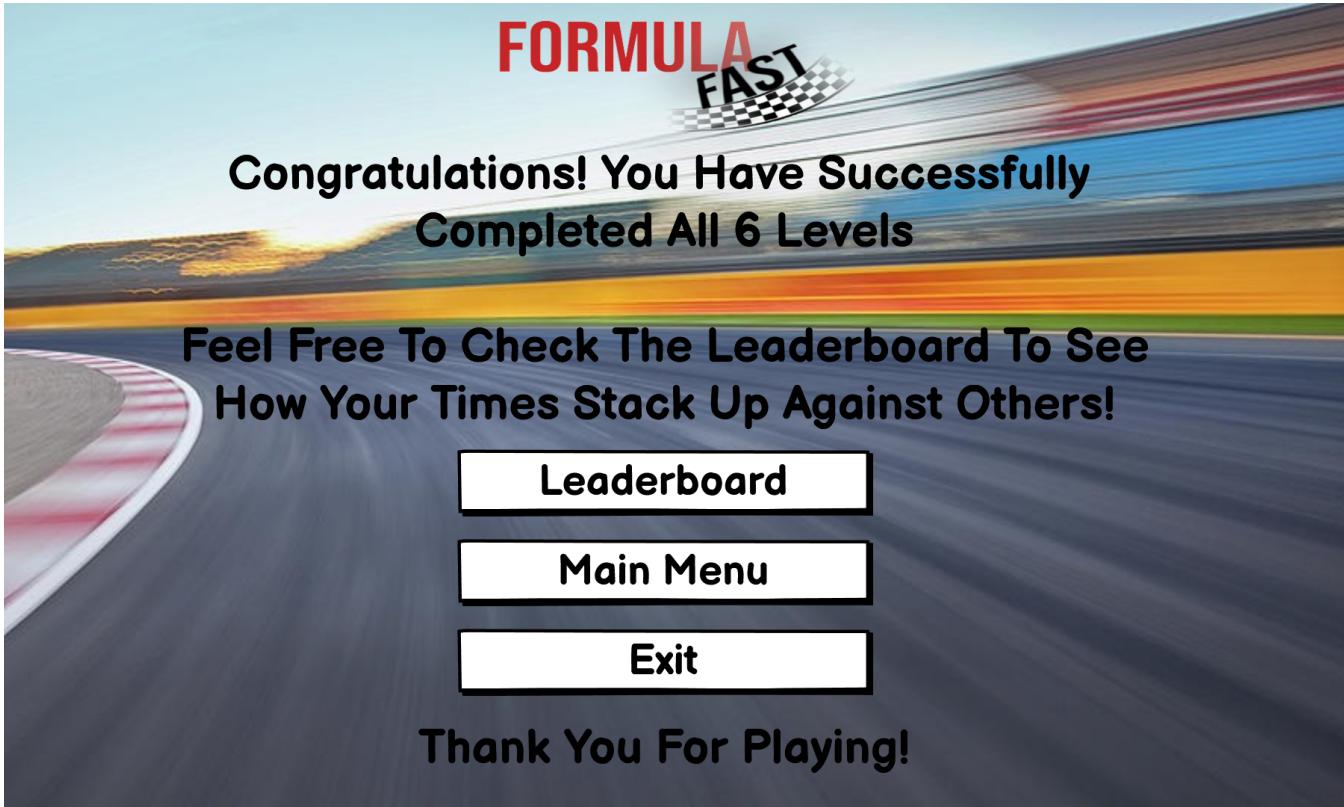


Link To Source File: [Formula Fast Gameplay UI Mockup](#)

Textual Description:

- For the Gameplay Screen, we decided to include each of the four cars in their own separate track to avoid the screen looking cluttered and keep the spacing effective. We decided it would be smart to include the time elapsed in the level on the screen so the player would know how fast they complete the level, as that is how we calculate their score. We also thought including additional information, such as the current level, the question the player is on, and the player's username, would help fill out the whitespace and make it look more like a normal game.

Game Completion Screen

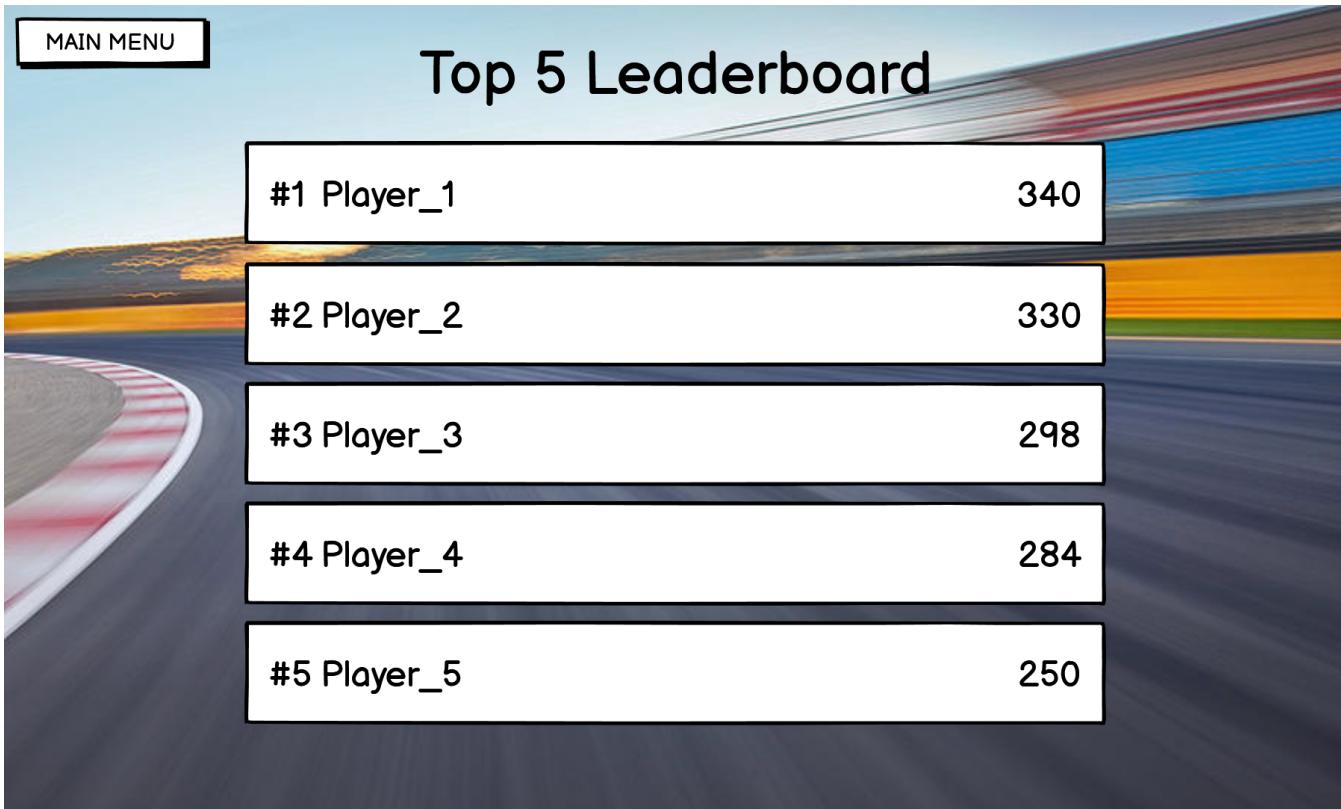


Link To Source File: [Formula Fast Game Completion Screen UI Mockup](#)

Textual Description:

- For the Game Completion Screen, we decided to keep it simple, using the same background image used in the main menu, including simple text boxes congratulating the player, recommending checking the leaderboard, and thanking them for playing. We also added three buttons, one taking players to the leaderboard, one taking players back to the main menu, and one exiting players from the game

Leaderboard

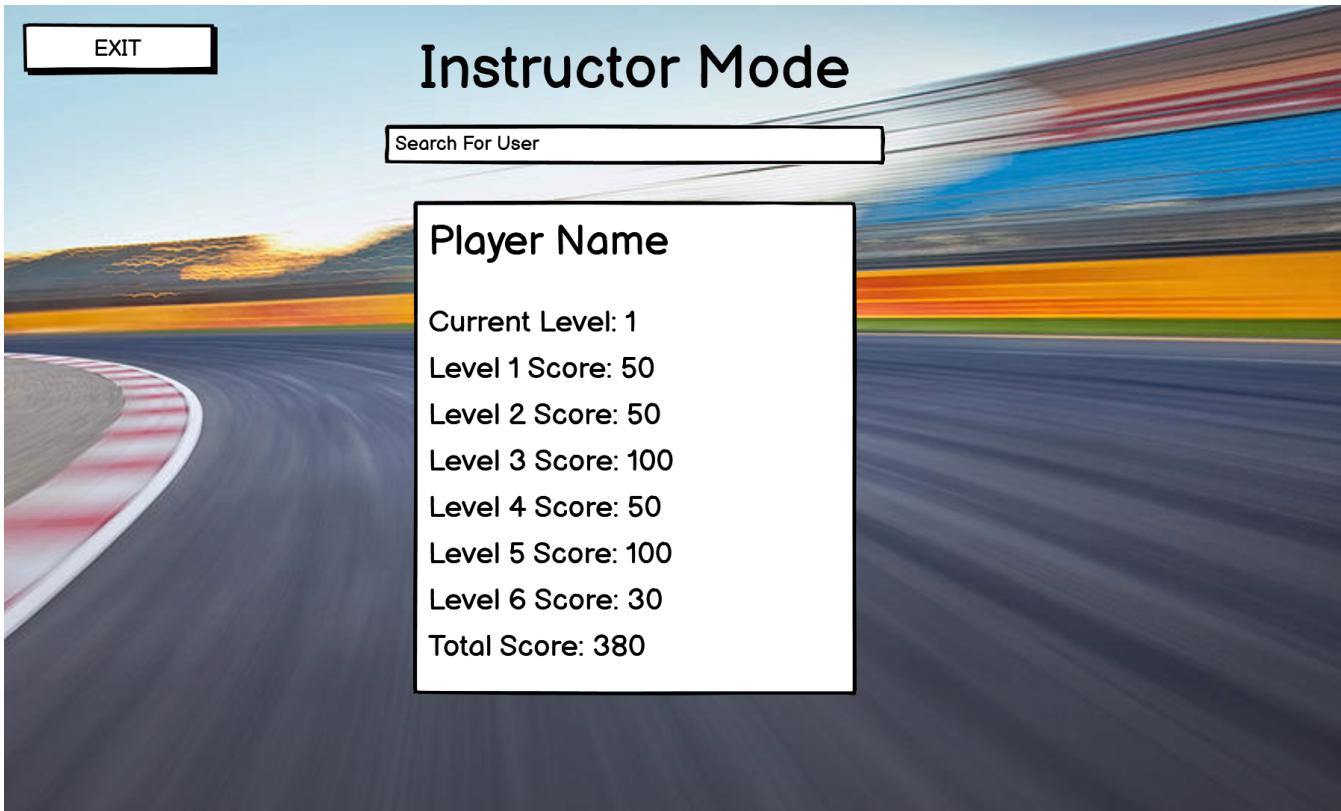


Link To Source File: [Leaderboard.png](#)

Textual Description:

- The leaderboard above displays the overall scores for the top 5 users, displaying their ranking, username, and score. The leaderboard also features a "MAIN MENU" button to return to the menu from the leaderboard.

Instructor Mode

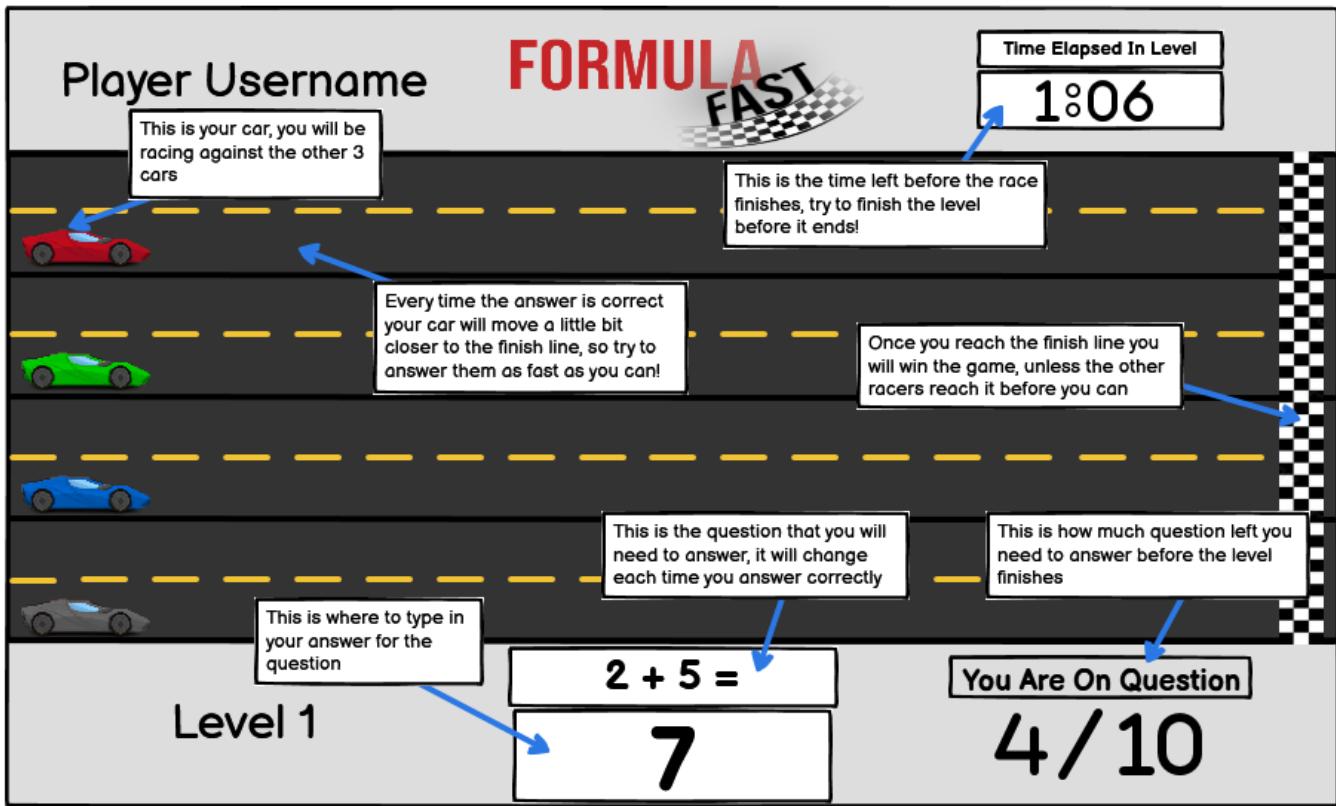


Link To Source File: [InstructorMode.png](#)

Textual Description:

- The instructor mode features a search bar just under the title which can be used to search for any player in the database. The large center box will display the name of the user being displayed as well as their current level and their scores for each level individually and their total score.

Tutorial Page



Link To Source File: [Formula Fast tutorial page.png](#)

Textual Description:

- The Formula Fast tutorial page will teach students how everything works and how to play the game. Each arrow will point to different part of the game interface and the text box will explain the information in each area to the students so they will know how to play the game.

File Formats

All data will be saved using simple CSV files. The software will support up to 10 individual users on one device. The CSV will split the different users data by rows, i.e. each row will represent an individual user.

Each row will be divided into columns, the first column will be the username and is what will be used to identify which data belongs to who, as well as displaying names alongside high scores in the leaderboard.

The second column will contain the users password used when logging into their account, this will not be encrypted in anyway, simply store as plain text.

The third column will be the "current level" this displayed the users highest unlocked level, NOT they highest level they have completed, if current level is 3 that means the player has completed the first 2 levels and next has to complete level 3. In the case that a player has completed all levels, this section will simply read "COMPLETE" signaling that all levels should be unlocked for this player.

The following columns will display the high scores for each level in ascending order of level. The score for a level will be calculated by $(MAX_TIME - (MAX_TIME - TIME_REMAINING))$, i.e. if you have 120 seconds to complete a level, and take 30, you will have a score of 90 for that level. Any incomplete levels will be given a score of 0.

When a player creates a new account their current level will be set to 1 and scores for all levels will be set to 0. When a player has completed all levels, current_level will be set to num_levels + 1, i.e. if there are 6 levels it will set to 7.

Below is an example of a csv file with three used player slots:

```
user, password, current_level, score_1, score_2, score_3, score_4, score_5, score_6
```

```
john,my_password,1,0,0,0,0,0  
bob,password123,3,60,80,0,0,0,0  
jill,cool_password,7,60,80,30,40,50,80
```

Development Environment

The development environment is what IDE, tools, and external libraries we will be using to develop our software. The IDE we will be using is VS Code, and our software will be a Java desktop application. We will be using Javadoc to document our code, and JUnit to test our software. For the GUI we will be using JavaFX and Scene Builder. Also, we will be using Balsamiq to create interactive mockups of the user interface, and CSV files to store any necessary data. Additionally, we will be using a Bitbucket Git repository to store all code and files for our project.

Patterns

Level Design Pattern

The Level Design Pattern will help create efficient progress for the students. Using six different levels, Formula Fast will be an engaging educational game that encourages students to be more active. Using this pattern will increase the game's difficulty, which prevents the students from feeling overwhelmed when playing the game. With each level, the method of calculation will become harder to ensure that students are effectively learning. The pattern will also create a structured learning experience, allowing students to be proficient in one area before moving on to other mathematical sections. In addition, each level will provide immediate feedback to help players track their progress and know where to improve their skills. While players can have difficulty completing a level, they will be encouraged to practice their skills to thoroughly understand the materials before moving on to more complicated problems.

Fixed Reward Design Pattern

By giving the students scores based on how fast they complete the questions, the learning process will be more rewarding, and players will work on improving their speed in calculating problems. The pattern also aligns with our game philosophy, a racing game where students aim to be the first to cross the finish line. When players get better scores, they can see their improvement in real time, motivating them to improve their mathematical skills. The fixed reward system will also help instructors track their overall class performance. We can also seamlessly integrate with the level design pattern to provide level-based performance for instructors to determine which sections they should focus more on during lecture time. In conclusion, using the fixed reward design, students will feel encouraged to learn and create an authentic racing experience, which the game aims to deliver to its players.

Composite Pattern

The composite pattern will allow us to unify individual math questions and six different math levels. The game's structure will manage single math questions and different levels through the same interface, simplifying the game logic. Since this is an open-source game, implementing this pattern will allow the game to be easily modified or expanded without creating difficulty for the game's flow. We can maximize code reusability with a scalable framework to ensure clean coding practices for formulas fast. By treating the game element as a tree-like hierarchy, we can insert and adjust math problems without negatively impacting the overall game's structure. Formula Fast will be able to use the same methods for both simple and complex game parts which help us develop the game more organized.

Navigation Tab Pattern

We will implement this navigation pattern in our main menu to help users easily navigate our game. The navigation will help user to create their users or log into their users and also let instructor to go to their portal to check their class's performance. After logging into their account, the navigation pattern will also be used to display the six levels for players, which make organizing the user interface more seamless. Players will be easily switch between their levels using the navigation level menu which reduces excessive interaction with the game.

Reference:

<https://ui-patterns.com/patterns/Levels>

<https://ui-patterns.com/patterns/Fixed-rewards>

https://www.tutorialspoint.com/design_pattern/composite_pattern.htm

<https://ui-patterns.com/patterns/NavigationTabs>

Design Documentation Summary

In conclusion, the goal of this documentation was to leverage the group project by producing documentation that captures several key aspects of the design for the software system of our game, Formula Fast. These key aspects include a [class diagram](#), that provides a visual representation of the software's structure, which gives us a clear idea of how to build and code each part of our game, several [user interface mockups](#), that showcase proposed designs of how the layout of each screen of our game will look, a description of the [file formats](#) used in the software, that describe how we will be storing data involved in our software, our [development environment](#), that describes the programming language and the several programming tools we will be using throughout our software, and lastly, the [design patterns](#) used to solve common design problems in our software. Overall, our design documentation will serve as a strong guide that we will follow to efficiently implement and design our game's software.

Gamification: The process of adding game-like elements (point system, competition with others) that encourage participation.

User Interface (UI): The visual elements that a potential user would interact with (buttons, icons)

CSV File: A type of plain text file that uses commas to separate values, where each value represents one data record.

Encrypted: Process of converted information/data into a secret code to prevent unauthorized access

External Library: A custom set of pre-written Java functions/objects used to eliminate having to write code.

Open-Source Game: A video game whose code used to create the game is publicly available

Scalable Framework: A set of principles and practices that allow a system to efficiently handle growth in workload

Tree-Like Hierarchy: A way of representing a structure in the software in a graphical form, using the hierarchy of each level/aspect of the structure.

IDE - An Integrated Development Environment is a software application that helps programmers develop software code efficiently

GUI - A Graphical User Interface is a digital interface in which a user interacts with graphical components (Ex. buttons, menus, etc.)