**Documentation of 2020 Olympic Pinball Game**

|  |  |
| --- | --- |
| Group: May Hair Be with You | |
| Xiaohua YU | 12987412 |
| Xuxi Chen | 11945045 |
| Molin Yuan | 13305406 |

1. **Background**

This project is a 2D-type pinball game called "2020 Olympic Pinball Game". It is designed to promote the Olympic Games 2020 and the game of pinball play. It allows two users to control the Attacker and the Defender respectively. To compete who is the first race to get score five. It is similar to a five rounds Penalty kick in football. The Attacker has ten seconds per game to control the ball to go to the scoring area through several obstacles via the mouse. And once the ball enters the scoring area. The Attacker gets a point. On the other hand, the Defender can move around the scoring area and have and only one chance to eject the ball. Once the Attacker did not reach score area more than ten seconds, the Defender gets a point.

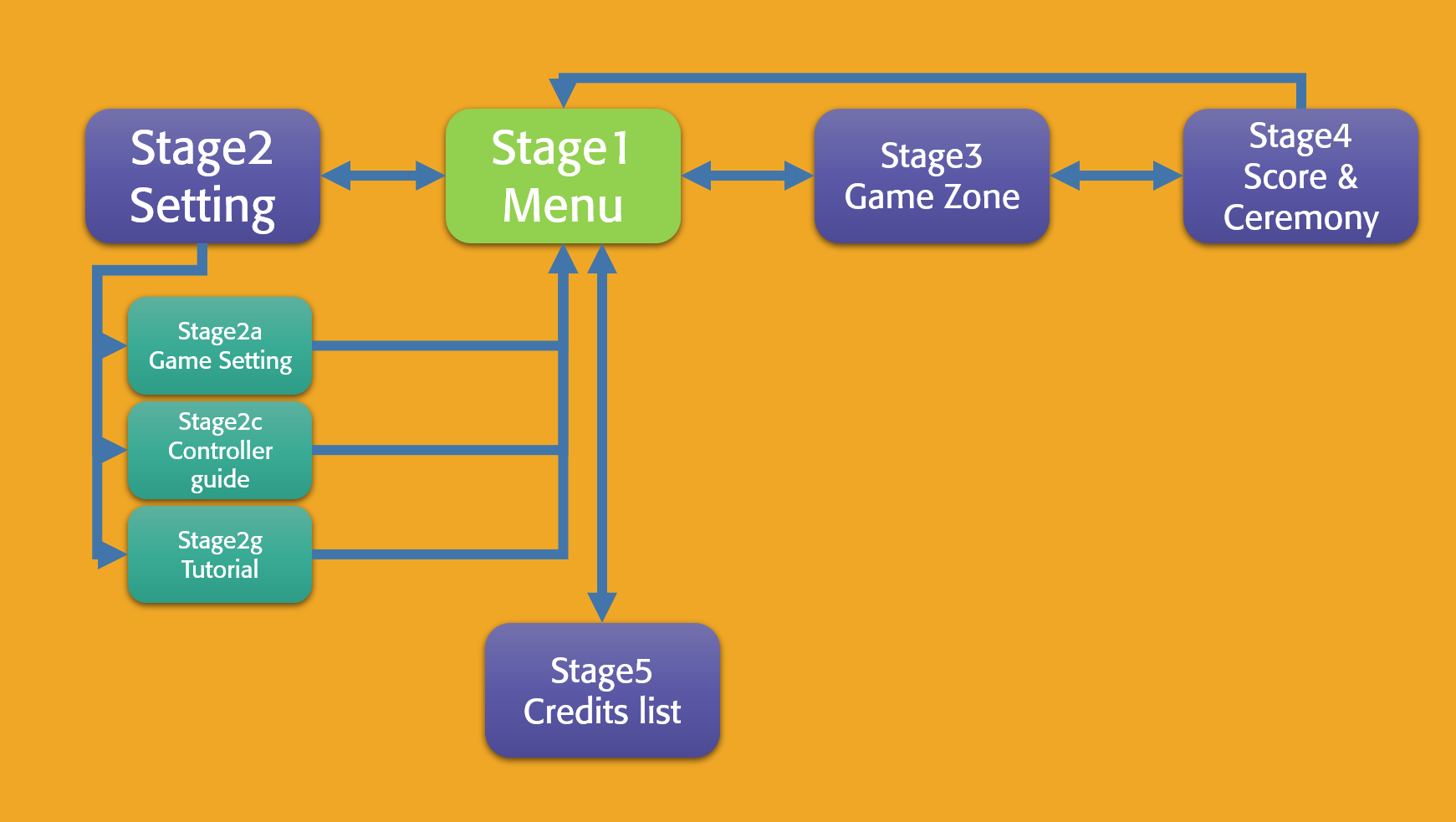
If the process of application development can be divided into several phases such as concept, design, prototype verification, module development, module validation, integration, beautification, debugging, optimization. The phase from concept to integration has been completed and the cycle of beautification, debugging, and optimization is being repeated at present.

The objectives of this project are as follows:

* 1. Attract more young people to learn about the Olympic Games,
  2. Give users the joy of competition and interaction without creating a strong sense of stress and frustration.

1. **Design Criteria**
2. **The process structure is clear and reasonable**

This program contains multiple interfaces (Figure 1). With Stage1 (Menu) as a whole at its core. It is reasonable and not tedious to allow the user switching among different Stages. All Stages can be linked to Stage1. There is no stage that is isolated and cannot be returned.



(Figure 1)

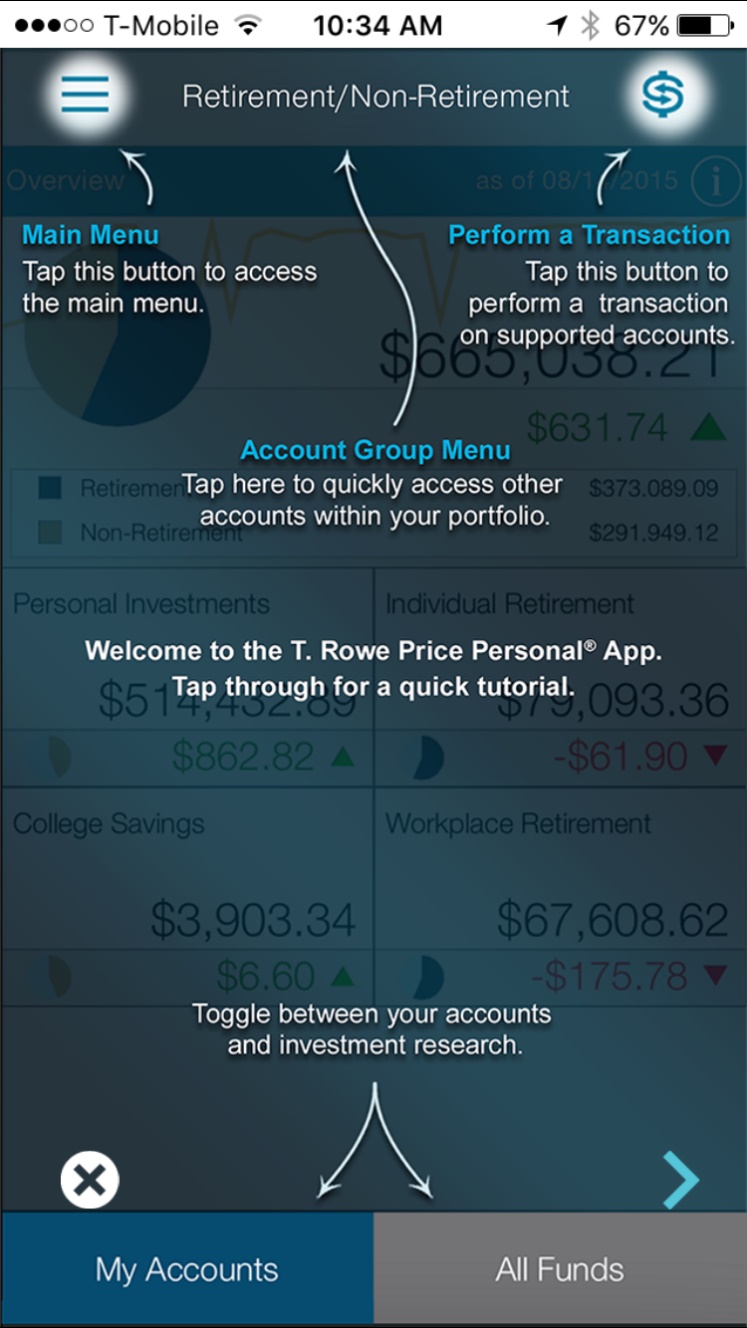
1. **Full consideration of user needs, give users a full choice right.**

At the beginning of our design application, we often do role shifts. Think about the psychological state of the user's role. Users often appear as "I just forgot to look at the instructions and want to go back to the next level of the page", "I want to go straight back to the next game instead of looking at the score settlement interface every time." or “I want to get straight out of this idea" and so on. We believe that users want to use the product as unrestricted as possible. This is known as the "Usability" in interactive design (Preece, J.Y., Rogers, and H. Sharp. 2002).

After balancing development difficulty, development time, and user needs. We decided to meet the "basic choice needs". For example, almost all interfaces can switch to each other. The Setting interface has three sub-stages that allows users to switch back directly to any one of these pages Menu Stage. And when the user is in Stage4 (Score and Ceremony zone), you can choose to start a new game directly (return To Stage3) or end the game (return to Stage1) and so on.

1. **Guidance for use**

We hope that for users who have never been in contact with this game, they will be able to get started first and not be confused about the operation of the game. In order to achieve this requirement, a more reasonable way is to pause the game screen and highlight the main game function area when the user first enters the main screen of the game. Functions by adding prompt information to these areas to introduce (Figure 2).



(Figure 2)

But limited by the difficulty and time of development, we chose a simpler implementation. The instructions will be displayed on a separate page (Figure 3). However, users generally do not enter the program and then view the setting interface first, but will try to click start to try the game content first. Also, the separation of play and game interface sparing causes the player to lose the way in the game playing. Therefore, we need to consider how to direct the first-time user of this program to the setting interface to view the play. We will try to find ways to optimize it.



(Figure 3)

1. **A unified interface styles**

The overall interface style will make the work more expressive. The users will think it is not a patchwork which is combined from many different things. To this end, we have been trying at the beginning of the design, drawing a large number of prototypes to determine the style (Figure 4).



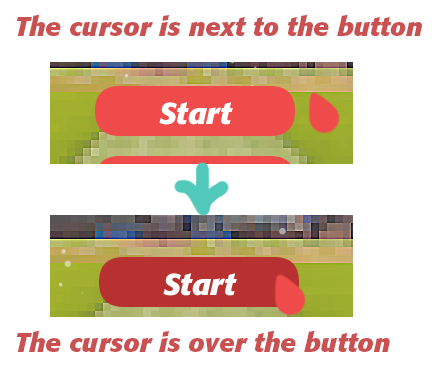
(Figure 4)

1. **Interactivity**

High interactivity should run through the beginning and end of the project. The main manifestations are visual feedback and auditory feedback.

Visual feedback, such as when the mouse moves over a button, changes colour (Figure 5). For example, set a different spacing for the exit program button than the other buttons to avoid accidental touch. At the same time, even if the user does not see the contents of the button, it will make subconsciously think that the first three buttons are one set of functions and the fourth button is another set of functions (Figure 6).

At the same time, we for the app's mouse move button, mouse click, ball impact, the end of the game is set the corresponding sound effects. These allow users to receive feedback as they perform actions to enhance the sense of experience.

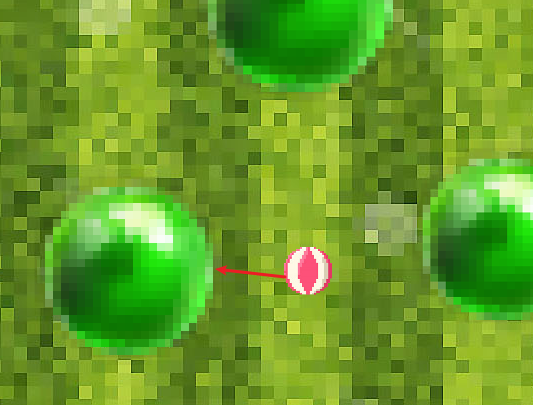


(Figure 5) (Figure 6)

c

1. **Playability**

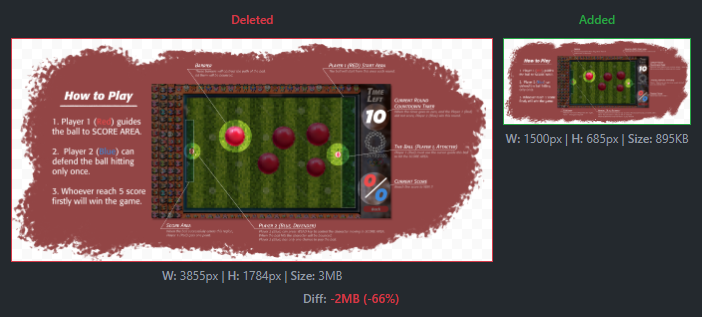
At the beginning of the design, only the user-controlled ball was designed to bounce back after colliding with the obstacle. However, in the production process, we think that we should add more playable content. So, we try to add "Ball movement velocity adjustment", "The obstacle is shifted after impact", "Allow the user to adjust the number of rounds" and other functions to increase the game playability. However, adding features can lead to more ripple effects of bugs. So temporarily shielding and removing some of the functions, leaving only the effect of rebounding after an obstacle collision (Figure 7).



(Figure 7)

1. **Low resource footprint**

As much as possible, the code logic and footage load of the optimizer reduces the resource footprint of the program. Lead to a better experience. For example, the following image (Figure 8) compresses the original 3 MB sized file to 900 KB is about. Reduce the loading time massively.



(Figure 8)

1. **Final Project (video link)**
2. **Critical Reflection**

In the next project we will make the following improvements:

* 1. **Fully understand the capabilities of team members**

In the course of this project, we found that the reasonable allocation of tasks according to employee ability is of great help in advancing the project. At the same time, we believe that the team needs constant running to clearly recognize each member's abilities and the roles he is suited to. For example, a programmer's development logic and processes will not design features that are difficult to implement. Or it does not design effects that take a lot of time but have little impact on the whole project.

* 1. **Clearer division of labour**

A good division of labor helps to balance the pressure strained. and enable each team member to make fuller use of his or her expertise.

* 1. **Develop a detailed timetable**

It is not just a timetable; it is a particularly detailed one. Its primary role is to review on a regular basis and to confirm the progress of the project in order to make timely adjustments.

* 1. **Increased communication with peers**

The "peers" here does not just mean teachers and classmates. Also refers to the Internet, forums in warm-hearted friends, life has been working friends and so on. Their insights are very unique, it will make you always feel fresh and new.

* 1. **Do a lot of testing of the program**

It is not enough just to design and develop products. Each product undergoes extensive testing to identify potential problems. At the same time, if given the opportunity, more non-project participants should be invited to test, their focus and developers are completely different, more focused on their own experience. Sometimes their feedback is very valuable.

* 1. **The choice of tools**

The developer's tool selection is important for project advancement. The first half of this project is produced. Team members are used to editing code in the processing and then manually merge the code by exchanging files. It is conceivable that every integration is painful. So, we started looking for a more convenient solution. In the end, it was GitHub that saved us. Convenient code integration and versioning allowed us to quickly leave those painful memories behind. Similarly, there are regular meetings. The team initially agreed to meet three times a week to make changes. Then decided to adjust to online communication. We chose Discord as an online chat tool. It is easy to push the project forward than ever, and it is also convenient to record the code that needs to be modified.

In summary, we believe that the selection of the right tools is very important to advance the project. We will make the choice of tools an important part of the next project before you start.

1. **Conclusion**

Overall, it is a work experience that juxtapositions both pleasure and stress. Team members feel their progress all the time. On the one hand, it is due to the pressure of the project itself. We have a high standard of design at the beginning of the project, so the end result must meet our expectations. In order to achieve this standard, we need to constantly learn new knowledge, challenge their own not good at the field, but also always maintain a good attitude. On the other hand, working with team members can clearly enhance your abilities. As a designer, for example, Xiaohua was very unfamiliar with code at first. But as he moved the project forward, he gained a deeper understanding of the JAVA language and the OOP idea. At the same time, he found it easier to take another course, Fundamental of Software Development. Additionally, Xiaohua provides excellent examples of visual and interactive logic references at the beginning of the project. And in the production process also constantly adjust and optimize the design. Xuxi and Molin, as programmers, under the influence of Xiaohua, he understanding of design is also more comprehensive and in-depth.

We also believe that the benefits of completing this program are not limited to the score of this course. It can be linked to other courses. Project Management, for example. This project is a small project, and the To-do List we set up at the time of the project (Figure 10 & 11) Like an original version of Work Breakdown Structure. Writing code makes it easily connected to Fundamental of Software Development courses. It is like a combination of the skills we have learned.



(Figure 10) (Figure 11)

Finally, we need to mention teamwork. At the beginning of the course, we were a group of five. But because the two students temporarily decided to go to Europe as an exchange student, the team left three people. This is our misfortune. Fewer numbers of people mean that the project is more stressful. But it is also a blessing for us. Because small teams are easier to manage and communicate with. In the course of this project, we work together to solve existing and new problems. The end result is something we all make proud of.

1. **Reference**
   1. Preece, J.Y., Rogers, and H. Sharp. 2002, *Interaction Design: Beyond Human-Computer Interaction*, Wiley, New York.
   2. Olympic News 2016, *IOC APPROVES FIVE NEW SPORTS FOR OLYMPIC GAMES TOKYO 2020*, <<https://www.olympic.org/news/ioc-approves-five-new-sports-for-olympic-games-tokyo-2020>>.