

Pitch Document

Processing Sketch

- **Concept:**

- My concept is based on movie 'Ready Player One'. Same as the movie, the player is going to collect three keys (the copper key, the jade key, and the crystal key) to win Halliday's easter egg. In my design, the player has to win three games to collect all the keys: a car racing game, a dancing game, and a key chasing game.



- **Special instructions:**

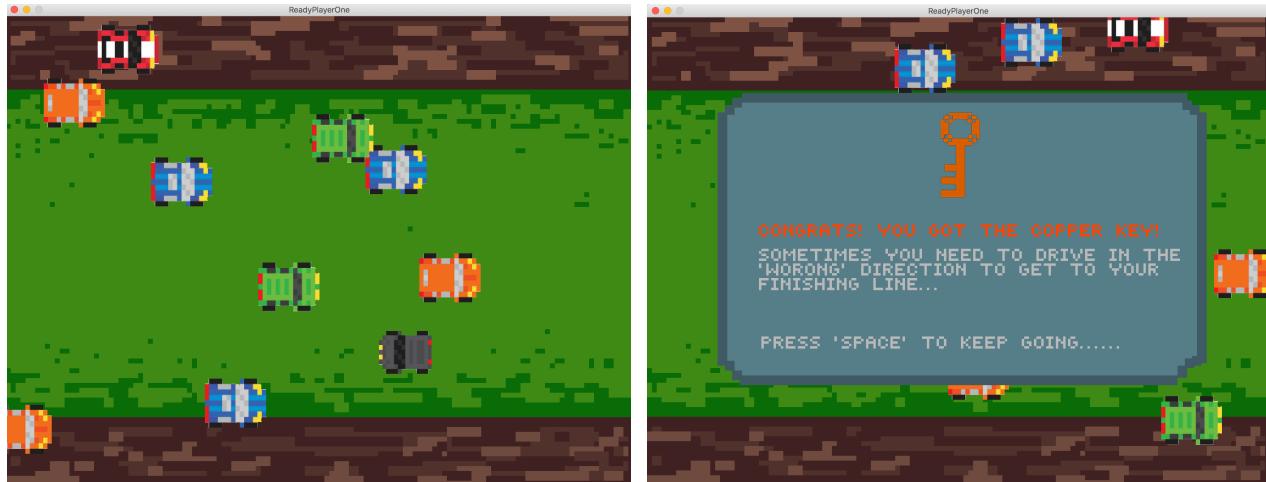
- Users must implement 'Minim' audio library to get access to all the sound effects.
- Once the user has won one of the three games, it is unnecessary (and not recommended) to re-enter the room, or there will be an error counting the games user has won.
- The user basically uses the keyboard to control the game: UP, DOWN, LEFT, RIGHT, 'S', SPACE, and 'R'. There will be instructions at the beginning of each game.

- **Chosen brief:**

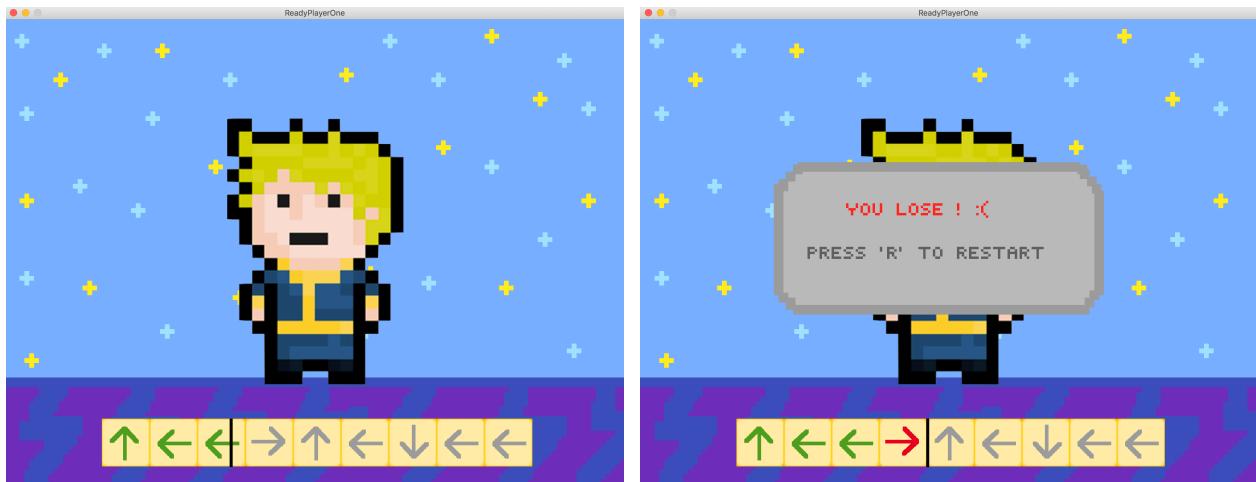
- All the games are related to the movie and the name of keys is the same as those in the movie. The games are connected by a game hall, where there are three houses representing three games.



- The first game is a car racing game, the player is driving in the different direction as other cars and he/she has to avoid colliding with other cars until the finishing line.



- In the movie, the hero saved a girl named Kira among some dancing zombies, so I design the second game as a dancing game, the player should press the corresponding key to follow the beats.



- In the movie, the director wanted to show us that the meaning of game is playing around with it, instead of winning it. So the third game is designed to be a key chasing game where the player has to eat all the snacks before they can get the key.



Design and Aesthetic Choices

• Game style:

- I chose the 8-bit video game style as this game's style.
- Even though the movie 'Ready Player One' is a rather new movie, there are still a lot of nostalgic elements in it. Many characters and other movie elements are from the 70s' and 80s' sci-fi. So it is reasonable to make the style of the game nostalgic.
- On the other hand, 8-bit video game is the memory of generations of people. It's easy to understand and control, so it would reduce the cognitive burden of people when playing the game.

• Storyline:

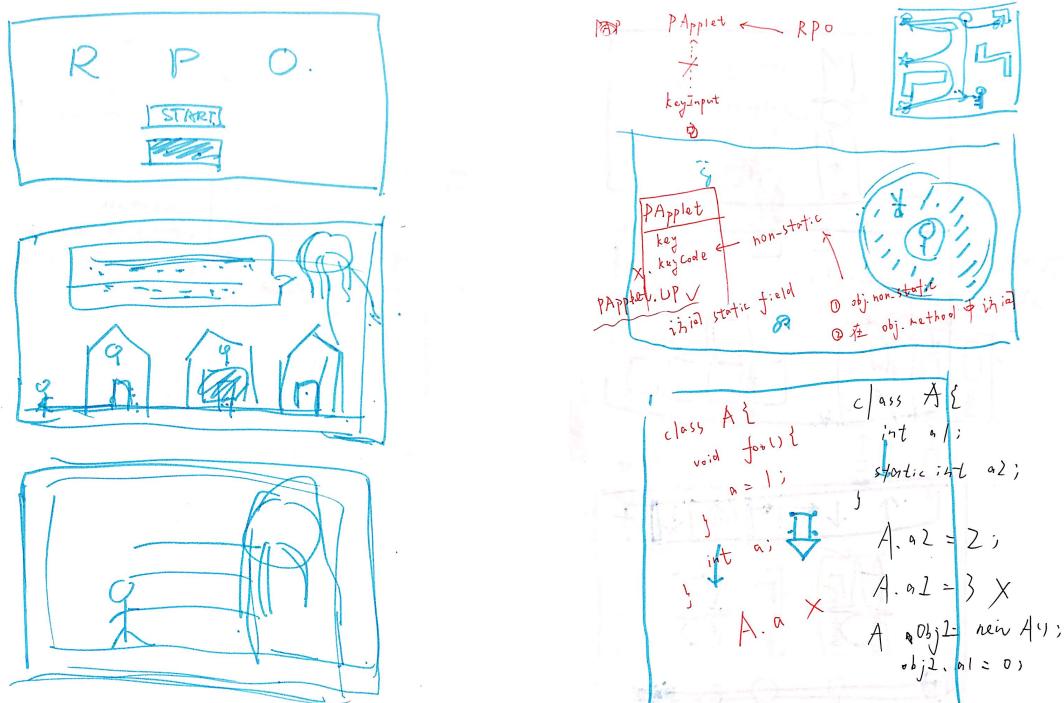
- Like most of the 8-bit video games, the game process is also a simple linear process with three games. A prize to win, three keys to collect - everyone could understand the storyline.

• Game music:

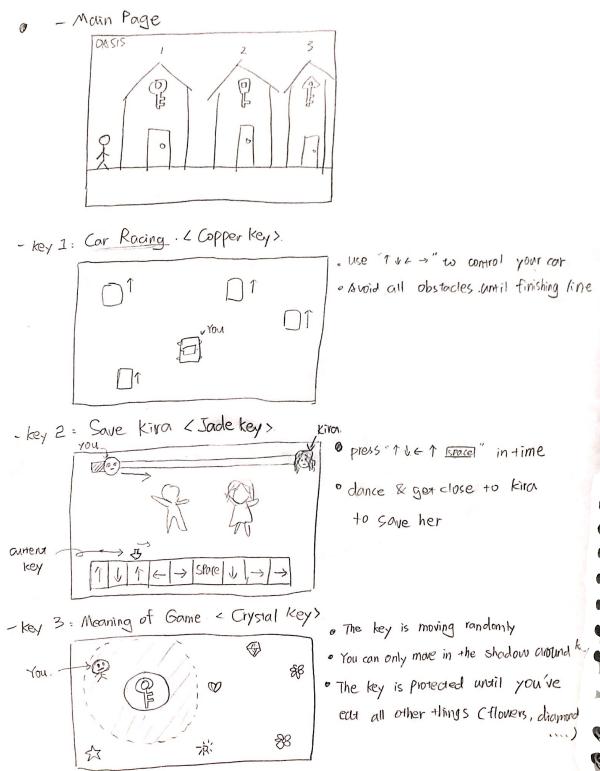
- The game music I chose are mostly sci-fi music as the movie is set in a future world. I also preferred music used in 8-bit sci-fi games as it is more related to the game I designed. The failure and victory sound effects are also conventional that everyone is able to understand.

Approach to the Design Process

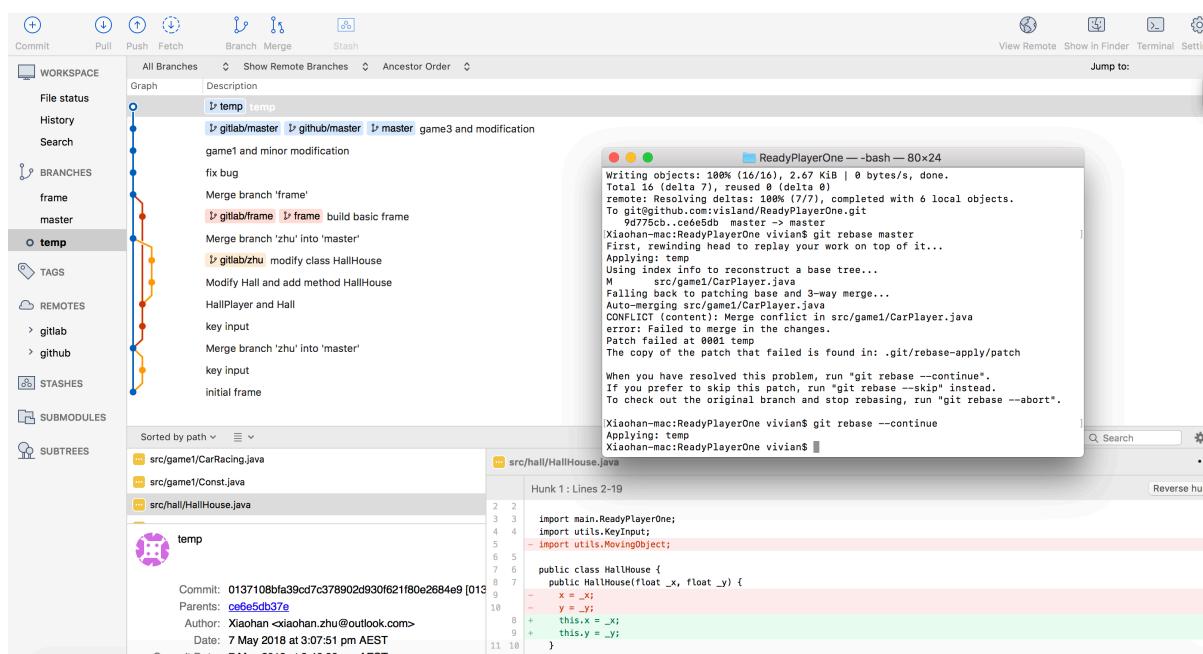
- I was interested in making games so I chose the first brief. After brainstorming, I have generated several ideas:
 - A zombie shooting game
 - A maze & puzzle game
 - A game based on 'La La Land' whose topic is finding lost memory



- I communicated my ideas with my tutor, and he gave me advice that the first two are done by other people before, but the third one seems interesting. But ‘La La Land’ seems to be not closely connected to the game, so it is necessary to find other inspirations.
 - The day after the tutorial, I went to see ‘Ready Player One’ with my friend and it came to me that this would be a perfect movie for creating a game. So I started sketching (See images on the previous page).



- After sharing my idea with my tutor and some of my friends, I decided the frame of the project as a game consisting of three games. Then I started coding.
 - I use GitHub and SourceTree to manage my code.



- I started by creating very simple prototypes of my game in Eclipse, making sure that the code runs successfully. Below are screenshots of my prototypes.

The screenshot shows three Eclipse windows demonstrating game prototypes:

- Top Window:** Shows the Eclipse IDE interface with the Package Explorer containing the project structure for "ReadyPlayerOne". The code editor displays the file "Key.java" which defines a class "Key" extending "CollisionObject". It includes methods for update and display, and logic for calculating velocity based on moving direction.
- Middle Window:** Shows a game screen titled "ReadyPlayerOne". A character with blonde hair and a blue vest is shown. The text "You lose! Press R to restart" is displayed. Below the character is a control panel with large green and red arrows pointing up, down, left, and right.
- Bottom Window:** Shows another view of the game screen with the same character and message. The control panel is identical. The Eclipse interface on the left shows a detailed commit history for the "master" branch, listing numerous commits related to game development.

- After all prototypes run very well, I add images and music into the game. I also moved my code into Processing.



Creative Influences

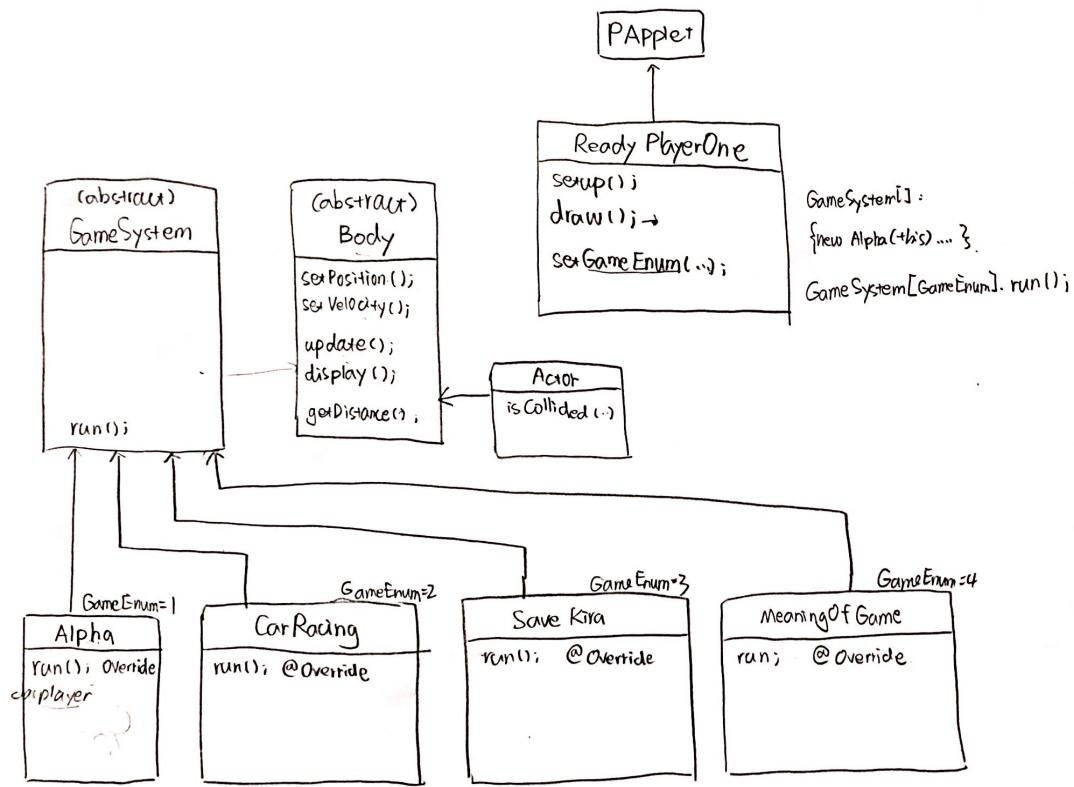
- **Creative shooting game - Duel** (<https://www.fal-works.com/minigame-posts/duel>)
 - When I began framing my game, I found it difficult to have multiple scenes in one Processing sketch. I tried using PGraphics to create different layers but it did not work well. So I looked at the code of a shooting game named Duel to find inspirations. This game contains different screens (start page, fail page, etc.).

```

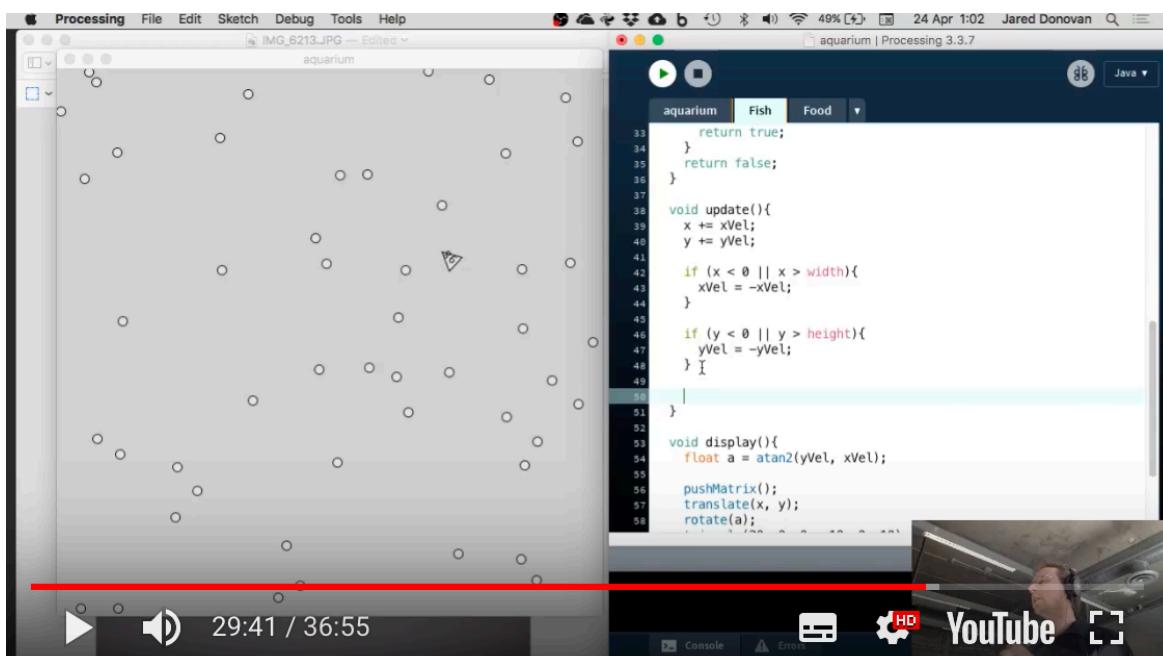
// Title: Duel
// Author: FAL ( https://www.fal-works.com/ )
// Made with Processing 3.3.6
/* Change log:
Ver. 0.1 (30. Sep. 2017) First version.
Ver. 0.2 ( 1. Oct. 2017) Bug fix (unintended change of strokeWeight), minor update
Ver. 0.3 (10. Feb. 2018) Minor fix (lack of semicolon).
Ver. 0.4 (12. Feb. 2018) Enabled scaling.
*/
/* @pjs font="Lato-Regular.ttf"; */
/*
The font "Lato" is designed by Łukasz Dziedzic (http://www.latofonts.com/).
This font is licensed under the SIL Open Font License 1.1 (http://scripts.sil.org/OFL).
*/
// CAUTION: spaghetti code!!!
private static final float IDEAL_FRAME_RATE = 60.0;
private static final int INTERNAL_CANVAS_SIDE_LENGTH = 640;
private static final boolean USE_WEB_FONT = false;
KeyInput currentKeyInput;

```

- While reading the code, I drew a UML of it. Then I realised that the author used an abstract class called GameSystem to manage different scenes. So I drew a draft UML of my own code and started coding.



- Tutorial - Fish Class Demo** (https://blackboard.qut.edu.au/webapps/blackboard/content/listContent.jsp?course_id=_137441_1&content_id=_7363295_1)
 - In the demo video the tutor showed how to create a fish moving and bouncing within the screen. This gave me the inspiration of the third game - the key chasing game - about how the key would move and bounce with the screen.



- **READY PLAYER ONE - 8-Bit Trailers** (<https://www.youtube.com/watch?v=43Rz2v6tKkI>)
 - This creative video illustrated the movie in 8-bit video game style, which inspired me of the style of my game.



References

- JoBlo Videos. (August 22, 2017). *READY PLAYER ONE - 8-Bit Trailers* (2018) Steven Spielberg, Ernest Cline sci-fi movie[Video file]. Retrieved from <https://www.youtube.com/watch?v=43Rz2v6tKkI>
- Soundimage. (n.d.). *8-BIT MAYHEM*[Audio file] Retrieved from <http://soundimage.org/looping-music/>
- Soundimage. (n.d.). *GAME MENU_v001*[Audio file] Retrieved from <http://soundimage.org/looping-music/>
- Soundimage. (n.d.). *SECTOR OFF LIMITS*[Audio file] Retrieved from <http://soundimage.org/sci-fi-2/>
- Soundimage. (n.d.). *FUTURE URBAN*[Audio file] Retrieved from <http://soundimage.org/sci-fi-2/>
- Splashdust. (February 17, 2009) *negativebeep*[Audio file] Retrieved from <https://freesound.org/people/Splashdust/sounds/67454/>
- unadamlar. (April 3, 2016) *GoodResult*[Audio file] Retrieved from <https://freesound.org/people/unadamlar/sounds/341985/>
- FunWithSound. (November 25, 2016) *Victory Celebration Movie Score.wav*[Audio file] Retrieved from <https://freesound.org/people/FunWithSound/sounds/369252/>