

PUNCH CARD CODE LEARNING

An interactive code learning game

Overview

Problem Domain

The domain of my concept is related to creative code learning, nowadays, programming skills have become more and more important and most of the major countries have added programming into children's courses. But even in this case, there are still some people who can hardly master programming skills. So my project goal is to develop a set of robust and effective programming learning equipment which can arouse people's enthusiasm for programming learning and help them learn programming skills efficiently through a more interactive teaching method. The target audience of my project are the people who never learn coding language. Another target group is physical gamers, so they may learn a lot about programming when they get involved in this interactive game.

Related Works

I was inspired by the punch card and 'Blockly'. The punch card is a kind of stiff card containing digital data in it. The data are represented by the presence or absence of holes in predefined positions, by swiping them on a punch card reader, the data can be sent to the computer. Now the punch cards are mostly used for recording voting in the voting machine. The 'Blockly' is a visual programming environment, and its main interaction is similar to puzzles. It separates the code into several code blocks, and users can combine different code blocks to achieve target functions, which is a good way to teach teenagers to learn how to code. As for the papers, the paper from Frederick (2011) introduced several reasons of why the students have difficulties on learning programming, complex cognitive activity (Robins et al, 2003) of programming itself, lack of motivation (Mathrani et al, 2016) and difficult association with the code and system (Milne & Rowe, 2002) are the three main factors. Another paper from Li & Watson (2011) analyzes 18 existing game-based programming learning systems is also useful for my project.

My Concept

As for our previous team concept, what we want to make is an interactive game to help teenagers learn the logic of programming, the control module consists of camera and instruction blocks and the other part is a physical maze with a small robot in it. The way people control the movement of the robot is to place the command block in correct order, these commands are like go forward, go backward, turn left, turn right, and some instructions for looping. The user only needs to place the side with the command they want to enter to the top, the camera will detect it and use it to control the movement of the robot.

On the basis of our previous team concept, I made some changes. Firstly, I changed the previous way of inputting information because our previous idea used blocks as information carriers which is very clumsy. So I changed that to punch cards, users just need to swipe the card, which is easier to operate. Another thing modified is that I chose to use web-based games to interact with users, rather than robots and maze, because moving the game to the screen has better expressiveness, and can save space in order to adapt it to the teaching environment. So, my current concept will provided users with different Instruction cards and each card contains a code fragment, what the users should do is to swipe these punch cards on the punch card reader, and the code fragment will be insert into the web-based IDE to control the character's movement in the web-based game to complete different tasks.

Design Process

12th March(Week 3)

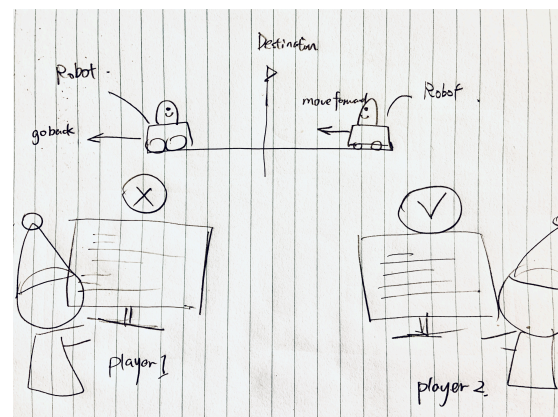
We formed our team based on the problem domain we chose in week3, and we conducted our first meeting. Our problem domain is about 'Creative Learning', so in this meeting we chose programming as our research direction from four topics we discussed: circuit learning, coding education, cooking learning and instrument learning. After determining our research direction, we start to brainstorm ideas about how to make programming learning more creative and interactive. At this stage, the ideation method we used is the affinity diagram. We summarized the ideas of each team member into keywords and sorted them into an affinity map, and then chose the most suitable plan from them. From the discussion about project purpose, target audience and main interaction three aspects, we finally decided our team concept.

24th March(Week4)

We did the proposal pitch as a team and got some feedback about our ideas. The feedback pointed out some problems on our team concept and we made some discussion around these problems and tried to make some changes. Firstly, we will change the practice content to some existing programming languages like python, c and javascript in order to help people get in touch with real coding language to improve their understanding. And the game format will be changed to a box moving game, because the previous concept is too limited.

5th April(Week 5)

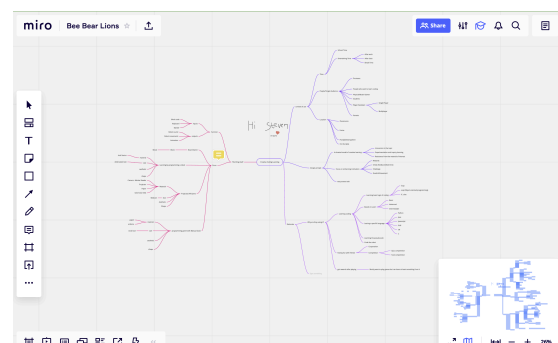
In week5, I started to explore my individual concept on the basis of our previous feedback and team concept and I got my Arduino kit and started to do some research on it. As for my individual idea, it is still a multiplayer game, the two robots were placed at two different sides, and the destination is in the middle. We will let the users fill in the blanks in the coding questions to help them understand the syntax of the programming language. When the player answers the question correctly, the robot will move forward and if they didn't complete the question, the robot will go backward, so the player who gets to the destination first will win the game.



The difficulty of this game will gradually increase from easy to hard, so that people at all levels can handle it successfully. In terms of my attempt on the Arduino, I just began to try connecting some simple components like LEDs and motors.

12th April(Week6)

In week6, we did an in-class activity to break down our team project on miro, this activity asked us to separate our team concept from context and the thing itself two aspects. And for each aspect, it can also continue to subdivide. This tree diagram can help us have a better understanding of our project and get more inspirations and in-depth consideration of our group project. Apart from that, I still continue to explore the arduino with the help of tutors.



19th April(Mid-Break)

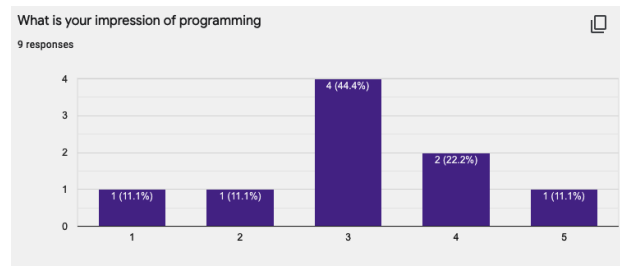
In mid-break, I conducted user research, which is a questionnaire conducted with university students to discover the factors that hinder people's programming learning.

This questionnaire has six simple questions, and the questions in it are mostly about people's impression of programming and the obstacles of programming learning. From the answers of these questions, I found some useful information:

- Most of the people thought programming is hard to learn.(shown in the picture above)
- Most people hold the opinion that interest is an important factor to encourage people to continue learning. Only a few people thought that pressure can also encourage us to learn.
- The syntax and logic of programming learning is the most difficult part.
- A warm heart feedback is better than error information and bugs.

The questionnaire link is shown below:

https://docs.google.com/forms/d/e/1FAIpQLSefXhwyK1xaV1bvw23tnSj7PRHW9Wvba3HpOmJDm8GCQr6cdw/viewform?usp=sf_link



26th April(Week7)

In week7, I asked some questions about improving the interaction in my concept and got the punch card reader idea from Steven, so, I want to use it as an input device in my project. The idea Steven mentioned is an interesting idea called 'punch card', which is a way to enter information physically, people can insert a small card with holes into the card reader, the card reader can recognize the content of the card and send it into the computer. In terms of the Arduino exploring, I have successfully used socket.io library to control the Arduino LEDs with the button on the browser and I made a simple punch card reader with the cardboard.



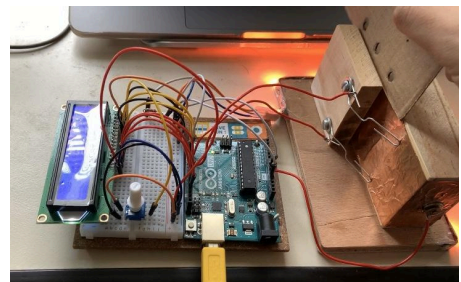
3rd May(Week8)

In week8, I still continue to work on my punch card idea. Firstly, I want to connect the ethernet shield with Arduino to send the data to the phpmyadmin, but some errors occurred. So I tried another solution to achieve this function, that is using a serial port library to fetch data from Arduino and show it on the node js console. And I also successfully displayed the data in the browser, so that the punch card reader has already linked with the browser, and each time we swipe the card, the card content will show up on the screen.



9th May(Week9)

In week9, I added a LCD screen to show the data of the card we swiped to help users check the card content more conveniently. I also added a LED strip on the punch card reader, and each time users swipe the card, LED lights will flash different colors according to the content of the card, which improved the interactivity of the device.



Interaction Plan

Interaction Paradigm

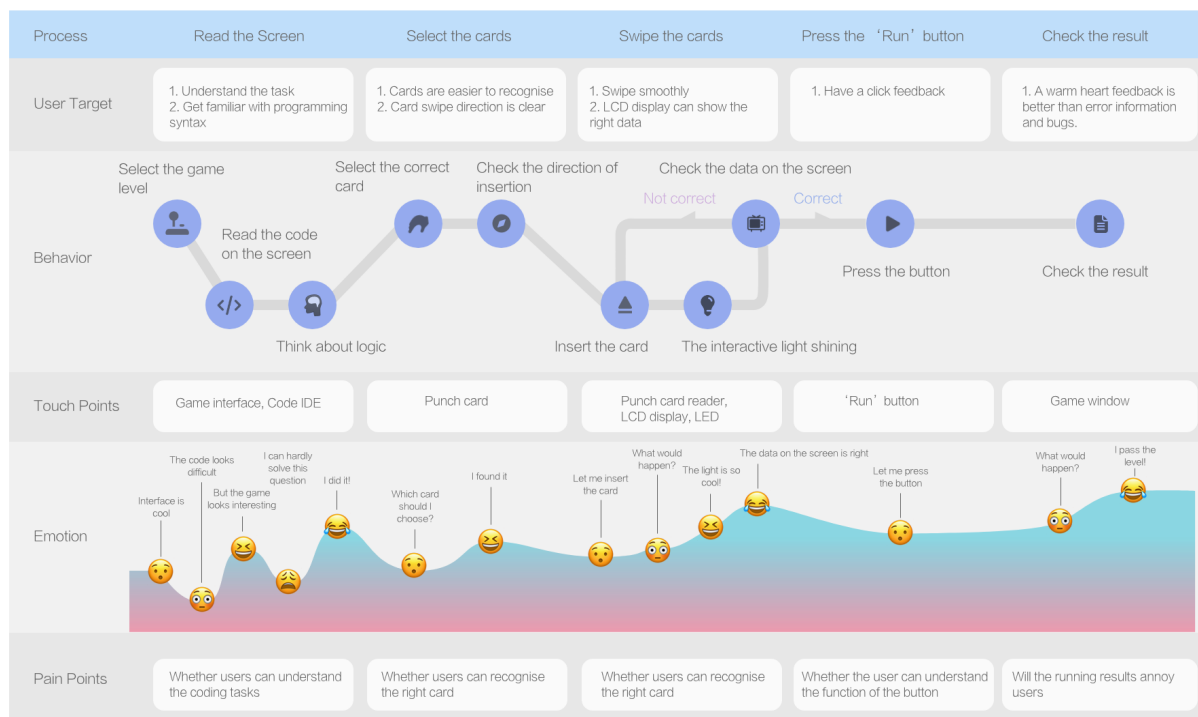
The interaction paradigm of my work can be divided into two parts: physical part and digital part.

As for the physical part, the core interaction is the swipe the card with the punch card reader. We will provide users with some punch cards, The user needs to select the card they want and swipe the card in the direction of the arrow on the punch card reader. Each card has a hole. When the stylus passes through the hole, the binary data is generated according to the combination of the hole position and passed to the computer. Apart from this, another interaction is interactive LED strip, when the user swipes a card, the LED strip will flash for two seconds as a feedback for successful card swiping, and the color of lights will be different according to different cards. The purpose of this interaction is to enhance the appearance and fun of the product. The last physical interaction is the LCD display, after the user swipe the card, the information contained in the card will be displayed on the display, the user can check whether the card input is correct.

In terms of the digital interaction, there will be a run code button, programming IDE and a game window on the browser. Each time users swipe a card, the data will be sent to the IDE, when the user clicks on the run code button, the character in the game window will move according to the command in the coding IDE.

User Journey

According to the using process of my work, I made a user journey map, and it is shown below:



In this user journey map, the main interaction process can be divided into five parts. In the behavior part, these processes are subdivided into detailed user operation procedures, then in the picture, the user goals, touch points and pain points of each stage are analyzed. The emotion part in this diagram shows how the users react in different stages of product using.

Project Objectives and Success Criteria

In order to assess the success of the project, I make the following success critique:

Project Objectives	Success Criteria
Users can select different learning levels according to their ability.	- A select level page should be added, the users can select different coding difficulties according to their needs.
Users can have fun in the coding game and arouse enthusiasm about programming	- The game function should be achieved - The theme and content of the game need to be connected with programming - The game needs to be interesting
Users should have a good operating experience	- The punch card reader can detect the card correctly - Rich interactive feedback (sound, light, vibration) after every swipe
Users can get tutorial to help them improve programming	- The website should have a good guidance for user - A tutorial page can be added for users
User want to get a warm heart feedback	- Users should get a warm heart feedback instead of just error information and bugs
Users prefer a better interface	- The interface information should be clear - The interface layout should conform to user habits - Interface must meet aesthetic standards
Users want to correct mistakes by themselves	- Give suggestions for their syntax errors

References

- Mathrani, A., Christian, S., & Ponder-Sutton, A. (2016). PlayIT: game based learning approach for teaching programming concepts. *Journal of Educational Technology & Society*, 19(2), 5-17.
- Khamphroo, M., Kwankeo, N., Kaemarungsi, K., & Fukawa, K. (2017, May). MicroPython-based educational mobile robot for computer coding learning. In *2017 8th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES)* (pp. 1-6). IEEE.
- Li, F. W., & Watson, C. (2011, December). Game-based concept visualization for learning programming. In *Proceedings of the third international ACM workshop on Multimedia technologies for distance learning* (pp. 37-42).
- Milne, I., & Rowe, G. (2002). Difficulties in learning and teaching programming—views of students and tutors. *Education and Information technologies*, 7(1), 55-66.
- Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer science education*, 13(2), 137-172.

Video Link : <https://www.youtube.com/watch?v=gsbMmCZT7mQ&t=10s>