I build various small software gadgets to bring convenience and fun to my life. Usually, I got my ideas from my daily life, fictional novels I read as well as my discussion with my friends. These ideas are turned into my real software products using programming languages. I have created more than 20 games and utility scripts in this way through my high school years, including implementations and modifications of popular games, games created by me or my friends, online game hacking scripts, daily utilities that improves life qualities like automatic notification feeds, as well as programming utilities and libraries that I produce as by-products of my creation activities.

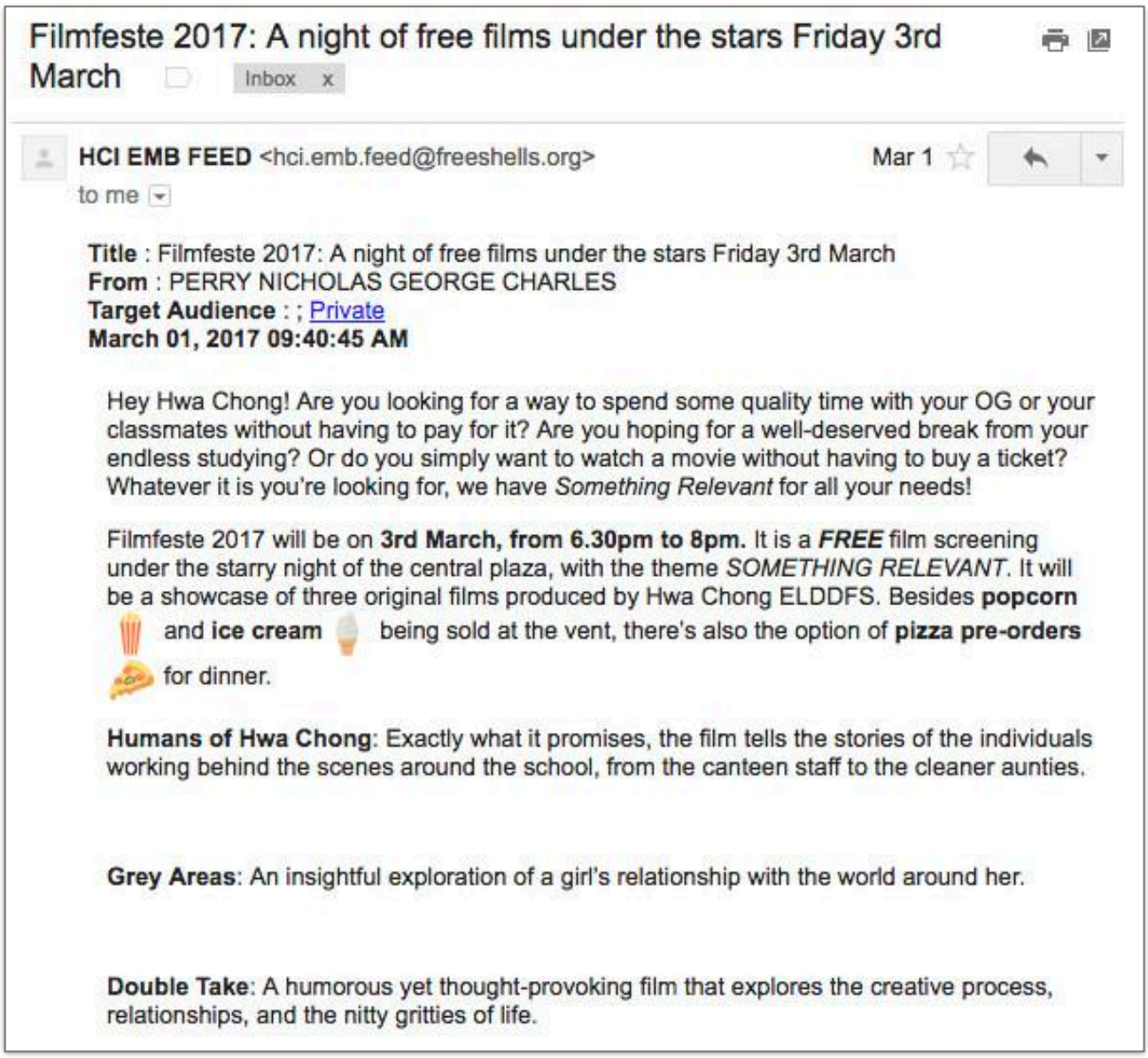
I built most of my creations using my computer, while a few are hosted on free servers I registered online. I use a variety of programming languages, most common ones being Python, Swift and Javascript, as I choose the most appropriate tool for each project. For most projects, I simply write code with a text editor, and compile / execute with command-line tools, but I also use some specific developer softwares like Xcode and Google's Android Studio for convenience in certain projects. For graphical applications, I use Photoshop for designing and rendering components of the user interface.

I started programming under my father's guidance, but have been self-studying most programming languages, developer tools and libraries I now use through online tutorials like Apple's Swift documentations. For specific code snippets that I need help with, I access Stack-Overflow for an answer.

The following is a listing of my proudest creations:

School Message Notification Feed

Through Python scripts running on freshells.org free servers, I constantly fetch new messages from our school's online message board by making HTTP requests, and transform them into emails as seen in the screenshot and push to my mailbox. The message body is extracted as HTML from our school's webpages, so that the inner format of a message is preserved and rendered as it is inside the email. This allow me to receive push notifications everytime a message is posted on the school message board, hence saving me the time to repeatedly login to the message board and see if there is anything new.



Filmfeste 2017: A night of free films under the stars Friday 3rd March

Inbox x

HCI EMB FEED <hci.emb.feed@freeshells.org> Mar 1

to me

Title : Filmfeste 2017: A night of free films under the stars Friday 3rd March
From : PERRY NICHOLAS GEORGE CHARLES
Target Audience : ; Private
March 01, 2017 09:40:45 AM

Hey Hwa Chong! Are you looking for a way to spend some quality time with your OG or your classmates without having to pay for it? Are you hoping for a well-deserved break from your endless studying? Or do you simply want to watch a movie without having to buy a ticket? Whatever it is you're looking for, we have *Something Relevant* for all your needs!

Filmfeste 2017 will be on **3rd March, from 6.30pm to 8pm.** It is a ***FREE*** film screening under the starry night of the central plaza, with the theme *SOMETHING RELEVANT*. It will be a showcase of three original films produced by Hwa Chong ELDDFS. Besides **popcorn** and **ice cream** being sold at the vent, there's also the option of **pizza pre-orders** for dinner.

Humans of Hwa Chong: Exactly what it promises, the film tells the stories of the individuals working behind the scenes around the school, from the canteen staff to the cleaner aunties.

Grey Areas: An insightful exploration of a girl's relationship with the world around her.

Double Take: A humorous yet thought-provoking film that explores the creative process, relationships, and the nitty gritties of life.

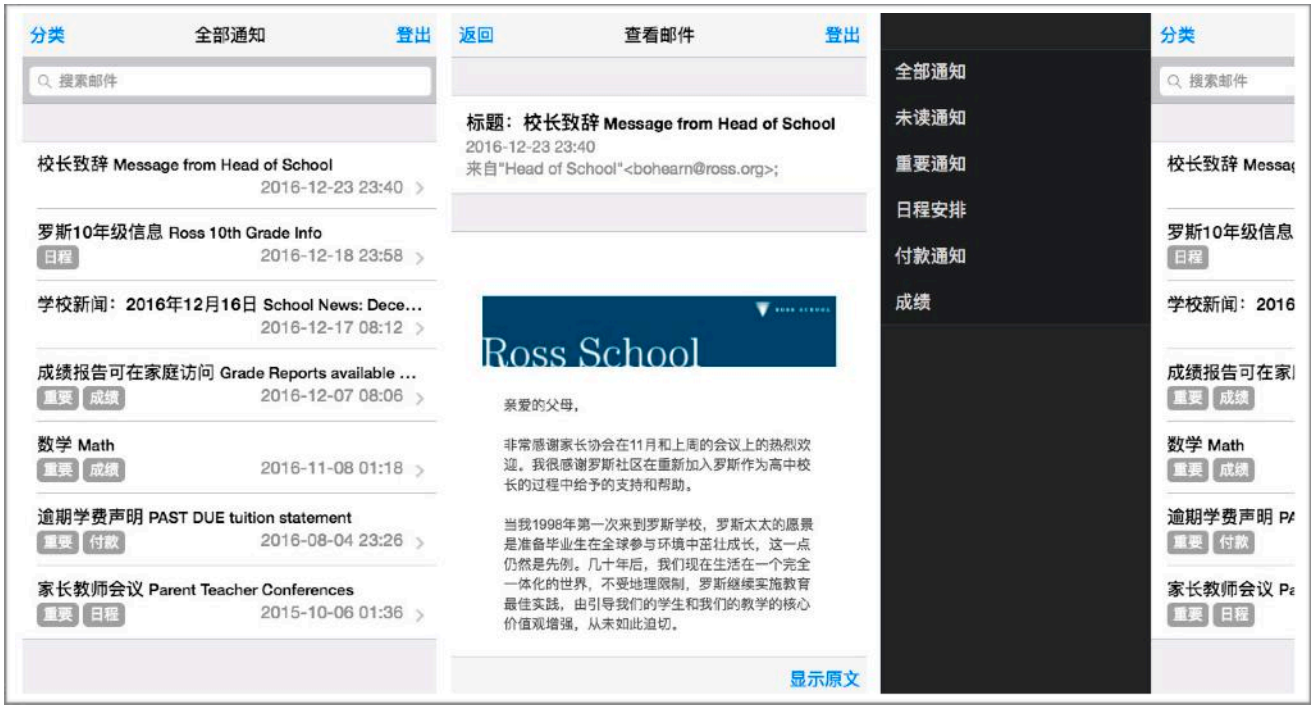
My message fetching script is easily configurable and can access the messages in any school account with username and password provided. I have used this feature to offer the message notification functionality to many of my friends in school.

The source code can be accessed at https://gist.github.com/beanandbean/1a14f606680c5af5188276553228c529

School Mail

I did this project to better facilitate the communication between parents and their children studying overseas. In a community of Chinese students in Singapore, my parents and my friends' parents all often encounter the issue of being unable to understand the english email notifications sent by our school. Therefore, I made the School Mail system to auto fetch

school notification email, translate them into Chinese through Yandex's API while preserving the original HTML structure, and categorize them into important emails, schedule notifications, grade reports etc. using Google's online AI natural language processing service api.ai.



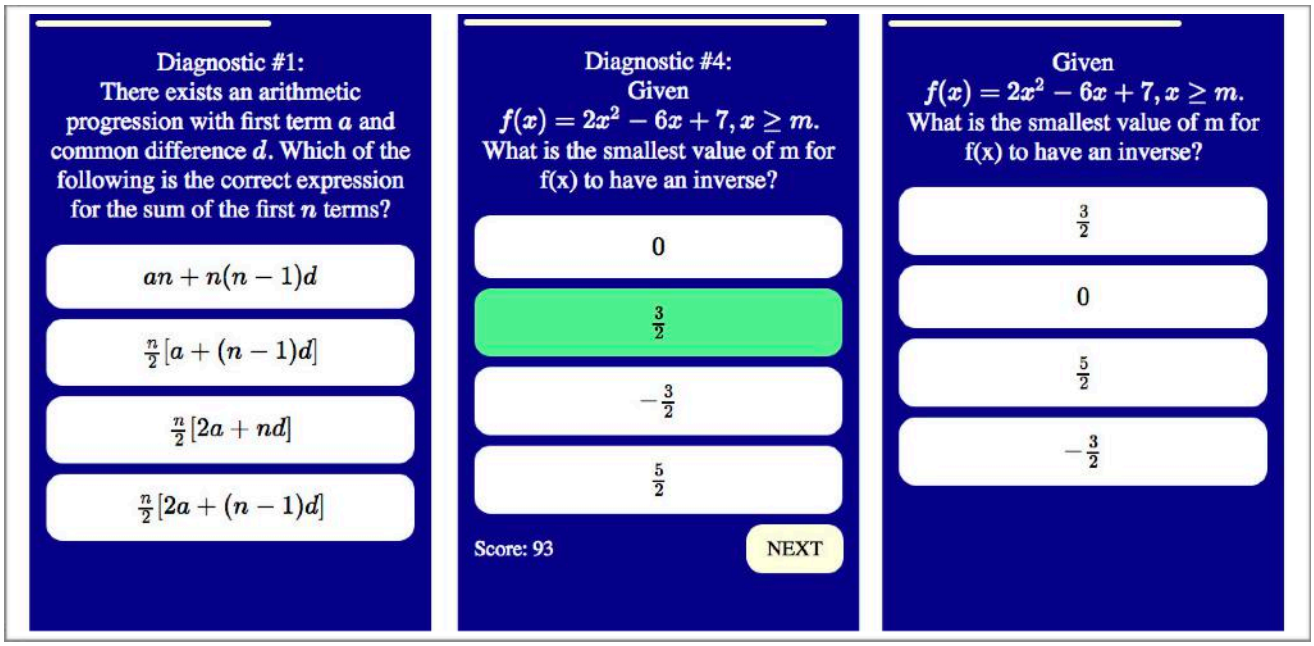
The application is built using Javascript library Framework7, while background email-fetching scripts are written in Python. It is built as a volunteer project for Weijun Technology, an overseas education firm in China, and is currently being used to benefit Chinese parents who have children studying abroad.

Quiz Know U

A machine learning project I did leading a team of 3 under mentorship of Dr Ji Feng from NTU, Singapore. We used recommender system to analyze different students' performance on different math problems, measured by hesitation time before the correct answer is chosen, and evaluated similarity and differences among a question set we created of 100 multiple-choice questions.

We then created a final product as shown on the screenshot to predict and suggest about students' math practice. As seen on the two images on the left, the user first attempts ten diagnostic questions, where they are scored on the correctness and time used for each question. Then we use our machine learning model to predict the user's weaknesses, and

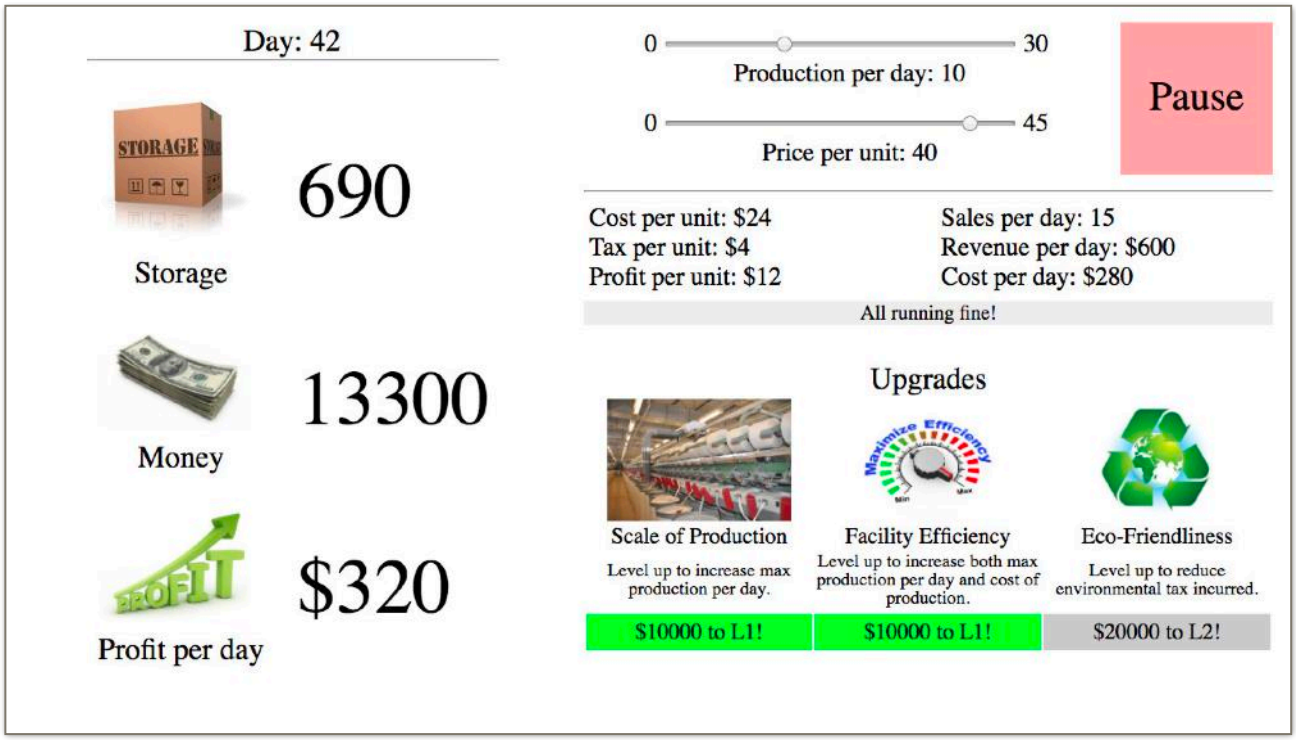
choose questions, as shown on the third image, to allow the user to



specifically practice on his weakness.

The machine learning algorithm is trained using a matrix of 2000 data points from 20 students, using MATLAB scripts. The front-end is implemented in Javascript.

Market Simulator

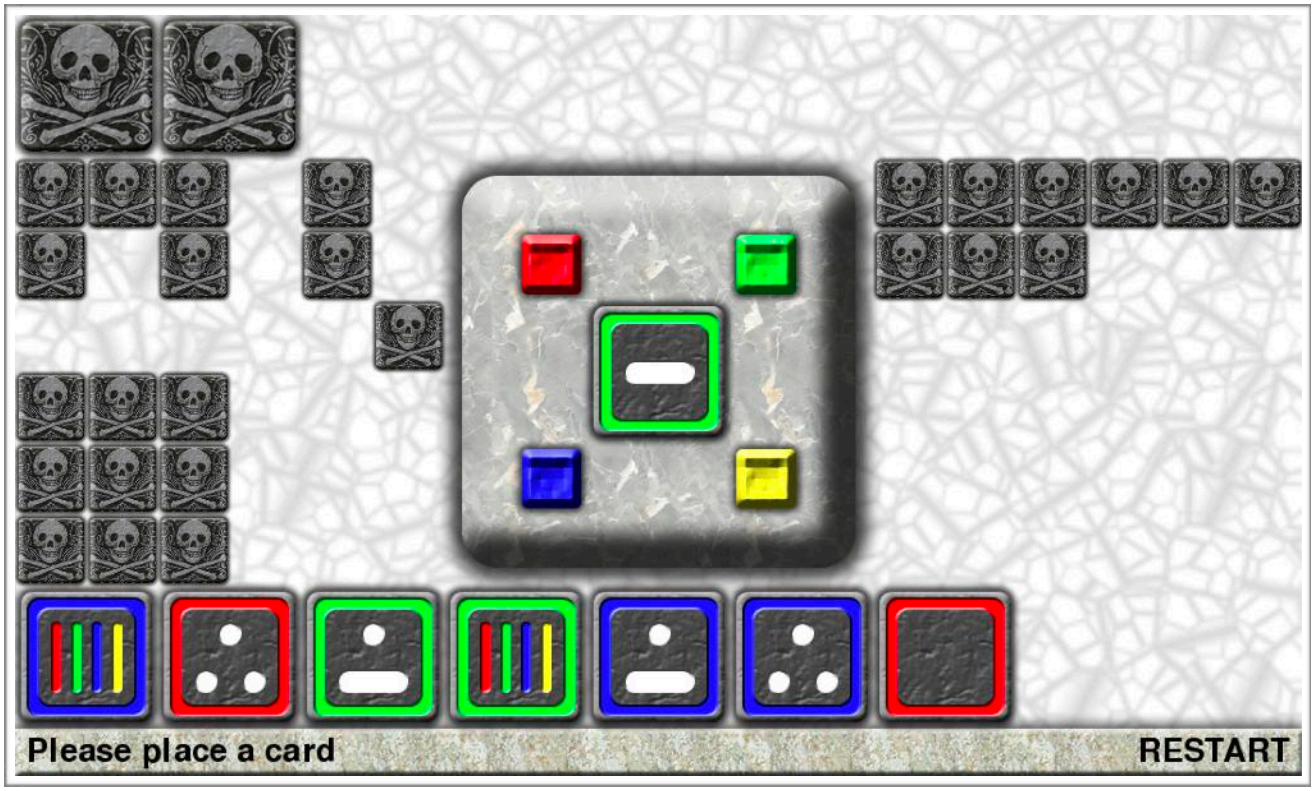


This is a web-based game built with Javascript. In this game, players act as a firm and make decisions about its daily productions, pricing strategies and investments. I utilized demand and supply formulas learnt in Economics lessons to calculate the daily sales of the firm, hence evaluate the changes in the firm's inventory and capital. The player's aim is to make appropriate decisions to max out the total profit of the firm within a shortest time.

I made this game to help my fellow friends learning Economics in high school. I hope with this game students can have a better understanding of the interactions between price, supplies and sales, and learn firm's decision making strategies with less effort by experiencing it themselves.

This game is greatly appreciated by my Economics tutor, who has given me a chance to demonstrate the game in front of the whole class. I am really glad to have helped my friends learning Economics with this.

Mayan Cards



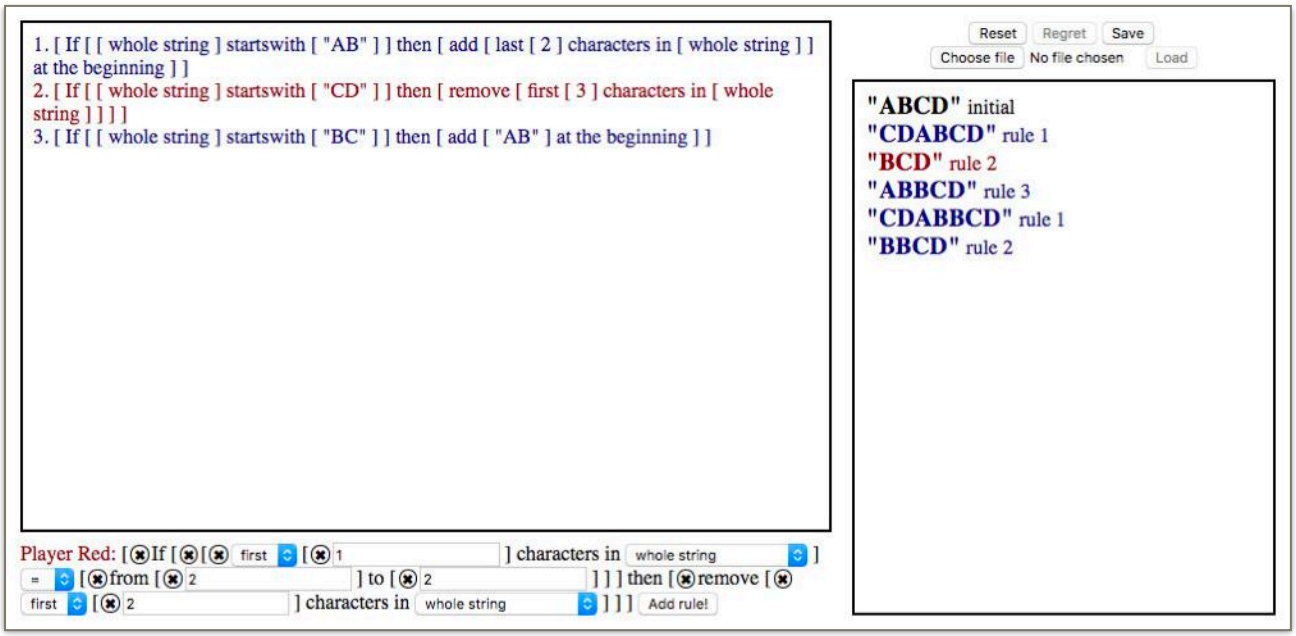
This was my PC reimplementation an iOS game I used to play. It is a card game similar to Uno, where players take turns to place cards that is

either the same color or pattern with the previous one, with some of the cards having special abilities.

I implemented this using Python game library Pygame, with online free image resources processed using Photoshop. This was the first game I implemented using Pygame, so it involved a lot of online tutorial reading, and the animations are not exactly smooth and satisfying, but the final product was quite playable, with a robot opponent implemented using a winning strategy I proposed, which could take quite some effort to defeat. This was popular for a while in the high school hostel I lived, and my friends often gathered around my computer, take turns to wrestle the AI player in the evening.

The source code can be viewed at https://gist.github.com/beanandbean/c816090397b537165baff6be0e59ef2b

Rules Manipulated!



This is a brain game that was mentioned in a fictional novel. Two players take turns to name rules to manipulate a string of character, each rule changing 3 characters at most, and the first person who changes the string of characters to an empty string wins.

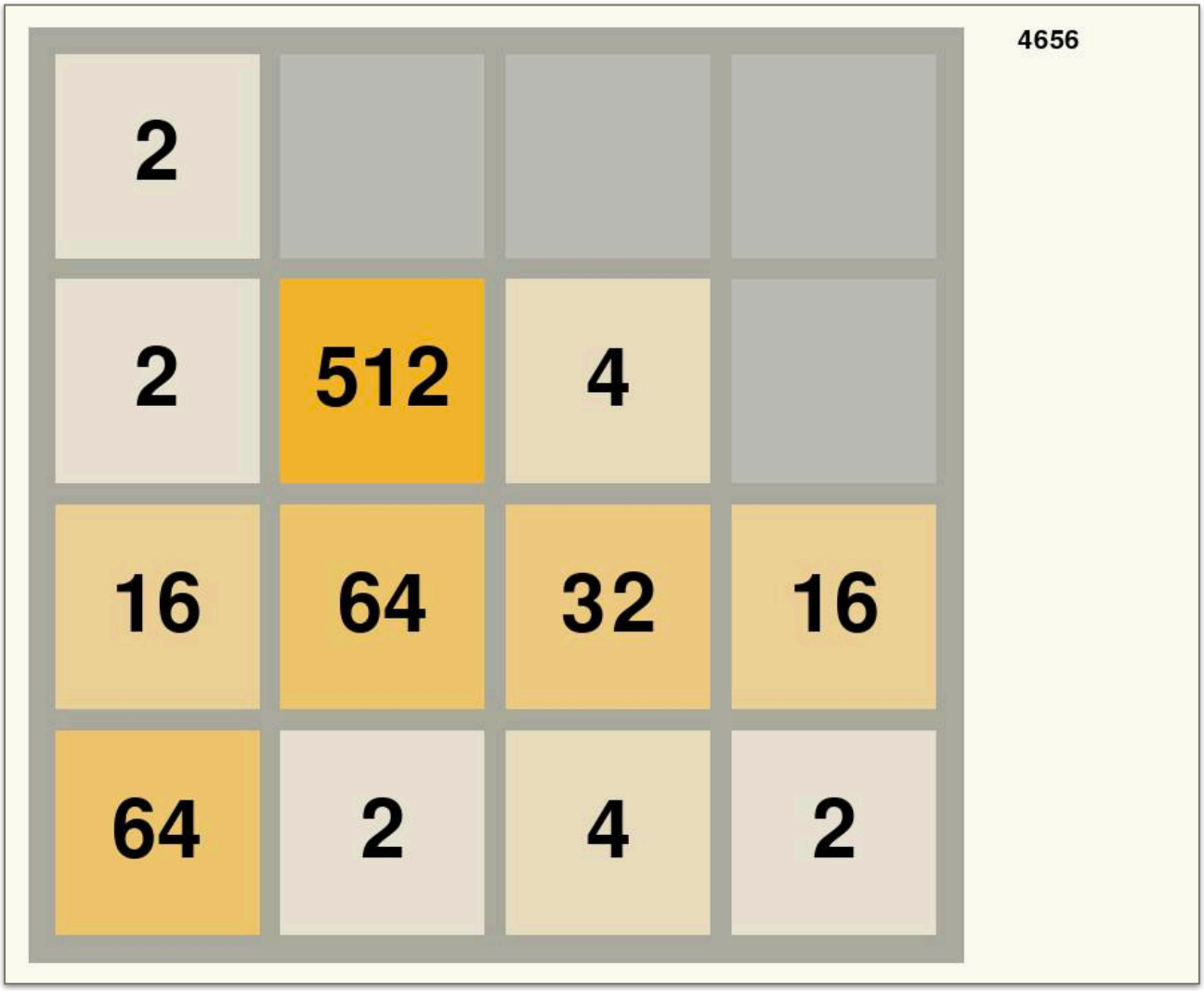
I used to play this with my friends on paper, but since in each round all applicable rules will be reapplied (see screenshot, where at the 3rd round all three rules are applied once), it become very difficult to simulate after a few rounds, without the "superhuman brain" the protagonist had in the

novel. Therefore, I made this game application using Javascript to automate the simulation process.

Utilizing object-oriented programming, I managed to dealt almost any arbitrary rule players can set by piling up components (see the bottom-left corner of the screenshot).

The source code can be accessed at https://gist.github.com/beanandbean/16ad5049c057bd412a423beb91c9e422

Python 2048



A PC 2048 game implemented in Python. I implemented the project in a modularized way, allowing me to not only manually play it, but also build various bots to automatically play the game and evaluative the difference between their proficiencies. After studying the game for a few weeks and playing it myself, I proposed a strategy, which I then implemented into the bot. On a traditionally 4x4 board, it was able to guarantee a 512, and

even reach 1024 if lucky, by merely following the rules I set as if-statements.

The front-end is completely implemented and rendered using Python and the Pygame library. The source code is at https://gist.github.com/beanandbean/e08c8a5256721aa4e5bf4cbbc7c71e13 where controller.py is the core module, user.py is the manual-play module, and maximum.py is the final bot utilizing the strategy I proposed.