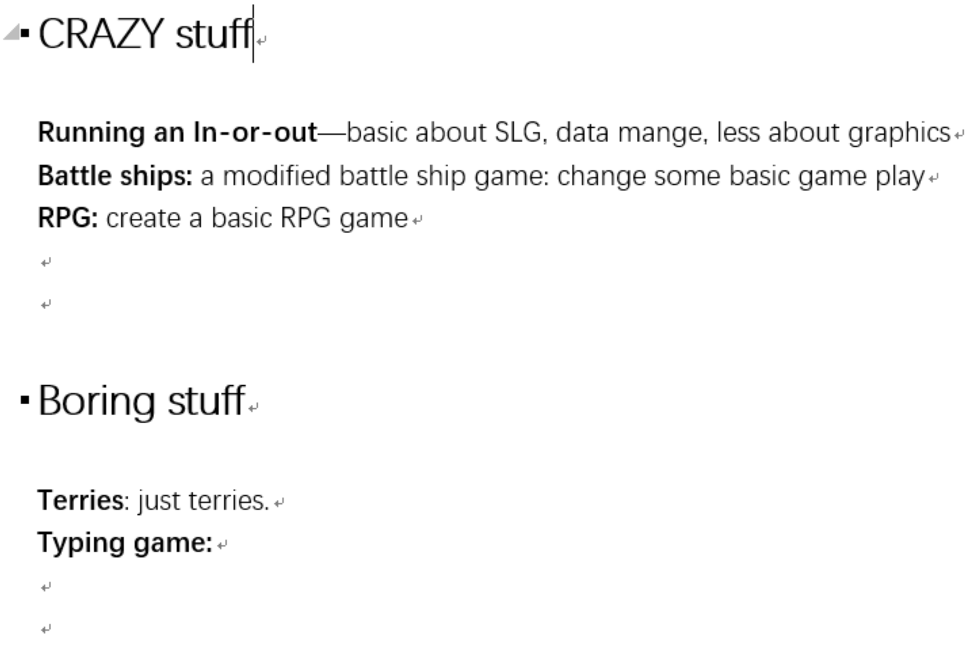
Postmortem document of MJ simulator

Before development

1. Brainstorming: the best part of the design of a game.

October 23 afternoon

At the very start of this project, the goal of the first meeting was to confirm the direction of the whole game. We were able to get together and brainstorm some ideas for the game. We made two lists of ideas: crazy ideas and boring ideas. The crazy ideas were all about what it meant: cool and innovative ideas but hard to develop. And the boring idea meant some basic gameplay like Tetris. The discussion didn't last long because we reached the idea about "simulator."



- CRAZY stuff

Running an In-or-out—basic about SLG, data mange, less about graphics
Battle ships: a modified battle ship game: change some basic game play
RPG: create a basic RPG game

- Boring stuff

Terries: just terries.
Typing game:

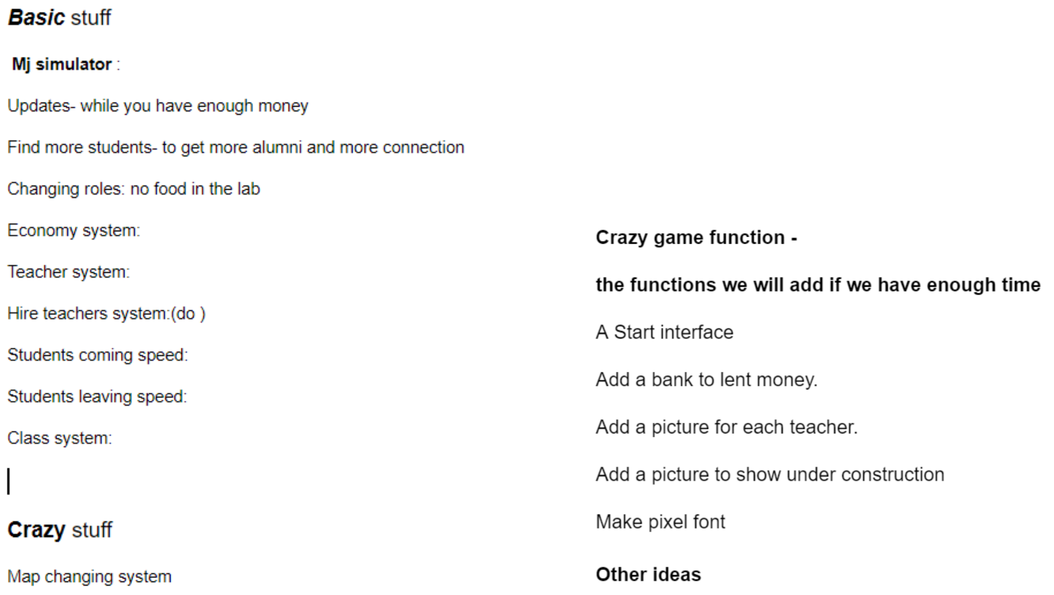
The ideas from brainstorming

The simulator was the top one on the crazy ideas, not only because it was data-based, but also it only needed little graphics. However, in the beginning, the idea about the simulator was about "Mcdonald's simulator" and something like that. The idea of <MJ simulator> suddenly came out, and it was so cool that everyone agreed that this was the one we were going to develop.

2. Ideas: the moment we are still adding functions instead of removing them

October 23 afternoon

Since the idea of the MJ simulator was confirmed, we moved on to the next part to design the system, function, and UI of the game. Just like the way we came up with <MJ simulator>, we listed “basic ideas” and “crazy ideas.”



Basic stuff

Mj simulator :

Updates- while you have enough money

Find more students- to get more alumni and more connection

Changing roles: no food in the lab

Economy system:

Teacher system:

Hire teachers system:(do)

Students coming speed:

Students leaving speed:

Class system:

Crazy stuff

Map changing system

Crazy game function -

the functions we will add if we have enough time

A Start interface

Add a bank to lent money.

Add a picture for each teacher.

Add a picture to show under construction

Make pixel font

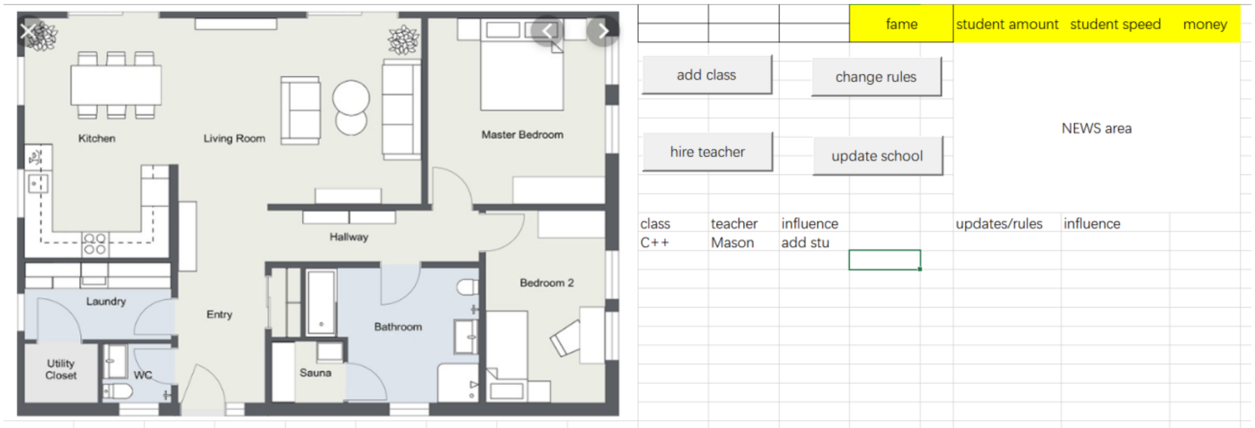
Other ideas

System design document(brainstorming version)

The basic gameplay is like a “school simulator” where players need to run the UCSC Silicon Valley campus and deal with the balance between funding, students, teachers. Just like other simulation games, resources are always the key to all systems. The resources were quickly settled due to the fact that it’s a school simulation game. Money, number of students and teachers, and incoming and leaving student speeds were in the first version of resources systems. You may notice that there were many differences between the first version with the final version. Also, it contained a lot of systems that we removed later. If you want to check them out, they are in the “Others” part.

3. Design and Discussion: the game seems totally different from the final version

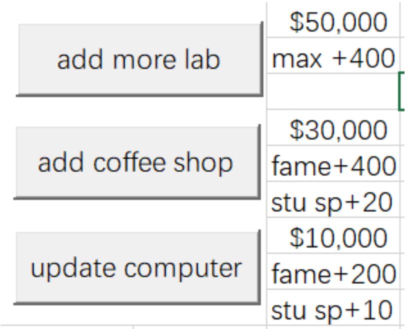
October 25 afternoon



Basic game interface (alpha version)

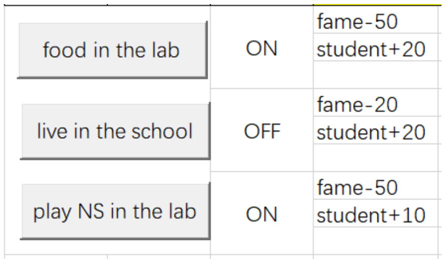
Brief introduction of alpha version:

The picture on the left part of the whole screen shows exactly what the campus looks like in the plan. The plan will change after the player updates the school. As you can see, we didn't have the "grab and generate classroom" function at the beginning of development because we wanted to control the plan and make graphics beautiful. Under circumstances like this, the classroom's location is designed and controlled by the designer. So add class has a similar visual effect with update school.



add more lab	$50,000 max +400
add coffee shop	$30,000 fame+400 stu sp+20
update computer	$10,000 fame+200 stu sp+10

Update school UI alpha version



food in the lab	ON	fame-50 student+20
live in the school	OFF	fame-20 student+20
play NS in the lab	ON	fame-50 student+10

Changing rules UI alpha version

The "changing rules function" allows players to change some basic rules of the campus in order to trigger more random events and make more positive feedbacks in the game.

The interface part is at the right of the screen. The state bar (yellow part) is at the top of the screen, showing resources the player has, like fame/student amount/student changing speed/money.

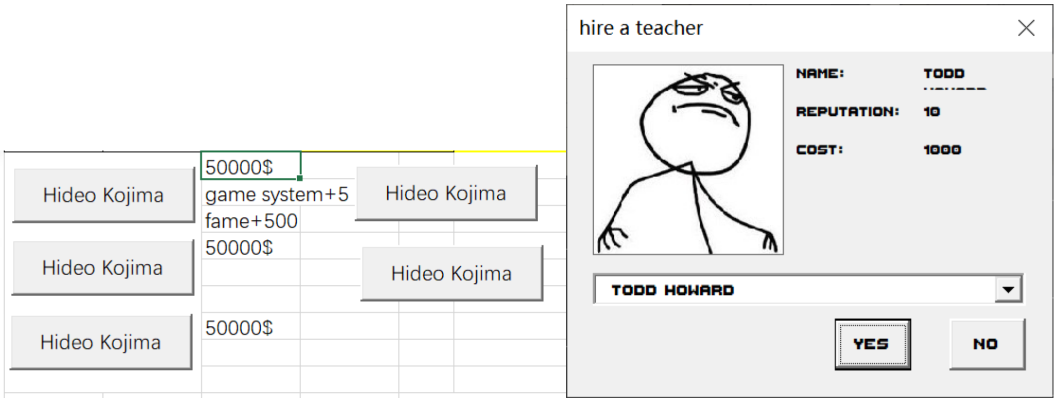
The NEWS area is on the right. The news will show up when the game reaches a specific point. For instance, when the school is more famous than EA, there will be news to write about it. The basic idea is to make more changeable things into the game.

At the bottom, there is a place where the players can see all the changes they made in one game. It will show the positive and negative moves they make.

During development

1. Systems and Functions: everything starts becoming complicated

1) Hire teacher function: the "hire teacher function" is a very basic function that suffers the smallest modification during the entire development. As you can see, the original version of the "hire teacher" function has the ability to match each class the player added. If the player matches the teacher and the class successfully, there will be a bonus effect on the class. Unfortunately, we removed this function in the final version, not only because this function is hard to develop, but also the "hire teacher function" is a very basic function that the player will use at the beginning of the game, so it shouldn't be too complicated.

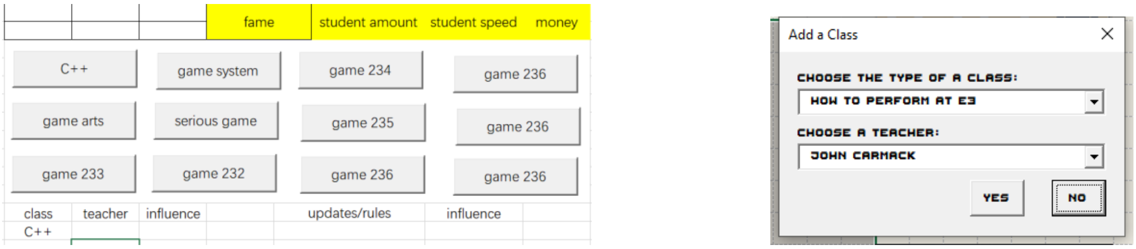


50000$
game system+5
fame+500
50000$
50000$

NAME: TODD
REPUTATION: 10
COST: 1000
TODD HOWARD

The evolution of the "hire teacher" function. The right one is the final version

2) Add class function: "add class" functionis totally different from the original design. The original design will allow players to add a classroom in a specific location, but we added the "drag and generate classroom" function to increase the gameplay and use some function that the spreadsheet already had. Actually, there was some conflict between the "drag generate classroom" and "fixed room" function design. As you can see, we kept two functions together: the "drag and generate classroom" for adding a class function and "fixed room" for the entertainment room only.



fame	student amount	student speed	money

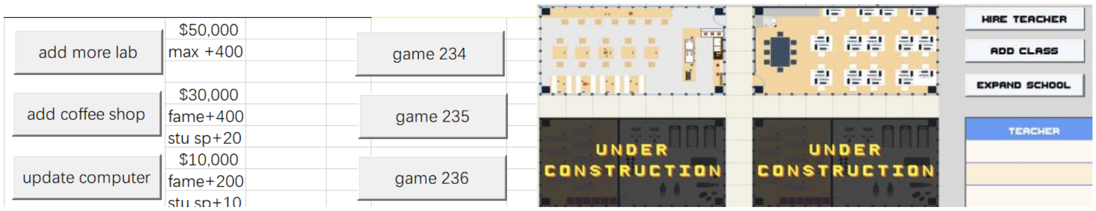
class	teacher	influence	updates/rules	influence
C++				

CHOOSE THE TYPE OF A CLASS: HOW TO PERFORM AT E3
CHOOSE A TEACHER: JOHN CARMACK

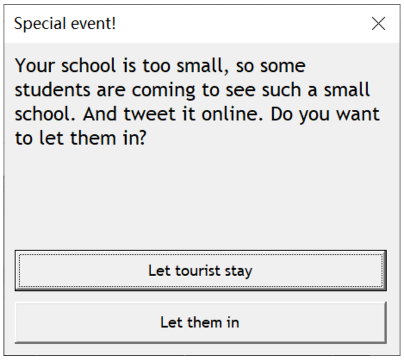
The evolution of add class function. The right one is the final version

3) Expand school function: "expand school" function gets the biggest attention in all functions. Not only it occupies half of the screen, but also, it triggers the end game situation. We put a lot of functions and data into the "expand school" function. However, it turns out that it's really hard to make those accomplishments in the spreadsheet. So,

instead of making functions, the "expand school" function became like a timeline to know the part the player reaches.



4) Random events: random events are the key function in the game. While designing the game, we found out that the game was kind of boring (there were no bulletin board and MJ's diary yet). The game was all about hiring teachers, adding classes, and watching numbers grow. That's the reason we added the "random events"function. With the game continues, there will be a lot of triggers to trig different kinds of random events. Random events will have a choice system to let players make some tough choices. After added random events system we realize that there are no negative feedback in the game. This means the funding will keep growing and it will be a boring game. We create reputation system in order to make the negative feedback in the game.


Special event!

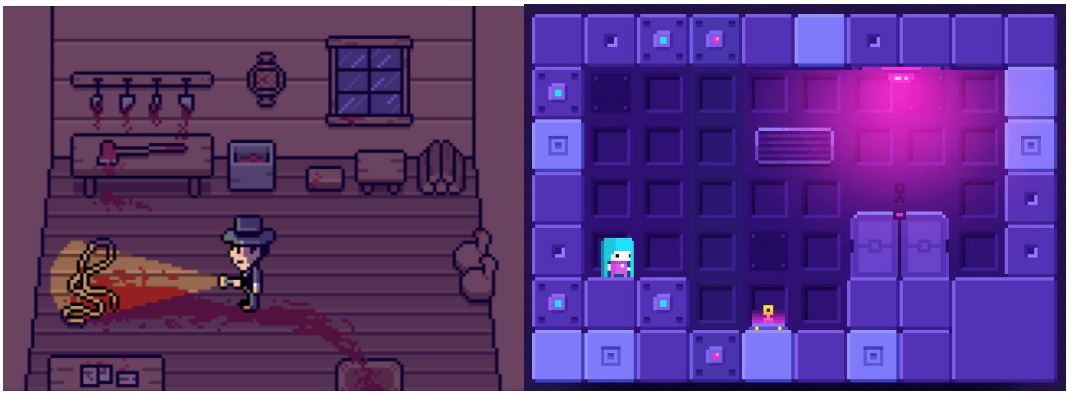
Your school is too small, so some students are coming to see such a small school. And tweet it online. Do you want to let them in?

Random event

5) Reputation system: the reputation system will have significant influence on the student coming speed in the very early game development. And players are able to get reputation in many ways. The final version of the reputation system is all about end game check and a system to judge whether the player makes too many awful choices.

2. Visual Design: daily scenario-“Can we put this fancy animation in the spreadsheet?” “Oh, forget it.”

For the visual design, pixel art was the first idea that popped into our heads. Not only because the retro style suits well for a game based on a spreadsheet (like a cold weapon in the modern world), but also because it would save us a fair amount of RAM in operating the whole system. Therefore, the initial interface, the tutorials, and the expression of the storyline are all based on pixel art.



(Case study of pixel art model)

The plan of the school is the most important part in our project. We have brainstormed a couple of ideas. The first one was like a fancy 3D model, in which the size and decoration of the building would be changing with the expansion of the school. This was the one we loved the most, except the programmer who had the foresight said this could never work well in a spreadsheet. We strongly agreed with it after we encountered a few limitations during this project. As a compromise, we also considered expressing the first idea with a changeable 2D plan, which was mentioned in the previous paragraph.

YOUR RESTAURANT
YOUR RESTAURANT GROWS AND IMPROVES

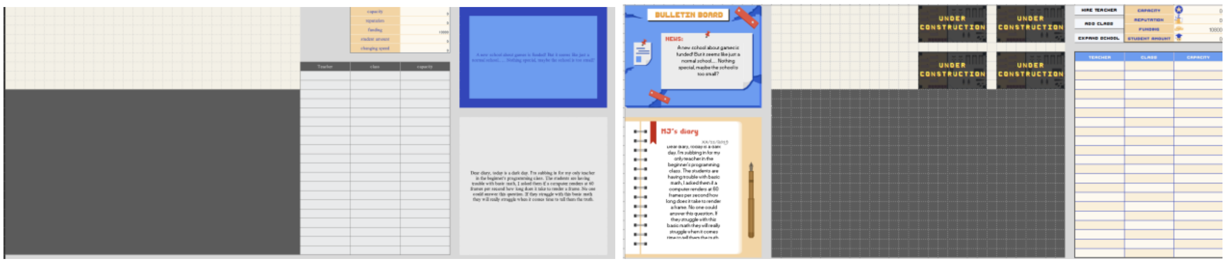
(The expectation of 3D model)

However, we finally realized that the feature of a spreadsheet was about the modular and the repeatability. Thus, we decided to make a repeatable classroom modular where players can customize the size of it and the location where they would like to put. We also designed other types of rooms in the right half part of the plan to make it more diverse.

Feedback from playtesters showed that the visual design of the storyline part (the bulletin board and the diary) was easily ignored. Also, more tutorials to guide players were also required. We revised them with the following changes:

1. Put the storyline part on the left side to make it more notable.
2. Add titles of the bulletin board and the diary separately to make the theme clearer, like NEWS and MJ’s diary.
3. Leave more space for the content of the storyline and use a font that is easier to read.

4. Draw players a basic picture of the rules and the bonus with a step-by-step tutorial. However, given the time limitation and the solid structure of our program, we can't put each tutorial right after the exact operation of the player. Instead, we put it in the beginning of the game like a rulebook.



A new school about games is funded! But it seems like just a normal school.... Nothing special, maybe the school is too small?

Dear diary, today is a dark day. I'm subbing in for my only teacher in the beginner's programming class. The students are having trouble with basic math, I asked them if a computer renders at 60 frames per second how long does it take to render a frame. No one could answer this question. If they struggle with this basic math they will really struggle when it comes time to tell them the truth.

(The layout: first version vs final version)

3. Economy system: the most excruciating part

All right, finally about the data. We wanted a 10-minute game because it's an ideal length for playtesting. Also, the player will update the school twice because of the cube design.
So, the game experience will be divided into three parts:
1. Beginning to the first update
2. First update to the second update
3. Second to end game

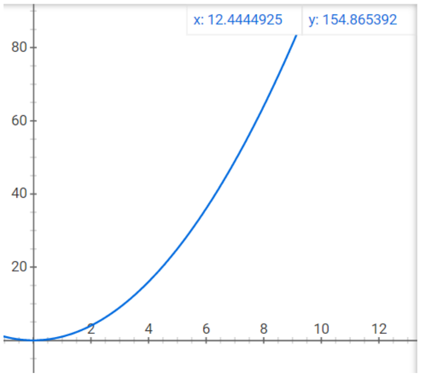
So, the game time will be: 3.1.1
This means 6 minutes, 2 minutes, and 2 minutes for each part of the gameplay.
The price of each update will be rising, like the first will be 1K, and the second will be 10K. Some kind of rising like that.

The first data need to confirm is the cost of the end game update.

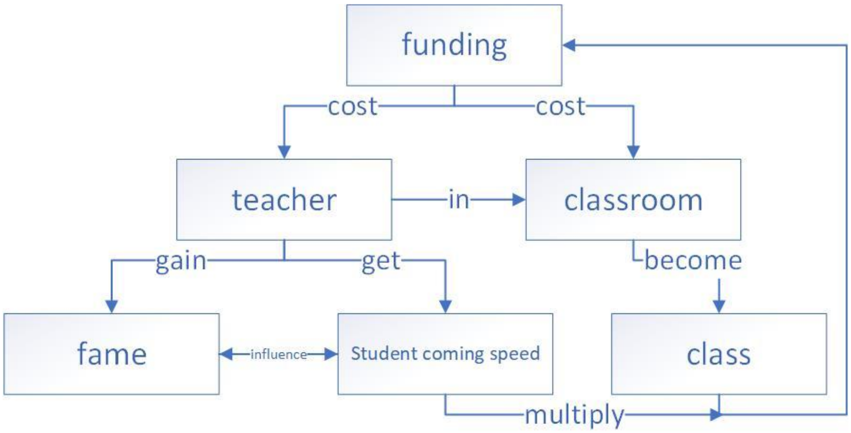
So the end game update will be 1,000,000. Because it's reasonable and easy to calculate.
The second update will be 500,000
The final update will be 100,000
The beginning will have 10,000 - 10K

If players want to build a normal classroom, the price will be 100 for each cube.
The beginning funding will limit the player to build a classroom no more than 10*10. But the beginning space will only have 9*10, so the first teacher will be 1000 to hire. That also means that the price to hire a teacher will begin at 1000. And the classroom's price will be the same.
And the changing speed unit is student number/day.
For each cube, capacity will be 10**,** so a 9*10 class will able to contain 900 students. So the max amount of students will be 3780(theoretically). Game time will be: 2 seconds in real life is 1day in-game. So the total game time will be: 5*60 = 300 seconds - which is 150days

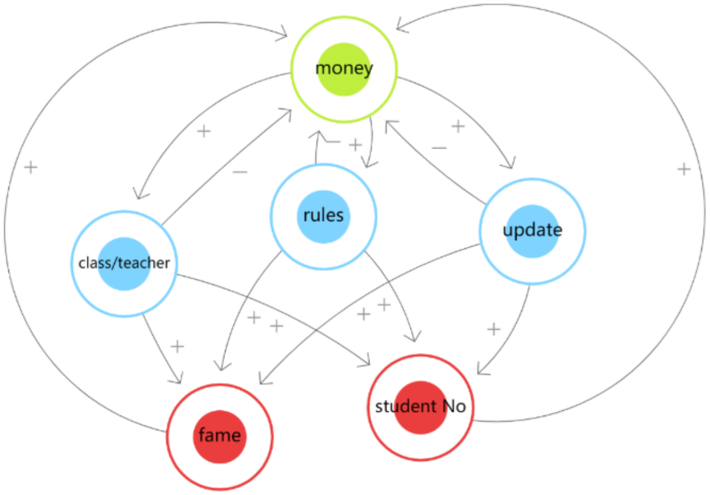
The time/ money graphic will be the picture below: like y=x^2. The X-axis means game time and the Y-axis means money.



Because the more player played, the more he will know about the game, so we want the late-game time to be as little as possible.
This means we need a number who share the same changing rate with “student changing speed”. The number of classes fits this demand perfectly.
That gives the money: money = student adding speed*100 * class number(as the picture shows below)



The game economy



The original economy loop

The final economy loop: https://bit.ly/36AcohI

So the data will be like:
reputation= hiring teacher +special event
Student amount = student changing speed * time
Funding = Funding + student adding speed * 100 * class number+100
Student adding speed = hiring teacher + special event

Cube design
The whole page of the game is 27*41.
The interactive part of the campus is 27*30.
And we definitely want to divide it equally.
So the part for the player to drag and generate classroom will be 27*15
And the part for design entertainment room is 27*15
The player will have a 9*30 space at the beginning of the game.
Each update will give player another 9*30 space
When the player update school twice, the school can't be updated.
So the maximum size of the school is 27*30.

4.storyline: the part where words don't make sense anymore

October 31

For the storyline, it was difficult to come up with an angle because there were so many possibilities. Three of them were:

- M.J. was an industry veteran who saw the rise of many great games. As he works on a new title he senses a disturbance in the universe's gameplay loop. To bring reality back to a stable state M.J. must train new game designers to bring the universe back to a balance.
- M.J. has done it all. He has surpassed what any mortal/human game designer is capable of. Hungry for a new design challenge, M.J. sets out to tackle his biggest design challenge, creating the perfect education system. With no financial support and completely clueless students, how will M.J. fare?
- M.J. figures out that he has been living in a spreadsheet game his whole life. It all makes sense now, the seemingly random election results, global warming, and full city power outages. Armed with only the ability to add and subtract resources from the gameplay loop, M.J. must figure out a way to break the game. He decides he can do it with the ultimate rng variable master students.

We liked the last one because it was funny and it reminded us of a video (https://www.youtube.com/watch?v=VufDd-QL1c0). While coming up with the storyline and diary entries, we were inspired by real-life events that we have experienced during our time at this school. At first, our story was linear, and there was only one ending. At first, the ending seemed like a bad ending, so we rewrote it to make it happier. After doing some playtesting, we decided to make the story have multiple endings (one good and one bad).

5.coding: VBA Programming: from beginning to giving up

We have used excel as a database to save data and designed different modules to load the data and calculate for different functions. In this case, the most difficult thing is that we need to learn a new programming language. However, we did it.

6.sound effect: the fun part where making weird noises is acceptable

October 28 at night

We were inspired by the *Hidden Folks* soundtrack to create our sound effects because we thought it was unique and hilarious. They were mouth sounds created by one person. We even bought the soundtrack to use in our game. Also, we recorded ourselves making noises that seemed to fit the situation and to give feedback to the player when they do an action. It was awkward but an interesting experience because we have never done it before. We also created background music, but it was cut in the final draft.

After development

1.playtest inside: useful somehow, but not so useful

We playtest the game a lot while we developing the game. But there are a lot of boundaries while we playtest. Like we know our game too much, we know every system so well that have no regular game experience that normal player has. And every design is designed for a reason, which makes it really hard for us to find the flaw of the game.

2.playtest by others: the thing we should do before the game was finished (which is impossible)

We finish the first version of the game and find someone to playtest it. There is some feedback like: change the font, change the entire UI, and the economy system. We make a list of all the changes need to update and line them by difficulties to see which is easy to update.

The first thing we changed is our font. There are a lot of suggestions about our font, we design the whole art style as pixel, so we find a pixel font and use it in the game. But it doesn't work well in the random event. We find that it's extremely hard to read the words if we use that font. So we change the font on the board and random events.

The storyline is one of the important parts of the game. During the playtest, we notice that it's really hard to let players notice there was a storyline and the "MJ's dairy" will change. So we move the entire UI and put the storyline in the left part of the game.

Others

Link to the project

https://drive.google.com/drive/folders/1l7DSSNpSAP4Z6x2aXamWkaGIbzZu8szF?usp=sharing