

Basic Information

Project Name	Kung-fu Plants
Supervisor	Jakub Gemrot
Annotation	Kung-fu plants is a game where you can create your characters, train them and fight against the other chemical plants which want to destroy your field. It requires strategic thinking to play and contains skill-based combats against the enemy characters.
Team Members	Federico Forti, Erdi Izgi, Varalika Rathore, Francesco Forti

Motivation

Today the mobile game industry is a large part of all the mobile application market. The most successful mobile games share some common characteristic like: good design, structure, polishing and artistic content. Our objective is to build a game based on those characteristic. The team will also collaborate closely with a graphic designer for the creative content creation. Working in a collaborative way will be another objective of this project. The content creation is usually one of the most critical part in a game creation process, since we want to make the game fast and smooth.

The most similar game to our game is Plants vs. Zombies which we also inspired by. Our designer was also inspired by the mini games inside the Lumosity. Although, both of two games have had a huge success, we want to give more chance to the player for character personalizing. Creating a character, training it and fighting using this character will make the player more attached to it. This is the main motivation that makes our game project promising and challenging.

The game can be designed with a modular way and this will make the process divisible into smaller iterations. Since we are a small team, using agile software development techniques will fit to the project. We will meet with our supervisor every two weeks and we will check our milestones. That's why the idea and technical tasks are suitable to do during the Software Project process and will be instructive for our team.

Project Description

Game Description

Kung-fu Plants is a strategy and fighting game where you can create your characters, train them and battle with the evil-chemical plants. So the game is based on a set of characters that the player creates during the game. Those characters are plants and they are used in the three different phases;

Seeding

In this phase, the player uses the “seeds” obtained during the game to grow up new plants. After planting a seed, the player needs to take care of it. For that reason, the player needs to check the health of it after a specific time and he needs to feed it with the right item. At the end, the seed will transform into a character based on what the player has done and it will be usable in the other phases. If the player has fed the seed in the right way, he gets a strong character otherwise a weak character.

Training

The training phase is used to teach new techniques to some of the player's characters. If the player decides to train a character, it will be unavailable in the other phases for a specific period of time. When the player assigns a character to a technique, this will unlock a set of game rounds. The difficulty of the game is based on some characteristics of the character. For example, if the game is based on the agility of the character, the speed of the game will be based on that characteristic. The more the player plays the game the more those characteristics are increasing. However, in order to get the technique, the player has to pass a “final exam” that means playing the specific game at the end of the training period. Therefore, if the player keeps playing during this period it will be easier for him to pass the final exam.

Fighting

In the fighting phase, you can use the characters that you have grown up and trained during the other two phases. The player has to choose 4 characters among the others before starting the fight. In this phase, you are fighting against the other plants. The objective of these plants is to destroy the field where you are. The objective of the player is to stop them before that they destroy all the field or they

arrive towards him.

The field is circular and is divided into sectors. The enemies are spawned at the edge of this circle and each of them is assigned to a specific sector to destroy. The player is positioned at the center of this circle. When the enemies destroy completely one sector, they arrive towards the player. This leads the player to lose.

In order to defend the battlefield, the player has to send his character to fight in the sectors. The player has to move between the sectors to check the situation there. The player can send one of his 4 characters or recall one of them to send him to another sector.

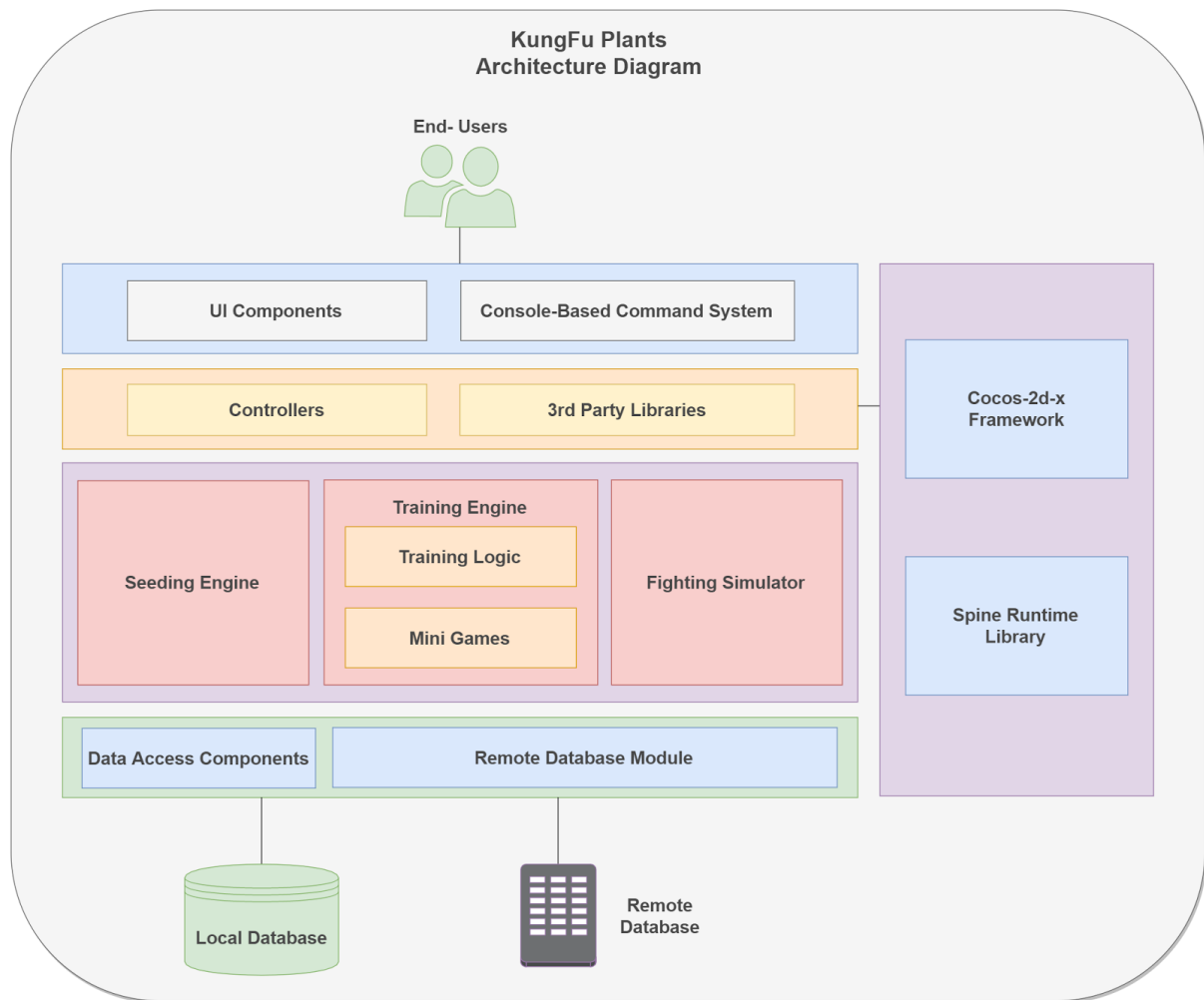
In this basic mode the characters can fight autonomously and if they die during the battle the player has to get them and recover them before reusing. However, there is another possible interaction for the player. If the user decide to concentrate on one of his characters that is fighting, he can see the statistics of the character and use its techniques. Every character has his own techniques that are learned during the training phase and every technique cannot be used for a certain amount of time after that it has been used.

Every level has a certain number and types of enemy according to the difficulty of the level. When the player passes a level, he can obtain seeds to grow new plants or items to feed them.

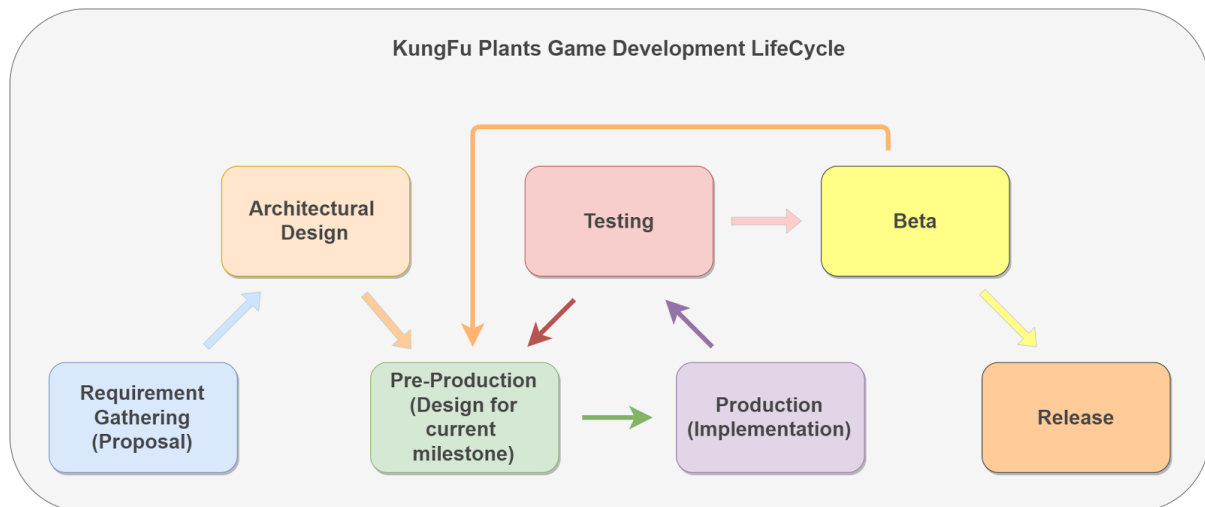
Average gameplay time for the fighting level is defined as 3 minutes at first. Since the game is planned as 7-10 levels, we can promise at least 20 minutes of gameplay time excluding the seeding and training phases since these are passive. After the game play tests, this numbers can change in accordance with the user test results.

Technical Details

The project mainly contains 6 major modules such as; User Interface, Seeding Engine, Training Logic Module, Mini Games Module, Fighting Simulator and Networking module. User Interface and Networking module depend on all the other modules, and Training Logic Module and Mini Games module are also interdependent. Other modules are independent of each other.



The project will be implemented modularly depending on the third party libraries, runtimes, and frameworks. However, creating a polished game needs lots of iterations to keep the game balanced and enjoyable. This means that all modules will have a change management process in accordance with the test results. Thus, we will follow the inception, elaboration, construction and transition phases (testing and change management) in an iterative manner for all the modules.

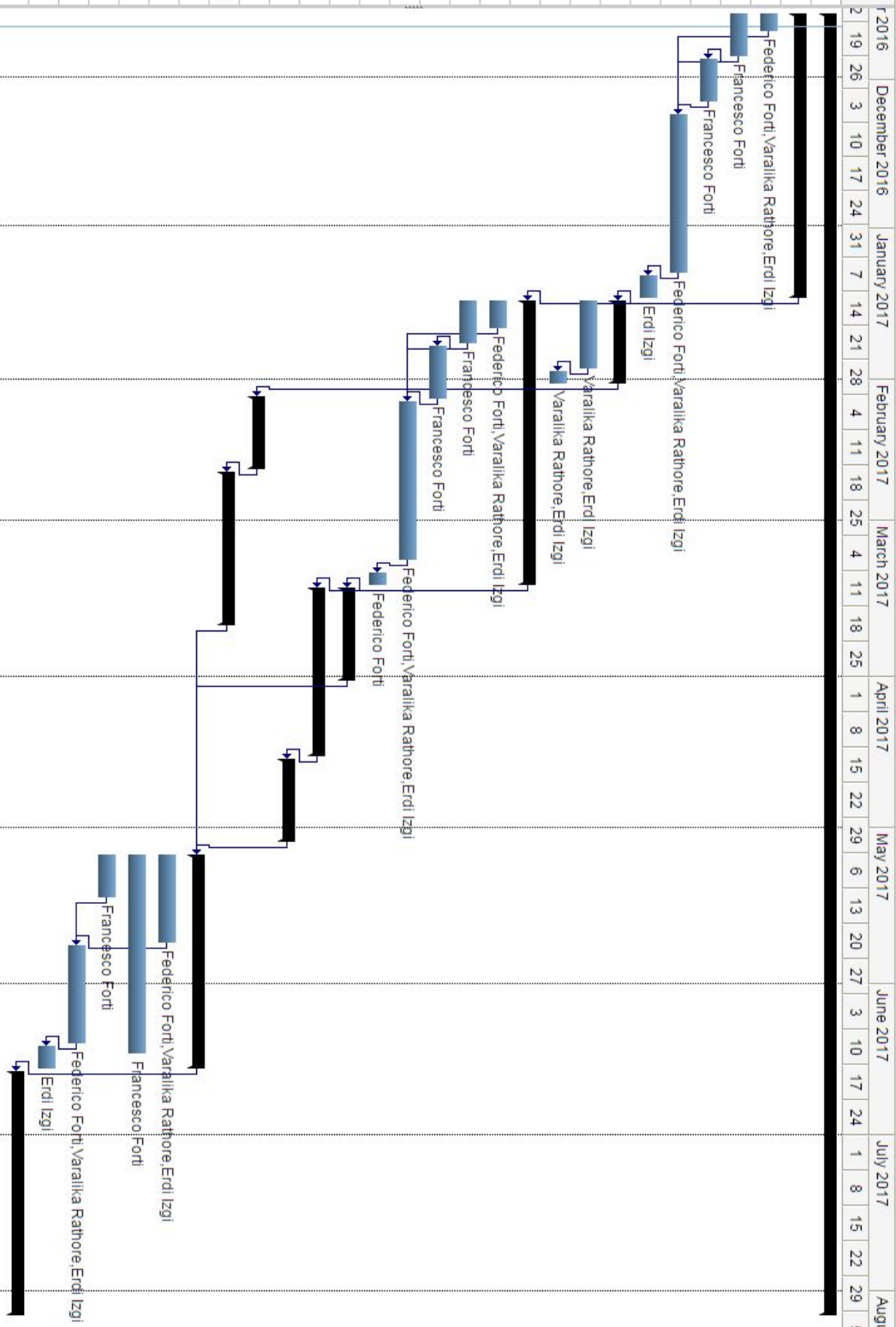


The testing phase of the project is the longest among all the phases. Creating a good polished game requires a lot of testing, we can say that if we finish a first working beta in 30% of the time all the rest is necessary for testing and re-implementing. Therefore, the structure of the project must be dynamic, we must use good design patterns to permit as few change as possible in the code. Just creating on our initial idea it's not enough for our scope, we need also the feedback from the people and when something is not working as we planned we must change and start over again.

It is accepted that creating a fun game requires many iterations. These iterations are a little bit different than other software development lifecycle. Applying tests just for the functional requirements is not enough to decide whether the game is fun or not. That's why we need so many prototypes. Prototypes should be prepared as fast as possible and correspond minimum requirements to play the game or a game part. There should be also a goal to do it. It should be parallel with the game design and answer some questions that the team are curious about?

The game will have 4 prototyping processes (see *Planned Prototypes* section for the details of the prototypes). Since the game has 3 different phases, a targeted prototype will be created for those separately. Related tests and playing experiments will be made, then using those results of the tests, a new prototype will be made. This prototype will include all the phases together, so the team will be able to test the whole idea and made improvements and finalize it for the release.

		r 2016	December 2016				January 2017				February 2017				March 2017				April 2017			May 2017			June 2017			July 2017			Augu										
		2	19	26	3	10	17	24	31	7	14	21	28	4	11	18	25	4	11	18	25	1	8	15	22	29	6	13	20	27	3	10	17	24	1	8	15	22	29	5	
1	📌 KungFuPlants																																								
2	📌 Prototype Fighting																																								
3	Software and Architecture Design																																								
4	Placeholder Graphics																																								
5	UI and UX Design																																								
6	Implementation																																								
7	Code Testing																																								
8	📌 Play Testing Prototype Fighting																																								
9	👤 Experiments with Players																																								
10	📌 Test Report																																								
11	📌 Prototype Seeding																																								
12	👤 Software and Architecture Design																																								
13	Placeholder Graphics																																								
14	UI and UX Design																																								
15	👤 Implementation																																								
16	👤 Code Testing																																								
17	📌 Play Testing Prototype Seeding																																								
20	📌 Prototype Training																																								
28	📌 Play Testing Prototype Training																																								
31	📌 Iteration for Prototype Fighting																																								
35	📌 Play Testing Iteration 1 Prototype Fighting																																								
38	📌 Final Prototype																																								
39	Design Change																																								
40	👤 Final Animations and Graphics																																								
41	👤 Complete UI and UX Design																																								
42	Implementation																																								
43	Code Testing																																								
44	📌 Iteration on the Final Prototype																																								



The planned iterations and the required time can be easily seen on the Gantt chart. The estimations have been made including other university responsibilities and other classes. Accumulation of the numbers are;

	ManDays	ManMonths
Federico Forti	184	7,36
Francesco Forti	97	3,88
Varalika Rathore	172	6,88
Erdi Izgi	181	7,24

Like every project, it may require some additional works. Iterations might not be finished on the exact dates we specified there. So the rest of the time will be separated for the finalizing the projects. During the last phase, additional features may be required, the code will be optimized, major bugs will be fixed, necessary documentation will be created. Project will be ready for the presentation.

Testing is one of the most important part of this project. Therefore, the required time will be given to it. There will be two testing period at the end of the iterations; code testing and play testing. Importance of the code testing will more important after creating the last prototype to keep the game more robust. However, play testing will be significant during all the iterations. It is planned to separate 2 weeks for the play testing. This time seems long but most of the tasks can be done in parallel with play testing. So this won't block the team for the other task.

The first thing that the player interact is the UI. That's why a great amount of time will be spent on this regular module in almost every iteration. Creating a fluid, hands on and easy to control UI is one of the key factors of the game. Besides, eliminating graphics discrepancy and maintaining a seamless graphics transition all the way through is needed another iterative phase between designer and developers.

The game contains a lot of different characters and animations with them. Therefore, they have to be in different sizes respect to the device resolution. If we include all the characters inside the application, the size will increase considerably.

One of the the main point of our game is the 2D animation. For that reason we have decided to improve that part using a professional tool called Spine. The latter uses its own runtime and files for the animation but, for now, they don't provide an official release for the ios framework. For that reason we are going to use an open

source multi-platform game engine called Cocos2d-x commonly used in many mobile game companies. The latter works well with the 2D animation software Spine and give access to a vast api of functionality including sprites, actions, animations, particles, transitions, timers, events (touch, keyboard, accelerometer, mouse), sound, file IO, persistence, skeletal animations, 3D.

We want this game to be portable to other platforms and we want that a player is free to start playing in some device and continuing to another platform. For that reason we want to keep the state of the player in a remote server as an addition to a local one so that it can be retrieved in any moment. The state is composed of:

- Player history (passed levels, unlocked achievement ..)
- Inventory (seeds and items)
- Characters with their characteristics and techniques
- Player Information (log-in, demographic information ...)

The game should be playable offline and the state of the player will be updated after the player is online.

Platform, technologies, frameworks

Target Platform: iOS Mobile

Technologies and Frameworks: C++, Cocos2d-x

Animation Tool: Spine

Time estimation

Seeding Engine: Seeding engine is the garden of the player. Using the inventory, player puts together some of the ingredients and creates different characters with this recipes. Since every different recipe can affect the last form of the character, we need to add more character features depending on the seeds and ingredients, for that reason this module should be extendable.

Estimated time duration : 4 man-months

Training Logic Module: After a character creation completed inside the Seeding Engine, we can train the character to teach some special fighting techniques. This module will be interdependent with the Mini Games module. According to the result of the mini games the character features will be changed.

Estimated time duration : 3 man-months

Mini Games: This module will contain mini games of different types. After completing a game with a character, this module outputs data to the Training Logic Module and it will decide whether the character will acquire a new technique.

Estimated time duration : 3 man-months

Fighting Simulator: This module will simulate the fights between our characters and opponents. They will have some level of artificial intelligence according to the level design. Thus, it should be able to simulate the fight with different inputs as well.

Estimated time duration : 12 man-months

Level Design: We are going to deliver 7-10 levels. Every level has a circular battle area and it is divided into sectors. Since opponent's positioning, their types, their artificial intelligence level, and design of the battle area will affect the hardness of the battle, level design will be based on these features.

Estimated time duration : 3 man-months

Game Balance Corrections: The success of the project is based on the game balance. After creating the first playable prototype, there will be continuous user tests and we will adjust the balance in accordance to the test results.

Audio Programming: Audio part is restricted by the Public Licensed sound libraries for now. There is no special type of audio programming implementation. The required sounds will be acquired from audio content providers. Then there will be a sound mastering process to equalize the tones and volumes of all these sounds. After obtaining the sounds, there will be the implementation process for them.

Estimated time duration : 1 man-months

User Interface: User interface will be implemented using the standard libraries. All the graphics will be outsourced inside the UI by graphic designer. The team might separate time to create placeholder graphics instead of waiting for the actual graphics.

Estimated time duration : 3 man-months

Unit Tests: Every developer is expected to write unit tests regarding the classes they created. According to the process, code coverage can change but the critical functionalities surely will be covered.

Regression Testing: After the integration of the different modules or altering major implementations, there will be routine regression testings to keep everything

cohesive and working as intended. This will be done by the following approach. After every iteration, or bug-fixing period, all the depended test cases will be run again. Creating a regression test suite can consume lots of time, so instead of using detailed scenarios, we are going to run essential game play scenarios with the dependent unit tests.

Estimated time duration : 2 man-months

User Experience Testing: Every playable small version of the game will be tested by players and the team will collect data to make the game more fun and playable. At the end of every iteration, there will be a standard play testing period.

Planned Prototypes

Prototype Fighting:

This will be a simple prototype for the fighting module. This module should be tested first because it is the one that probably requires more iterations and improvements.

Planned Features

- Changing view between the characters in a circular way using swipe mechanic of the touch device
- Defeating the enemies simply by clicking on the character
- Some non-static enemies which try to destroy the field
- Ability to move the characters around the field
- Simple overall view for understanding how the enemies are approaching
- No special effects nor animations

Prototype Questions

- Does the swipe feature on the fighting screen really work when other sectors are in danger?
- Is the player able to move the characters easily between different sectors during the battle?
- Which are the difficulties in playing such a game like the pace of the game or controlling the character using touch mechanics?
- Is the overall view useful to control the game?
- Is the fighting fun and challenging, what should be changed to make it more playable?

Prototype Seeding

This prototype is going to be created for the mostly static module seeding. This process affects the quality of the characters. Thus, finding the balance in results of this phase is really important to make the game fun and sustainable.

Planned Features

- Some seeds and items should be already provided to the player to use
- Seeds should need related care such as sunlight and water
- When the processing time is finished for the seed growing, the player gets a character based on the used items.

Prototype Questions

- Is the caring of the seed fun and interactive enough?
- Can player easily predict the effect of an item?
- What difficulties can the player have while deciding to use items?
- Do player get the idea that the final character depends on the used items during the seeding process?
- Is the final character as it is expected?
- What possible improvements can be made on the balancing the expectation of the player?

Prototype Training

The training module is the module that the characters can improve their skills or get some special techniques. So this is the module that player can have improvements to enjoy the characters more.

Prototype Features

- 4 sketched ideas of minigames.
- The character should be assignable to the mini games.
- Hardness of the mini game should increase level during the gameplay
- All the training mini-games should have a duration and a final game is required to finish that training to obtain the technique.

Prototype Questions

- Is the training phase itself fun?
- Increasing the hardness of the mini game is enough to keep the player or what enhancements can be made on this idea of training?
- Which mini game ideas are more fun and what are the reasons for that?

Final Prototype

This is the final prototype that contains all the modules and there will be enhancements in accordance with the playtesting reports for the older prototypes.

Prototype Features

- Ability to perform a battle from the start to the end
- Some power-up items to help the character during the fight. This should require user interaction
- Differentiating the enemies and increasing difficulty of the battle while the final boss fight phase closing
- Four characters are ready to fight, however, one character should be able to use special techniques during the fight
- 2 mini games are enough for this prototype for the training
- The first basic animations should be visible
- Short time testing version for seeding
 - ❑ The seed must be almost matured.

- Long time testing version for seeding:
 - ❑ A variety of seed should be available for the player and can be planted in the garden.
 - ❑ Player should take care of the seed during all the seeding process.
 - ❑ Time related feeding care system should be integrated with notification system of the device
- Training mode should be available in an easy mode for the created character
- Different parameters should be able to test features such as the speed of the game or the strength of the enemies.

Prototype Questions

- Same question with the other prototypes? Are those now fulfilled?
- Are the Items useful for the player or is the game getting more complicated and incoherent?
- Which interactions are more fun? should those be polluted in different ways?
- Is it too hard or too simple to finish a battle in different circumstances?
- Caring for the plants should be more detailed or less detailed?
- Do mini games interrupt the player from the main goal of the game?
- Is the fighting module is feasible enough to play with different paces?
- How many times the user is opening the game? What enhancements can be made to increase the frequency of play times?

Project Scope

Discrete models and algorithms	
	discrete mathematics and algorithms
	geometry and mathematics structures in computer science
	optimizations
Theoretical computer science	
	theoretical computer science
Software and data engineering	
x	software engineering
x	software development
	web engineering
	database systems
	analysis and processing of large data sets
Software systems	
	system programming
	reliable systems
	performance systems
Mathematical linguistic	
	computer and formal linguistic
	statistical methods and machine learning in computer linguistic
Artificial intelligence	
	intelligent agents
	machine learning
	robotics
Computer graphics and computer games development	
x	computer graphics
x	computer games development

PART 1 PLANT

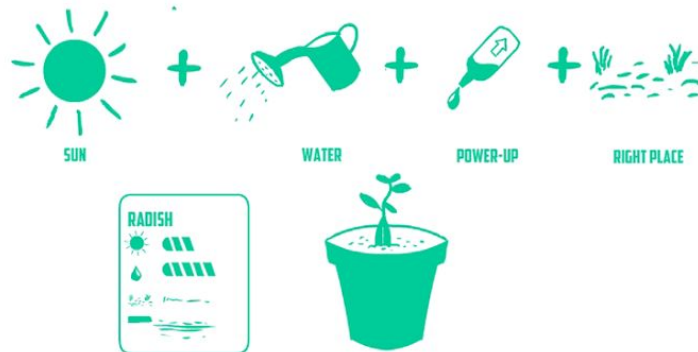
1. BUY OR DISCOVER SEEDS



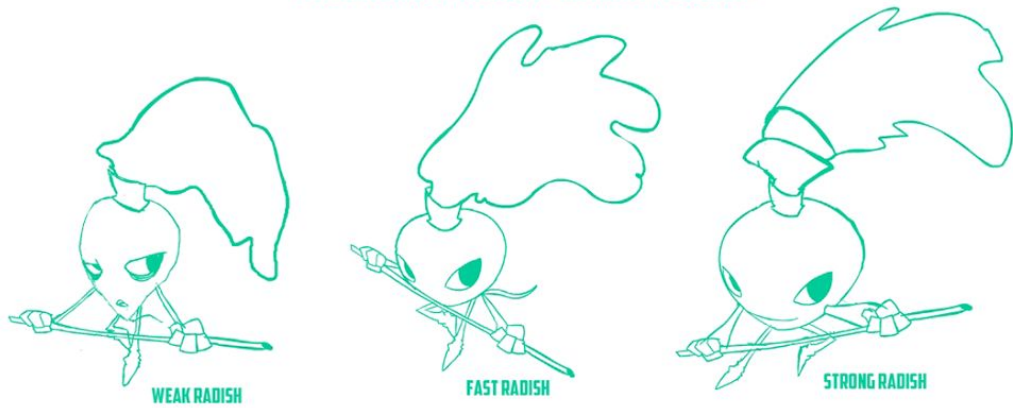
2. PLANT



3. MAKE IT GROW

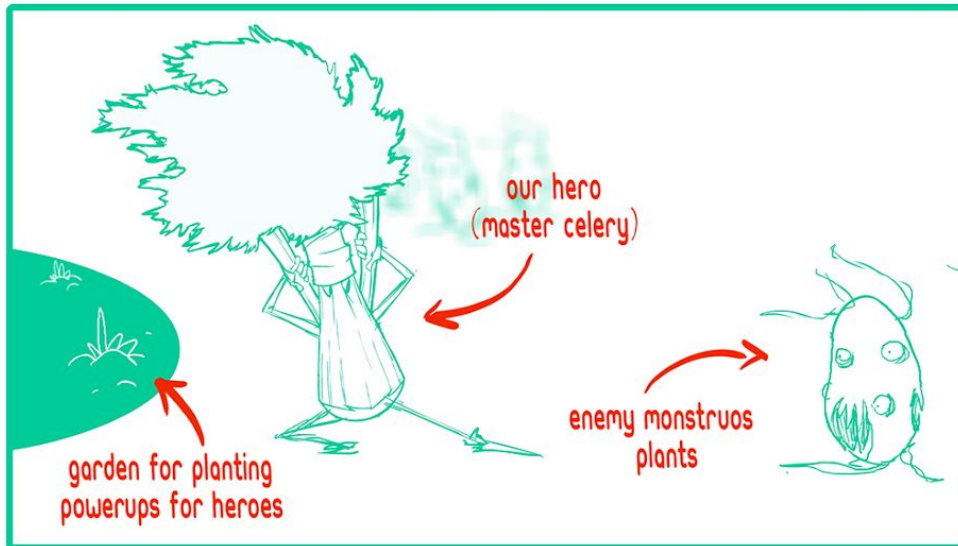


4. SEIZE THE PLANT DIFFERENT CARES, DIFFERENT RESULTS



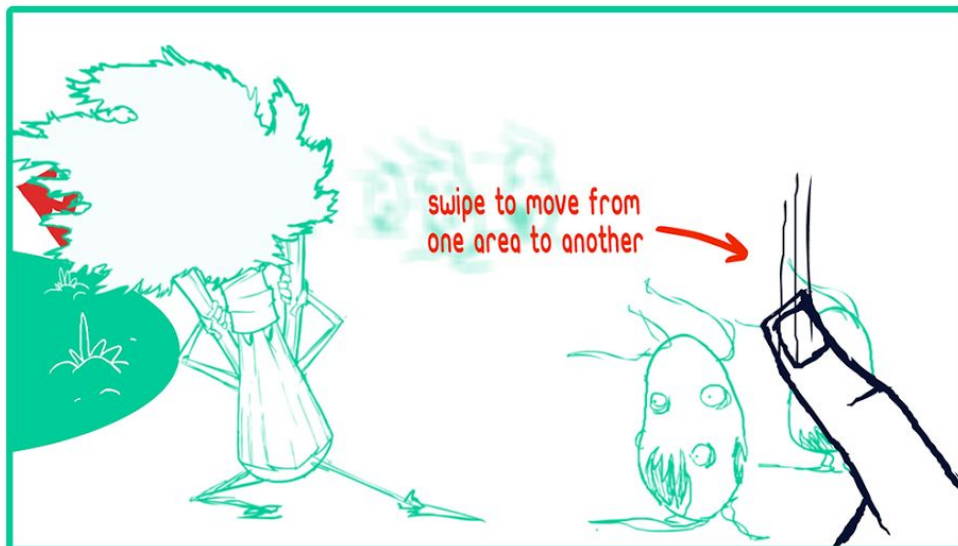
PART 2

FIGHTING



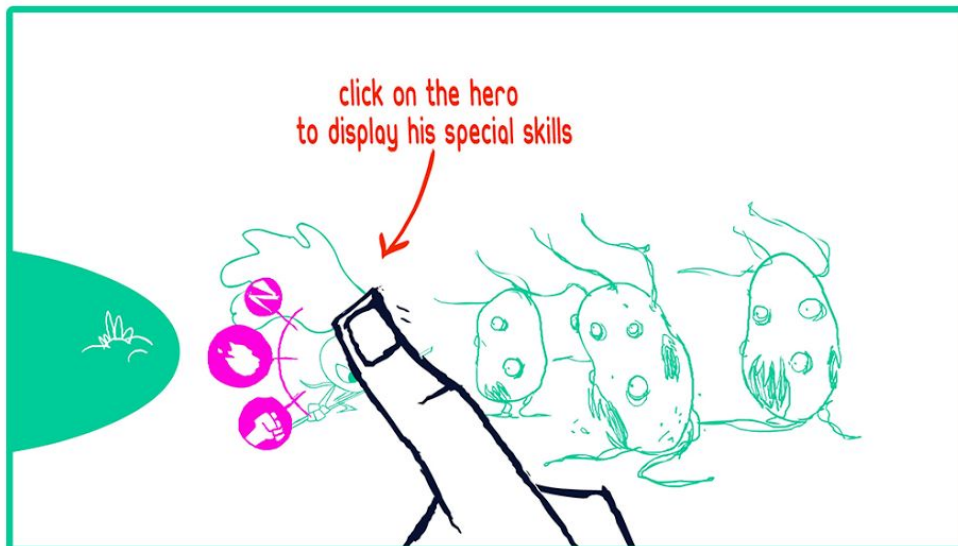
PART 2

FIGHTING



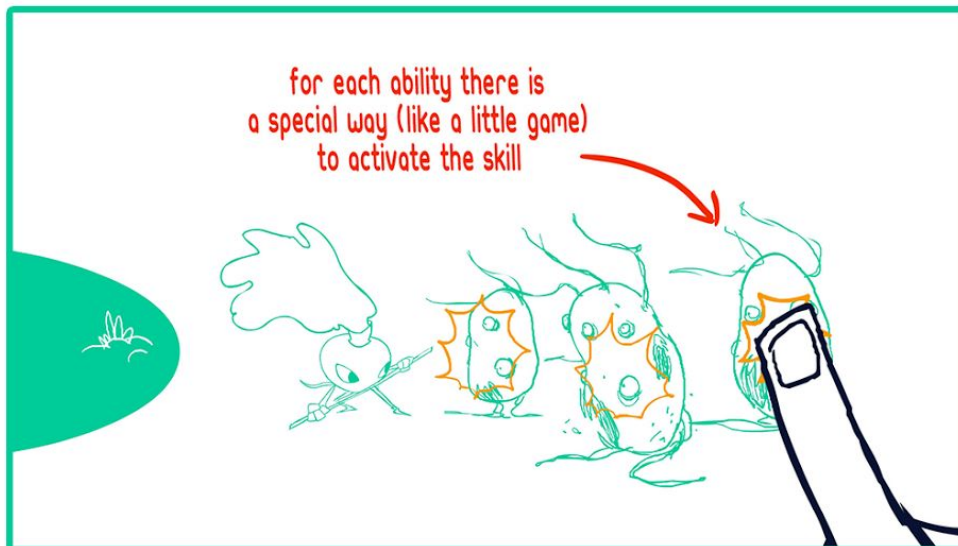
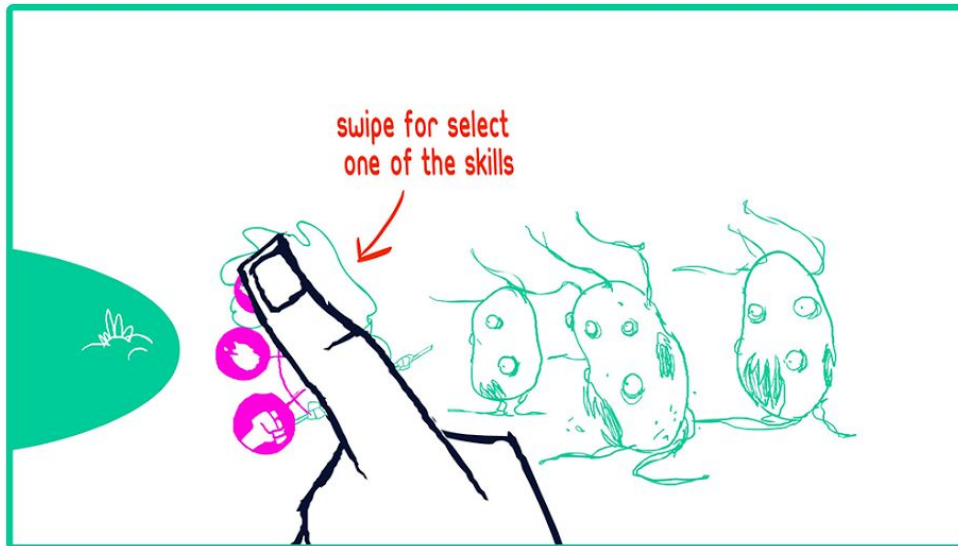
PART 2

FIGHTING



PART 2

FIGHTING



PART 2

FIGHTING

5



PART 3

TRAINING

1. SEND TO TRAIN A PLANT



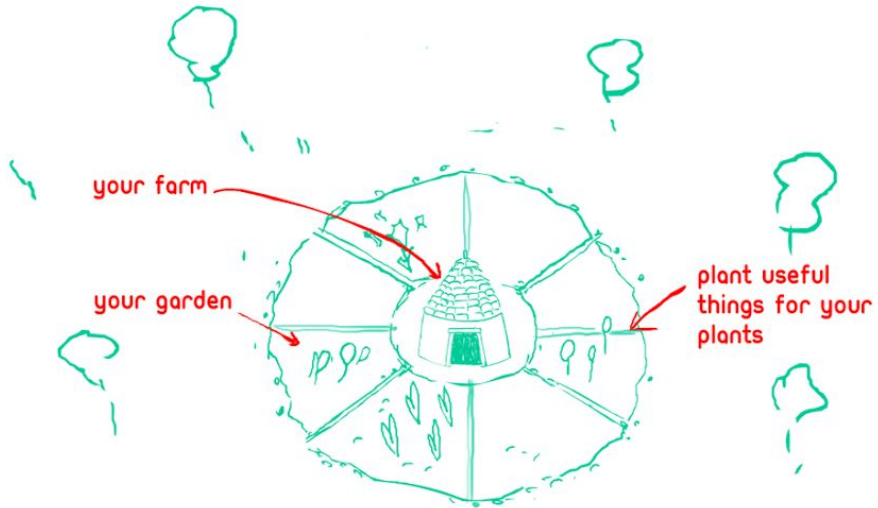
2. TRAIN PLAYING A SPECIFIC MINI GAME (TRAIN EVERY DAY TO GET STRONGER TECHNIQUES)



3. LEARN NEW SUPERPOWER



YOUR FARM



YOUR FARM UNDER ATTACK

