Team name: Rubber Ducks
Project name: The Forest
Team members:
Qinghang Hong qhong@ucsb.edu
Haochen Yang haochenyang@ucsb.edu
Xinyu Liu xinyuli@ucsb.edu
Jerry Liu jiarui@ucsb.edu
Shan Qing s_qing@ucsb.edu

Link to github repo: https://trello.com/b/ehX42rxE/the-forest
Link to trello: https://github.com/liujiaruirz/rubber_ducks/

Project description: A strategy and role-playing game set in a land of Middle Ages. In the game, players own a kingdom. They collect resources like wood, metal, and stone to build the kingdom and make the army stronger. Outside the kingdom, there is a huge forest. In order to explore the mysterious forest, players need to recruit an expedition and forge the powerful weapon to step further into the forest.

Problem Statement: Sometimes people are bored with the mobile games in the market today. Although those games have fantastic graphics and attractive plots, players also need to spend a lot of time to master the complex operations. In our game, players just need to use their intelligence to organize properly and then they can have a good experience.

Project outcome:
The game is divided into two parts, the castle and the forest. First, players are given a huge castle and are able to build buildings and collect resources. The main goal of this game is the exploration of the forest. Therefore, players have to recruit soldiers in the castle and control the expedition in the forest.

How the problem is solved today:
We can see there are many adventures mobile games like this in the app store. One famous one is called Dungeon Survivor which you gather resources in the castle and lead heros out for the adventure. This game is object oriented. Basically, each element has its own features. Depends on different circumstances, the feature can be changed to follow the plot of the game.

Features:
1. Collect resources (gather natural resources and alchemy the gold)
2. Refine
3. Build
4. Recruit heros
5. The Forest (The map and the coordinates of the players)
6. Level up
7. Inventory
8. NPCs
9. Heroes (abilities and the consumption)
10. The battle system

Milestone definitions:

Specification:
The game builds an unreal and mysterious world, in which players can create their own kingdoms. To construct strong empire and army, they need to gather resources including food, wood, metal and stone. Outside the kingdom, there is a huge forest for players to explore. In the forest, players will encounter different types of animals.

Design:
Our game is based on Unity engine and C#. We are going to create objects of forest, people and weapons and link them together.

Prototype:
The castle that players can collect resources and recruit soldiers.
The forest that players can control their expedition.

Testing:
Test the basic and edge cases that customers will face. Mock the users and provide inputs to the web and test the results.

How do you plan to articulate, design, and implement a solution:
It is an offline game on laptops. We want to use mainly C++ libraries, like Unreal. We will need to draw a diagram to construct the game, including all of the layers and simple graphics.

Working Schedule:
Week 1-2: decide the goal of the project, make a plan and divide into two groups, one is learning tech and another is build the basic model (share the difficulties and progress and teach others every two days for 15 mins)
Week 3-5: finish the prototype of the main interface and the maps of the forest (Have a standup meeting daily to share the progress and motivate group members to stay on track)
Week 6-7: done with another feature - the battle system and the characters (NPCs, enemies)
Week 7-8: test the basic and edge cases (Write some possible test cases and give some potential users to experiment)
Week 9-10: finish the development and prepare for the presentation

Architecture:
UML diagrams of 4+ primary data structures:

- Scene class
  - main camera
  - event system
  - canvas
  - map
  - character
    + loadScene()
  + cameraMove
  + update()

- Contains
- * contains
- * contains

- Character class
  - speed
  - jump force
  - Animation
  - groundCheck
  + move
  + fight
  + jump

- Map class
  - Trees
  - Houses
  - collider
    + onCollide()

- Button class
  - position
  - Text
  - Color
    + onClick()
    + SceneManager()
Sequence diagrams:

1 Synchronous for key interactions between classes:
(There is no asynchronous for key interaction between classes in our RPG game project.)
1 User interactions with the system:

- User interaction
- Character
- Monster

- Press key
- Fight()
- GetHealth()
- Fight()
- GetMonsterHealth()