Name\_\_\_\_Baoying\_Lou\_\_\_\_\_\_\_\_ Mark \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/50

[**Instructions**: Remove everything that is not a heading below and fill in with your own diagrams, etc.]

## Brief introduction \_\_/3

I am responsible for the user interface design, which is mainly about the interaction between the software and user. It includes login, logout, view help, view manual, choose game level, view score, view collectables, save score.

## Use case diagram with scenario \_\_14

### Use Case Diagrams

### 

### Scenarios

**[You will need a scenario for each use case]**

**Name:** Login

**Summary:** The user uses the Login button to login

**Actors:** Player

**Preconditions:** Game has been initialized

**Basic sequence:**

**Step 1:** Enter user name.

**Step 2:** Enter password.

**Step 3:** Click login button

**Exceptions:**

**Step 1:** Wrong user name or password: invalid user.

**Post conditions:** Game screen is displayed.

**Priority:** 1\*

**ID:** C01

\*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

**Name:** Logout

**Summary:** The user uses the Logout button to login

**Actors:** Player

**Preconditions:** Game has been initialized

**Basic sequence:**

**Step 1:** Click the logout button

**Post conditions:** Game window is closed.

**Priority:** 1\*

**ID:** C02

\*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

**Name:** View Help

**Summary:** The user uses Help button to see basic operation of the game.

**Actors:** Player.

**Preconditions:** Game has been initialized.

**Basic sequence:**

**Step 1:** Click help button.

**Step 2:** View the information.

**Step 3:** Close help window.

**Exceptions:**

**Step 1:** The help window is closed by mistake: open it again.

**Step 2:** User can’t find the information he/she wants: try manual.

**Post conditions:** Help window is closed.

**Priority:** 2\*

**ID:** C03

\*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

**Name:** View Manual

**Summary:** The user use Manual button to see the detail information of the game.

**Actors:** User.

**Preconditions:** Calculator has been initialized.

**Basic sequence:**

**Step 1:** Click Manual button.

**Step 2:** View manual information.

**Step 3:** Close manual window.

**Exceptions:**

**Step 1:** The manual window is closed by mistake: open it again.

**Step 2:** User can’t find the information he/she wants: Give feedback to developer.

**Post conditions:** Manual window is closed.

**Priority:** 2\*

**ID:** C04

\*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

**Name:** Choose game level

**Summary:** The user can choose a game level he/she wants from home menu.

**Actors:** User.

**Preconditions:** Game has been initialized.

**Basic sequence:**

**Step 1:** Click the home menu.

**Step 2:** Click game level.

**Step 3:** Choose a level from 1 to 3 from the pull-down menu.

**Exceptions:**

**Step 1:** User choose a wrong level: choose it again.

**Post conditions:** The corresponding level of the game is displayed.

**Priority:** 3\*

**ID:** C05

\*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

**Name:** View score

**Summary:** The user could see what score he/she gets.

**Actors:** Player.

**Preconditions:** Game has been initialized.

**Basic sequence:**

**Step 1:** The score is displayed in the up left corner all the time.

**Post conditions:**

**Priority:** 2\*

**ID:** C06

\*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

**Name:** View collectables

**Summary:** The use could see what collectables and how much of them he/she has collected.

**Actors:** player.

**Preconditions:** Game has been initialized.

**Basic sequence:**

**Step 1:** Click the collectables button.

**Step 2:** See the type and number of the collectables.

**Step 3:** Close the collectables window.

**Exceptions:**

**Step 1:** No collectables is collected: Display 0.

**Post conditions:** The information about the collectables is displayed.

**Priority:** 2\*

**ID:** C07

\*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

**Name:** Save score

**Summary:** The user uses the save score button to save the score.

**Actors:** Player.

**Preconditions:** Game has been initialized.

**Basic sequence:**

**Step 1:** Click the save score button.

**Exceptions:**

**Step 1:** User gets 0 score: Store 0 to the user-score database.

**Post conditions:** Calculated value is displayed.

**Priority:** 3\*

**ID:** C08

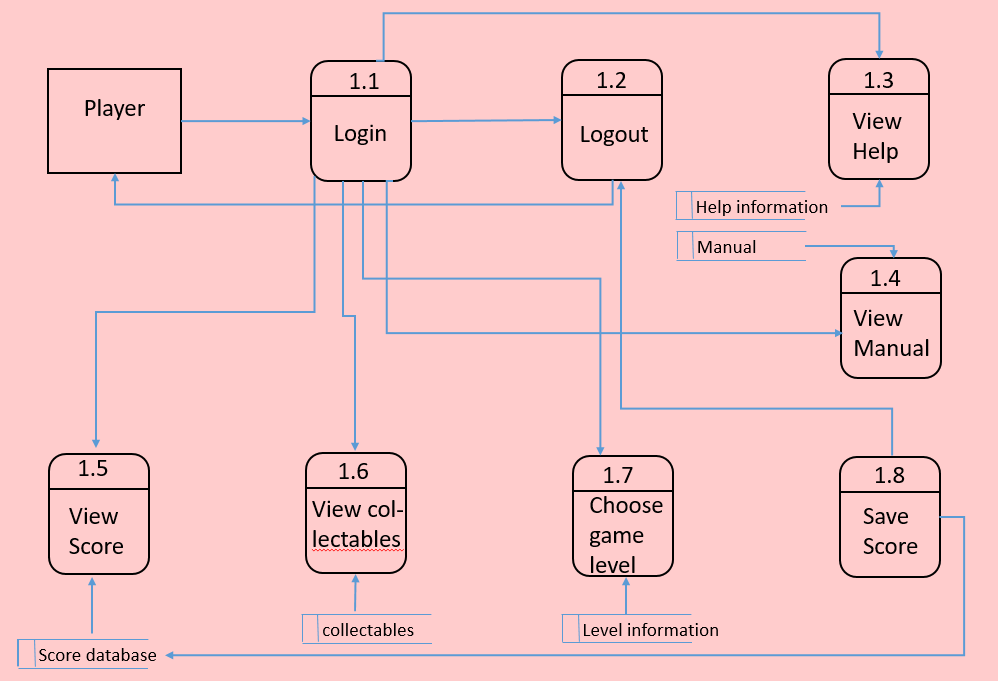
\*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

## Data Flow diagram(s) from Level 0 to process description for your feature \_\_\_\_\_\_\_14

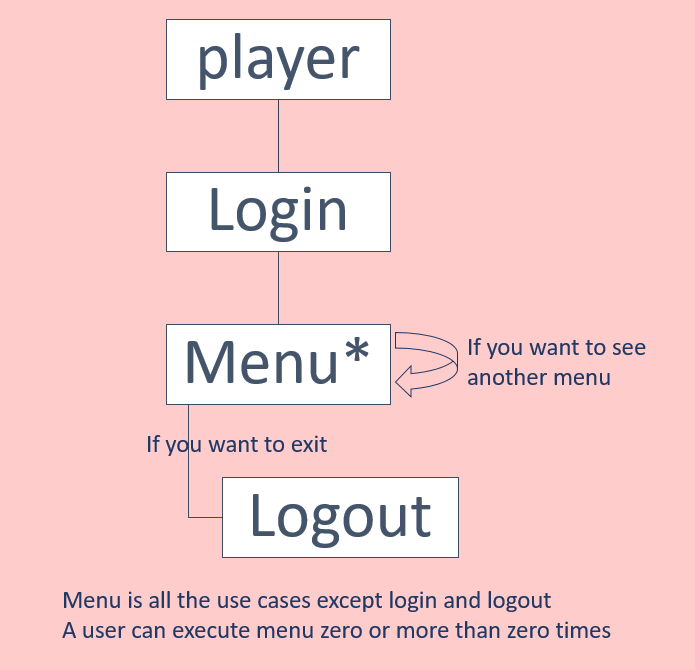
[Get the Level 0 from your team. Highlight the path to your feature]

Example:

### Data Flow Diagrams



### Process Descriptions



## Acceptance Tests \_\_\_\_\_\_\_\_9

[Describe the inputs and outputs of the tests you will run. Ensure you cover all the boundary cases.]

1. Test login and logout. Make sure user could login and logout normally.

2. Test Help menu. It should display the basic information about how to play this game.

3. Test manual menu. It displays all the detail information about the game.

4. Test Score. It should display the correct number that the user wins.

5. Test View collectables. It should display all the collectables, including type and number.

6. Test Save score. It should save the user’s attribute and score to a database if user clicks it.

7. Test Choose game level. If one level is chosen, the corresponding scene should be diaplayed.

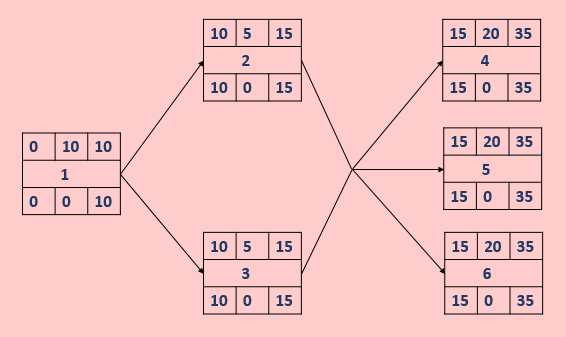
## Timeline \_\_\_\_\_\_\_\_\_/10

[Figure out the tasks required to complete your feature]

### Work items

|  |  |  |
| --- | --- | --- |
| Task | Duration (day) | Predecessor Task(s) |
| 1. Requirements Collection | 10 | - |
| 2. Create .h files | 5 | 1 |
| 3. Create .cpp files | 5 | 1 |
| 4. Fill in .h files | 20 | 2, 3 |
| 5. Fill in .cpp files | 20 | 2, 3 |
| 6. Testing | 20 | 2.3 |

### Pert diagram



### Gantt timeline

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |  | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |  |  | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 2 | , | 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 2 | , | 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 2 | , | 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 |