Tower Dream

Incremental & Regression Testing

Team 15

Yixiong He, Tiger Lu, Kevin Jiang, Weifeng Huang, Zhe Chen

❖CLASSIFICATION OF COMPONENT

Login Module:

- Login Screen
  - Input: Email, Password, login button, Register button
  - Output: Navigate to the game screen or Descriptive error
  - Parent Dependency: None
  - Child Dependency: Login action, Register Screen

- Register Screen
  - Input: Email, Password, Register button
  - Output: Register Successfully Screen or Descriptive error
  - Parent Dependency: Login Screen
  - Child Dependency: Register Action, Main Menu Action
- Login Action
  - Input: Email and Password
  - Output: Login success or failure
  - Parent Dependency: None
  - Child Dependency: Main menu screen

- Register Action
  - Input: Email and Password
  - Output: Register successfully or failure
  - Parent Dependency: Register Screen
  - Child Dependency: Login Screen

UI Module:

- Main Menu Screen
  - Input: Main menu options
  - Output: Entering the new game
  - Parent Dependency: Login Screen
  - Child Dependency: In-Game Screen

- In-Game Screen
  - Input: None
  - Output: Defending towers and defensive troops moving
  - Parent Dependency: Main Menu Screen

    - Child Dependency: Help-info Box Action
- Help-info Screen
    - Input: info button
    - Output: Help-info information
    - Parent Dependency: In-Game Screen
    - Child Dependency: In-Game Screen

- Pause-Game Screen
    - Input: Pause button
    - Output: The current game stops
    - Parent Dependency: In-Game Screen
    - Child Dependency: In-Game Screen, In-Game Sell Defenses Actions and In-Game Optimal Target Action

Game Activities Module:

- Main Menu Action
    - Input: Main Menu Options
    - Output: Each new game
    - Parent Dependency: Login Screen
    - Child Dependency: None

- In-Game Sell Defenses Action
    - Input: None
    - Output: Sell the defenses
    - Parent Dependency: In-Game Screen
    - Child Dependency: None

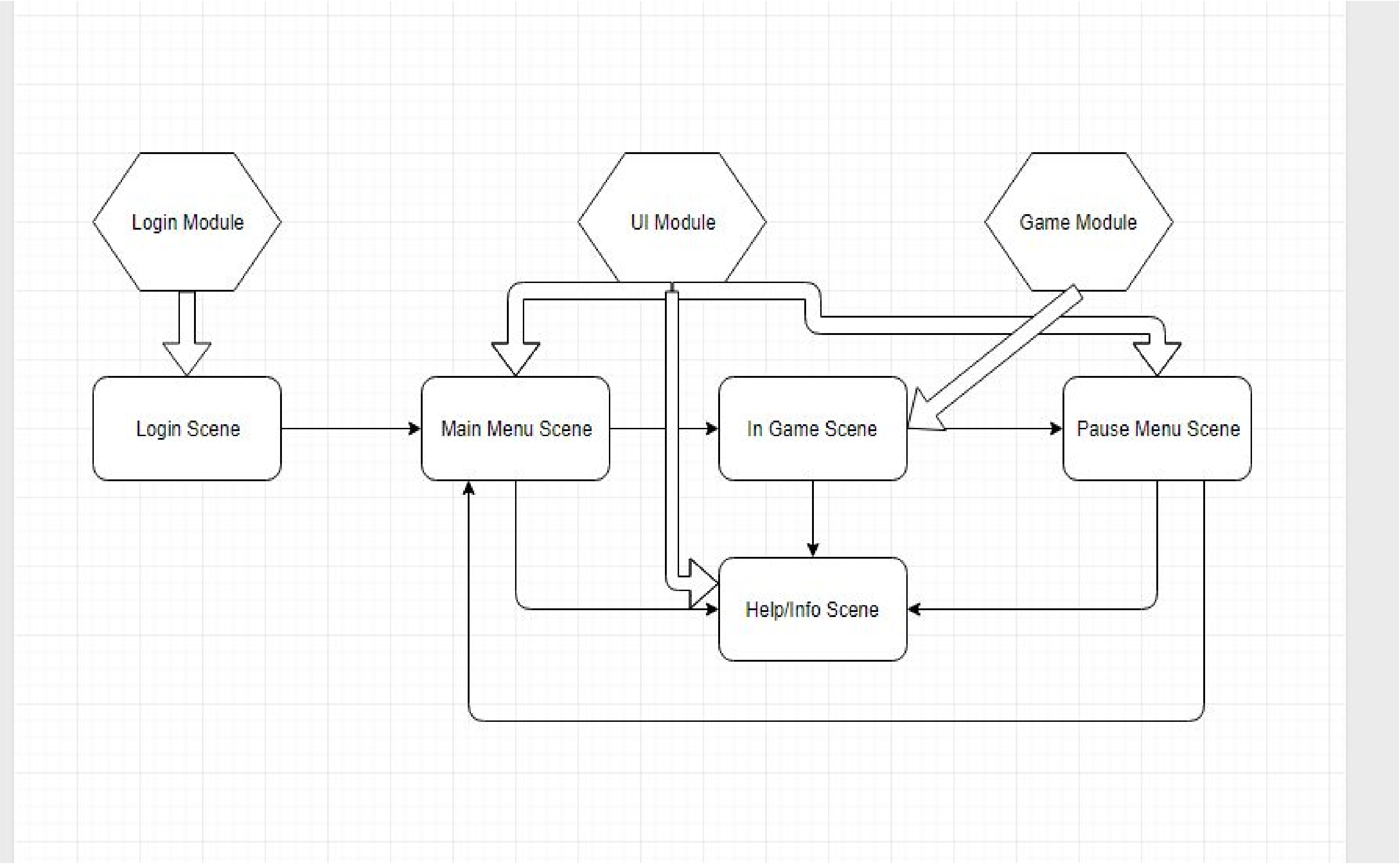
- In-Game Optimal Target Action
    - Input: None
    - Output: Choose the optimal target automatically
    - Parent Dependency: In-Game Screen
    - Child Dependency: None

- Help-info Box Action
    - Input: info button
    - Output: Pop up Help-info information
    - Parent Dependency: In-Game Screen
    - Child Dependency: In-Game Screen

- Pause-Game Action
    - Input: None
    - Output: Pause the current game
    - Parent Dependency: In-Game Screen
    - Child Dependency: In-Game Screen, In-Game Sell Defenses Actions and In-Game Optimal Target Action

Incremental testing technique

We are using the bottom up testing approach, as we tested different modules/scenes individually first, then we tested transitions and integrations between scenes and modules



Using top down testing would make it too hard to find individual bugs, therefore bottom up testing was more efficient and useful.

Incremental and Regression Testing

Register and Login Module

- Testing is designed for checking the normal functionality of login and register, also checking for invalid characters.

Defect Log for Register & Login Module

Defect #	Description	Severity	How Corrected
1	Login is not implemented into game	1	Implement working php scripts into unity game.
2	Non-alphanumeric characters are allowed	2	Write checks for invalid characters

Regression Testing

Defect #	Description	Severity	How Corrected
1	Adding two buttons into the login page and organize with the right position when the keyboard pops up to enter the text into the email and password fields	1	Relocate buttons on the login page
2	When the cursor is in the box of the email/password, the keyboard pops up.	3	Modify the properties of the login layout and keyboard popping up action

Menus/UI Module Module

- Testing is designed for checking the functionality of each button directing the user to the related scene in the game.

Defect Log for Menus/UI Module

Defect #	Description	Severity	How Corrected
1	Moving to next scene from main menu always goes to scene #1 from scene #0	1	Changed scene moving to name based rather than order based, it will fit in to future use for level selection too.
2	Pause menu wouldn't pause the game	1	Edited pause menu script to change game time to 0 when pausing, and back to 1 when pause menu is gone

Regression Testing

Defect #	Description	Severity	How Corrected
1	Modifying the components to be a different type of GUI, does not look like the other screens in the app	3	Modify the position of the pause button and make it coincide with the previous screens.
2	EnemyPrefab function is still expecting different element for the game screen	3	Change the EnemyPrefab function in order to have a better view of the game

Gameplay Module

- Tests that are designed to check in game features such as combat and building.

Defect Log For Gameplay Module

Defect #	Description	Severity	Corrected
1, 2	User able to walk off the map, Map boundaries aren't concretely defined	1	The levels are not designed yet, and each levels will have different restricted boundaries. So the boundaries are not set yet but it will be implemented through layers and movement restrictions along with level creation.
2	Hard to tell when it is setup time and wave time	2	Created an obvious indication to separate building wave and enemy wave, it will also automatically adjust to new wave set.
3	Standing directly in front of an enemy unit will cause it to be blocked and unable to move.	1	Made player unit to ignore unit collision, so user will not be able to stall the wave on purpose. The enemy unity will keep its collision for possible compatibility when we implement offensive end on the gameplay module.
4	Pausing only pauses friendly units and towers, and enemy units still move	1	Edit enemy movement to move along with game time (Not yet

			implemented)
5, 8	Friendly units also show up as a counter in enemies left. Enemies left counter bugs due to refresh rate not being quick enough. If enemy dies too fast, counter sometimes doesn’t decrease	1	Fixed the possible cause of the problem with tags (used to count enemies), added distinguish tags to all units.
6	Gold gain is not fully functional	1	It will be implemented in Sprint 2.

Regression Testing

Defect #	Description	Severity	How Corrected
1	Adding new fields to Tower.prefab file to make it indicate the building wave	3	Modify the components of the tower fields
2	UpdateClock() function is still expecting different element for the remaining time	3	Update timer more frequently so clock is displayed correctly
3	EnemyCrossed() function called different times every frame	1	In the function, when no more waves and enemies remain, player beats the level.