

Implementation

Requirement Changes

The requirements published by the previous group had already been fulfilled and assigned associated risks. We did modify a lot of these already existing requirements, however, because we felt that they were in some cases ambiguous, they allowed for exploitation, and in other cases, they were out of date due to changes in the game brief for this assessment iteration. An example of this is the addition of more colleges and departments into the game. We did, also, add in a new set of requirements which can be seen in the updated requirements document. We then implemented the requirements that we added and we have presented all of our new and/or modified requirements tabular form. Due to the page limit we must adhere to, all of the modified or newly added requirements are listed in the functionality update document with appropriate justification. The functionality update document can be found here: <http://limewire.me/docs//assessment3/FunctionalityUpdates.pdf>.

You can see a full list of the updated requirements here:
<http://limewire.me/docs//assessment3/UpdatedReq3.pdf>.

Architecture changes

Numerous small changes were made to the pre-existing code passed on from the previous group as well as some large changes and additions such as the minigame. We have shown these changes in UML diagrams. The general structure of the architecture remained moving forward with the project and LibGDX is still the framework used for all development including the new minigame. We have added, to the UML class diagrams, any new methods, classes and variables which we created in the development of the source code during this assessment. The changes to architecture can be seen in the UML class diagrams listed on our website:

<http://limewire.me/docs//assessment3/UMLBase.png>
<http://limewire.me/docs//assessment3/UMLCombat.png>
<http://limewire.me/docs//assessment3/UMLMinigame.png>
<http://limewire.me/docs//assessment3/UMLRelationship.png>
<http://limewire.me/docs//assessment3/UMLScreen.png>
<http://limewire.me/docs//assessment3/UMLYorkPirates.png>

You can see the architecture document here:
<https://taylorwillmott.com/SEPR/Assessment/2/Arch2.pdf>

Significant New Changes

Minigame, new map (new college and new department), combat system rework, new upgrade, new weapons, weapon purchase feature, weapon management feature, Final boss department, End game screen

Probably the most significant change was the addition of a playable minigame to each college in the map, The ghost maze minigame was made independently of the main game and completing it will only give points upon completion of the maze as to not affect the progression of the player in their playthrough of the main game.

New colleges and new department

One of the main requirements following on from the previous assessment was to increase the number of colleges and departments to five and three respectively. When deciding which buildings to add to the game, we considered which would be most suitable according to the requirements of the project and balance of the game. In this case, the map needed to be accurate to the real university, and the buildings need to be well spread to ensure the map is balanced and filled with features. We decided that Alcuin and Wentworth colleges and the Economics department would be fit best.

Reworked map

With similar consideration to the previous point, the map needed to be changed to more accurately represent the university, so islands had to be moved and redesigned to fit the new additions made during this assessment.

Added ability to purchase weapons from departments

The departments classes were expanded to give them more functionality and use within the game to reach the final objective of killing the boss ships. In addition to selling overall ship upgrades, each department now sells a unique weapon that the player can use. This fully satisfies the requirement for departments to act as shopping points through these purchasable items. The new Economics department also had a new upgrade added for it that increases accuracy.

New weapons and weapon management feature

Two additional weapons have been added to the game, and a previously inaccessible weapon has been made usable for the player. This change has been made in order to make the combat a more interesting and dynamic experience. The two new weapons added are the Double Shot and the Explosive Shell. The weapon previously unusable for the player that was added is the Grape Shot. Since the player can't equip more than 4 weapons at once, a feature was added to friendly college menus to unequip and equip owned weapons. This lets the player personalise their ship how they see fit before engaging enemies in combat, fulfilling the requirement of the friendly colleges acting as a hub to customise the ship.

Combat Rework

The player's ship's health is now split up into two parts: sails health and hull health. The player must manage the health of these two parts during combat, adding in another element of strategy as damage to either part will now impact the effectiveness of their attacks. Damage to the sails will result in reduced accuracy for weapons like the broadside cannons while damage to the hull will decrease the amount of harm the ram attack causes to the enemy ship. Different weapons may also cause damage to a specific part of the ship. For example, the grape shot will only cause damage to the sails while ramming will harm the hull. Other attacks may cause damage to both parts in varying amounts; up to a quarter of damage from the broadside cannons for example may be applied to the sails with the rest going to the hull.

Gameplay and quality tweaks

Various tweaks were also made to better meet the requirement that the game should be easy to pick up and enjoyable to play. Changes include faster and greater control over the ship in the sailing mode, improvement to the map to help differentiate the islands. More options were given to heal different amounts and manage the ship status. Descriptions for weapons in combat are less ambiguous, making the effect of weapons clearer. The health of the player is now shown on all screens to help keep track of their status during different parts of the game.

Combat was also reworked to speed up the game, as the previous balance of the game had combat we thought was too slow and tedious.

When carrying on the project from the previous group in order to progress the game further amendments had to be made to existing classes, any bugs needed to be squashed and more classes needed to be created to add more functionality to the game, they are as follows:

Minigame

When the player sails to a college when he enters the interface he has the option to take part in the minigame where upon successful completion of the minigame will gain 100 points. It is a 2D maze game where the player controls a sprite trying to reach a goal tile while avoiding dangerous mobs along the way. The classes created for the minigame include:

- BaseScreen - The base class of all the Screens created in the game, which implements the interface Screen. BaseScreen sets the size of the window and the rewrite Screen by adding stages (main stage and UI stage).
- PhysicsActor - The base class of all the actors that have physics collisions created in the game. Extended from BaseActor.
- MiniGameScreen - The main screen of the mini game, which updates based over time. The class gets resources from a tile map and stores the position of walls, players and enemies. It displays all the elements and call functions to make the mini game playable.
- MiniGameFinishScreen - The finish screen of the mini game. Shows different message base on the status of the game (win and lose), lead player back to sailing screen and give extra points to the player if they win.
- MiniGameEnemy - The enemy class of mini game extends PhysicsActor, which is the function setup mini game enemies with its texture, move speed and also its random movement AI.
- MiniGamePlayer - The player class of mini game extends PhysicsActor, allow the game setup the player. Also includes the rules of player movement, and how the player interacts with enemies.

Unimplemented requirements

All required features which are given in the assessment brief, we feel, have been implemented in this assessment iteration. During development, we did not manage to implement all requirements due to tight time restriction on the assessment which was consumed by documentation and testing as well as programming. We prioritised the requirements which we wanted to implement and made sure all of these were implemented and fully tested with appropriate documentation discussing these requirements.

Requirements can be seen here: <http://limewire.me/docs//assessment3/UpdatedReq3.pdf>.

The requirements which we did not implement are discussed below:

Save/Load feature	Due to relatively short playtime of the game, we did not feel like this feature was a priority and we prioritised gameplay functionality first
Weather events	We did not manage to implement this feature as it involved altering too much previously written code from assessment 2 which, as discussed in the change report, would have caused disruptions in the progress of the project and required new tests to be written
Fog of war	We did not implement fog of war as with the time restrictions, we felt that adding this feature would have made it difficult to test other features efficiently. It also would have required us to change a lot of previously written code which would have required us to write a brand new set of unit tests which is very time consuming

References

- Functionality update document:
<http://limewire.me/docs//assessment3/FunctionalityUpdates.pdf>
- Updated requirements:
<http://limewire.me/docs//assessment3/UpdatedReq3.pdf>
- Concrete UML diagrams:
<http://limewire.me/docs//assessment3/UMLBase.png>
<http://limewire.me/docs//assessment3/UMLCombat.png>
<http://limewire.me/docs//assessment3/UMLMinigame.png>
<http://limewire.me/docs//assessment3/UMLRelationship.png>
<http://limewire.me/docs//assessment3/UMLScreen.png>
<http://limewire.me/docs//assessment3/UMLYorkPirates.png>
- Rear Admirals Architecture Document:
<https://taylorwillmott.com/SEPR/Assessment/2/Arch2.pdf>