Team iPatch

Assessment 4: Implementation report

Christian Pardillo Laursen
Filip Makosza
Joseph Leigh
Mingxuan Weng
Oliver Relph

Github repository with the code located at
github.com/Team-iPatch/Team-iPatch

Final requirements found at
Team-iPatch.github.io/Assessment-4/Req4.pdf

Required changes:

For assessment 4, two changes to the software were required: implementing crew members and adding a new type of obstacle. This was tackled as follows:

1. **Obstacle: whirlpools**

Related requirement: 18
Whirlpools fit the requirement changes well, since they cannot be defeated and hinder the player's progress by returning them back to the start. Requirement 18 was added to fit this change.

To implement whirlpools, the **WhirlpoolControl** class was created, which manages most of the functionality of whirlpools when attached to the corresponding Spatials. This involves detecting the proximity of the player and performing an animation which teleports them to the spawn location. This is all specified by R18.b.

They are distributed across the map by extending the **BuildingGenerator** with method **generateWhirlpool**. This takes care of placing whirlpools on the map and giving them the appropriate attributes when given coordinates. This is required by R18.a.

2. **Crew members**

Related requirement: 12.c
To realize the other requirements change, what was needed was the implementation of a new upgrade, which can then be used to upgrade the ship every time a given objective is completed. The chosen objective was defeating colleges, since it is central to game completion and it also gives the player a sense of progression. This also had the secondary effect of making the game much easier each time a college was defeated, so the requirement for increasing enemy difficulty with colleges defeated was added and implemented.

To implement crew members, the **PlayerControlState** mechanic for shooting was modified to shoot several bullets from a single cannon. Then, the **levelUp** method can be used for both tracking game difficulty and number of cannons, since the player levels up when they defeat a college.

Additional changes:

1. **Levelling**

Related requirement: 15.c, R6.b

In order to provide a sense of progression to the player, as well as keeping the game at a reasonable difficulty level as the player gains upgrades, levelling was added.

The level is stored in the **PlayerControlState,** together with the **levelUp** and **getLevel** methods. Whenever a college is defeated it calls **levelUp**. Then, when enemies are spawned or colleges are engaged their parameters are adjusted according to the level. For regular ships this only increases the health, for colleges it decreases shot speed and increases shot frequency, rotation speed and health. It also increases the amount of points received when defeating enemies.

2. **Removed redundant code**

Although we have no specific requirement regarding code quality, our code had accumulated a few redundant or unused methods that added clutter and reduced readability. This was most apparent in **DepartmentControl** and **PlayerControlState,** where some methods had been overloaded or implemented elsewhere but the originals had not been removed. Therefore for this assignment an attempt was made at minimizing the amount of unused code.

3. **Restricted shot speed, made holding the left mouse button shoot continuously**

Related requirement: 12.c

This change is not directly required by R12.c but prior to this change, the player's firing rate was only bounded by their click speed. Having this in the completed game would make clicking quickly a great advantage and de-emphasize the more interesting mechanics, such as avoiding bullets, since enemies can be defeated too quickly if the player clicks fast enough.

Therefore a timer was added between shots and an auto-shot feature was introduced, which places a consistent limit on the player and makes balancing the game easier, as we do not have to account for click speed.

4. **Improved the win screen to show points and time**

Related requirement: 6.c

In our design, we specified that points would be added at the end of the game according to how quickly it was completed. This has been reinterpreted to show the player how long they have taken to complete the game. We believe this has more value, since the game has high replay value due to it being randomized every time and players might enjoy seeing how quickly they can complete it.

This change was carried out by modifying the GUI for the win screen, shown when the player defeats all 5 colleges, adding parameters for points obtained and time taken. Time taken was tracked in **PlayerControlState** with a variable which increases every frame.