# Sprint 4 | 17th November – 23rd November

## **Sprint Overview**

### Preview

After the completion of the last sprint, we had created our first iteration of the game. Going into the fourth sprint, we planned to continue with the development of the game, with a particular focus on the creation of user stories, use cases, and CRC cards. These were discussed in the previous sprint and a few user stories were created. We had expectations that we would create these during this sprint. We also wanted to start working on the UI to adhere to the dungeon theme.

### Review

At the start of this sprint, we broke down the work into designated roles for each team member so that we were able to manage a greater number of tasks in one sprint. The documentation team started to create more structured and logical sprint documents from this sprint and created further content for the game. The development team were intending to build more core functional classes of the game like pipe generation, rotation, scoring, and a menu with a dungeon design. However, after meeting with client, it was suggested that whilst still working on the game development, a greater focus should be placed on documentation.

## **Tasks**

This sprint was split into two sections. As a development team, we completed the ‘First Tasks’ before moving on to the ‘Second Tasks’ later on in the Sprint.

### First Tasks

|  |  |  |
| --- | --- | --- |
| Code | Team Members | Tasks |
| **S4.1-T1** | Development Team (Marcus, Zoe, Matthew, Leo) | Create User Stories for the game. |
| **S4.1-T2** | Development Team (Marcus, Zoe, Matthew, Leo) | Create Use Cases for the game. |
| **S4.1-T3** | Development Team (Marcus, Zoe, Matthew, Leo) | Create CRC cards for the game. |
| **S4.1-T4** | Whole Team | Designate roles for this project. |
| **S4.1-T5** | Development Team (Marcus, Zoe, Matthew, Leo) | Add detail to the backlog under the parent tasks “Create core functionality of the game”, “Implement dungeon design to the game”, “Add features.”. |
| **S4.1-T6** | Shawn (Chin) | Overview and user interface design sections for sprint documentation. |
| **S4.1-T7** | Rachan | Amend meeting records to sprint documentation. |
| **S4.1-T8** | Michelle | Write preview section for sprint documentation and format sprint documentation and meeting minutes for customer meeting. |

### Parent Tasks

*Note:* Parent tasks are those which contain various smaller tasks. They are ranked in order of priority (e.g., PT1 is more important than PT2, so any tasks in PT1 take priority over any from PT2). From here on, tasks shall refer to their parent task in the Priority column in brackets.

The parent tasks in the above backlog were formed from the TBD tasks from Sprint 3.

|  |  |  |
| --- | --- | --- |
| Code | Team Members | Tasks |
| **PT1** | Development Team (Marcus, Zoe, Matthew, Leo) | Create core functionality of the game. |
| **PT2** | Development Team (Marcus, Zoe, Matthew, Leo) + Shawn | Implement dungeon design to the game. |
| **PT3** | Development Team (Marcus, Zoe, Matthew, Leo) | Add features. |

### Second Tasks

|  |  |  |
| --- | --- | --- |
| Code | Team Members | Tasks |
| **S4.2-T1** | Leo | Refactor the pipe functionality and character movement from the mock-up of the game to reflect the CRC card design. This is complete when the acceptance criteria of R2, R3, R4, R5 is fulfilled and the use case tests for Use Case 3 and 6 are passed. |
| **S4.2-T2** | Leo | Add pipe rotation to the game with logic to decide if pipe piece can be added to pipe system. This is complete when the acceptance criteria of R3, R6 is fulfilled and the use case tests for Use Case 3 are passed. |
| **S4.2-T3** | Zoe | Create and implement the basic functionality of the score class. This is complete when the acceptance criteria of R8, R9 is fulfilled and the use case tests for Use Case 7 is passed. |
| **S4.2-T4** | Matthew | Create and implement the basic functionality of the menu class. This is complete when the acceptance criteria of R1, R9, R10 is fulfilled and the use case tests for Use Case 8 is passed. |
| **S4.2-T5** | Marcus | Create and implement the basic functionality of the pipe generation class. This is complete when the acceptance criteria of R7 is fulfilled and the use case tests for Use Case 2 is passed. |
| **S4.2-T6** | Shawn | Design the menu background. This is complete upon customer approval. |

### Other tasks formed from S4.1-T5 + bugs to fix at the end of Sprint 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Code | Team Members | Tasks | | |
| **PT1**-S4.2-7 |  |  | | |
| S4.2-7-T1 | Dev Team | Allow character to enter both sides of the pipe when rotating. | | |
| S4.2-7-T2 | Dev Team | Implement high score saving, and menu/game-over display. | | |
| S4.2-7-T3 | Dev Team | Change the game over wall, either to another character or water etc. | | |
| S4.2-7-T4 | Dev Team | Make character die if it touches the top or bottom of the screen. | | |
| S4.2-7-T5 | Dev Team | Add coin generation. | | |
| S4.2-7-T6 | Dev Team | Implement logic to prevent adding a pipe onto itself. | | |
| S4.2-7-T7 | Dev Team | Add a pause and play button during play (that works in accordance with previously defined documentation). | | |
| S4.2-7-T8 | Dev Team | Implement coin collection to increase score. | | |
| S4.2-7-T9 | Dev Team | Highlight the pipe piece that is currently selected by the player. | | |
| S4.2-7-T10 | Dev Team | Users test the game to get feedback on functionality. | | |
| S4.2-7-T11 | Dev Team | Add obstacle generation. | | |
| S4.2-7-T12 | Dev Team | Get camera to speed up continuously as the game plays. | | |
| S4.2-7-T13 | Dev Team | Implement death on obstacle collision. | | |
| S4.2-7-T14 | Dev Team | Add a game over screen, with a restart and home button. | | |
| S4.2-7-T15 | Dev Team | Make character appear within pipes rather than in front of them. | | |
| S4.2-7-T16 | Dev Team | Stop rotation of pipes after game over (or in pause if pause is possible). | | |
| S4.2-7-T17 | Dev Team | Implement pausing the game. | | |
| S4.2-7-T18 | Dev Team | Add a restart and home button during play. | | |
| **PT2**-S4.2-7 |  |  | | |
| S4.2-7-T18 | Dev Team | Add the menu background to the game (from Shawn). | | |
| S4.2-7-T19 | Dev Team | Add the rolling background (from Shawn) to the game. | | |
| S4.2-7-T20 | Dev Team | Change the character to a more appropriate design. | | |
| S4.2-7-T21 | Dev Team | Add lighting to the scene. | | |
| S4.2-7-T22 | Dev Team | Stylize UI to match dungeon theme. | | |
| S4.2-7-T23 | Dev Team | Users test the game to get feedback on design. | | |
| **PT3**-S4.2-7 |  |  | | |
| \*\*\* Not needed until the above tasks are complete. | | |  |  |

## **Backlog**

### First Tasks Ordered Backlog

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Priority | Tasks | Date of Creation | Date of Completion | Status at End of Sprint |
| Whole Team 1 | S4.1-T4 | 17/11/21 | 17/11/21 | Complete |
| Dev Team 2 | S4.1- T1 (prev. TBD) | 17/11/21 | 18/11/21 | Complete |
| Dev Team 3 | S4.1-T2 (prev. TBD) | 17/11/21 | 19/11/21 | Complete |
| Dev Team 4 | S4.1-T3 (prev. TBD) | 17/11/21 | 20/11/21 | Complete |
| Dev Team 5 | S4.1-T5 | 17/11/21 | 21/11/21 | Completed |
| Shawn (Chin) 2 | S4.1-T6 | 17/11/21 | 22/11/21 | Completed |
| Rachan 2 | S4.1-T7 | 17/11/21 | 22/11/21 | Completed |
| Michelle 2 | S4.1-T8 | 17/11/21 | 24/11/21 |  |
| Dev Team 8 (prev. WT) | PT1 (parent task – no set sprint zone) | 10/11/21 | - | Not started |
| Dev Team 9 (prev. WT) | PT2 | 10/11/21 | - | Not started |
| Dev Team 10 (prev. WT) | PT3 | 10/11/21 | - | Not started |

### Second Tasks Ordered Backlog

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Priority | Tasks | Date of Creation | Date of Completion | Status at End of Sprint |
| Dev Team | **PT1** | 10/11/21 | - | Ongoing |
| (1) Leo 1 | S4.2- T1 | 17/11/21 | 20/11/21 | Ongoing |
| (1) Zoe 1 | S4.2-T3 | 17/11/21 | 22/11/21 | Ongoing |
| (1) Matthew 1 | S4.2-T4 | 17/11/21 | - | Ongoing |
| (1) Marcus 1 | S4.2-T5 | 17/11/21 | 22/11/21 | Complete |
| (1) Leo 2 | S4.2-T2 | 17/11/21 | - | Ongoing |
| (1) TBD | S4.2-7-T1 | 21/11/21 | - | Not Started |
| (1) TBD | S4.2-7-T2 | 21/11/21 | - | Not Started |
| (1) TBD | S4.2-7-T3 | 21/11/21 | - | Not Started |
| (1) TBD | S4.2-7-T4 | 21/11/21 | - | Not Started |
| (1) TBD | S4.2-7-T5 | 21/11/21 | - | Not Started |
| (1) TBD | S4.2-7-T6 | 21/11/21 | - | Not Started |
| (1) TBD | S4.2-7-T7 | 21/11/21 | - | Not Started |
| (1) TBD | S4.2-7-T8 | 21/11/21 | - | Not Started |
| (1) TBD | S4.2-7-T9 | 21/11/21 | - | Not Started |
| (1) TBD | S4.2-7-T10 | 21/11/21 | - | Not Started |
| (1) TBD | S4.2-7-T11 | 21/11/21 | - | Not Started |
| (1) TBD | S4.2-7-T12 | 21/11/21 | - | Not Started |
| (1) TBD | S4.2-7-T13 | 21/11/21 | - | Not Started |
| (1) TBD | S4.2-7-T14 | 21/11/21 | - | Not Started |
| (1) TBD | S4.2-7-T15 | 21/11/21 | - | Not Started |
| (1) TBD | S4.2-7-T16 | 21/11/21 | - | Not Started |
| (1) TBD | S4.2-7-T17 | 21/11/21 | - | Not Started |
| Dev Team | **PT2** | 10/11/21 | - | Ongoing |
| (2) Shawn 1 | S4.2-T6 | 17/11/21 | - | Complete |
| (2) TBD | S4.2-7-T18 | 21/11/21 | - | Not Started |
| (2) TBD | S4.2-7-T19 | 21/11/21 | - | Not Started |
| (2) TBD | S4.2-7-T20 | 21/11/21 | - | Not Started |
| (2) TBD | S4.2-7-T21 | 21/11/21 | - | Not Started |
| (2) TBD | S4.2-7-T22 | 21/11/21 | - | Not Started |
| (2) TBD | S4.2-7-T23 | 21/11/21 | - | Not Started |
| Dev Team | **PT3** | 10/11/21 | - | Not Started |

## **Meeting Records**

### Meetings

|  |  |  |  |
| --- | --- | --- | --- |
| Overview | Duration | Date & Time | Attended By |
| Development Internal meeting | 40 minutes | Friday (19th November 2021) - 16:00 - 16:40 | Marcus, Zoe, Leo, Matthew |
| Documentation Internal meeting | 50 minutes | Saturday (20th November 2021) - 12:00 - 12:50 | Michelle, Rachan |
| Internal meeting (Development x Documentation) | 19 minutes | Sunday (21st November 2021) - 16:00-16:19 | Marcus, Rachan |
| Client Meeting | 15 minutes | Wednesday (24th November 2021) - 11:45 - 12:00 | All Team members |

### Minutes

|  |  |
| --- | --- |
| Time | Wednesday 24th November 12:00-12:15 (Sprint 4) |
| Led By | Rachan and Leo |
| Minutes Taken By | Michelle |
| Participants | Whole team and customer |

Agenda

1. Discuss documentation
2. Discuss what has been achieved in the previous sprint

|  |  |
| --- | --- |
| Agenda Item | Summary |
| Discuss documentation | * Need to add section numbers and page numbers to documentation |
| Discuss what has been achieved in the previous sprint | * Pipe logic was added * Menu is in progress * Rotation of pipes and continuous pipe generation * Focused on core functionality first * Client asked when the design will be implemented. This will be after further progress of the core functionality * Split the development team – half to fix the bugs, and the other half to work on progressing the game functionality * Finish the core functionality and add the current designs by the next week * Add other features later factoring in the complexity * Priority of features: coins, obstacles to avoid, more complex pipe shape, pipe characterisation, beat bonus if you place pipe to the beat of the music * Decided to lock roles in development, documentation, and design rather than switching round each week * Further down the line it is planned for more people to join the documentation team |

Action Items

|  |  |
| --- | --- |
| No. | Action |
| 1 | Finish the core functionality add the current designs. |
| 2 | Add section numbers and page numbers to documentation. |

## **Customer Interview and Analysis**

**Interview Highlights:**

**Q (Team):** Does the half-finished game we've shown so far meet your requirements? What improvements do you think we should make?

**A (Client):** Cool! I like it!

**Q (Client):** I was wondering what you would do to increase the fun and complexity of the game?

**A (Team):** I think we can add obstacles and coins to add complexity and fun to the game.

**Q (Team):** What do you think of our documentation? Is there anything that needs to be improved?

**A (Client):** It looks like you've done a lot of work, but some details still need to be worked out. First, in the customer interviews section, you need to summarize and analyse the records of each customer interview and compare what you have accomplished in your current sprint.

**Q (Team):** We have found a background picture and an interface picture for our game. Would you please check whether it is suitable?

**A (Client):** Cool! I think that's fair enough.

**Analysis:**

In this meeting, the customer was interested in the details and extensions of the game. The client put forward some detailed requirements which need to be considered for the next week's meeting. This mainly includes the revision of documentation, including the records and analysis of each customer interview, as well as detailed records of development content.

The customer was satisfied with the game development we have achieved so far but felt like we had fallen behind on documentation. We will therefore pay greater attention to this over the remainder of the sprint and the subsequent sprint.

## **User Stories**

The User Stories R10, R11, R12, R13, R14, R15, R16, R17, R18, R19 and R20 were added this week.

### R10

|  |  |
| --- | --- |
| User Story: Menu R10 | Acceptance Criteria |
| As a Player,  I want to have a home menu,  So that I can control when I play the game. | Criterion 1.  **IF** the player loads up the game...  **THEN** take the player to the home menu. |
| Priority: 4/5  Version: 1  Date: 17/11/21 |

### R11

|  |  |
| --- | --- |
| User Story: Animate Character R11 | Acceptance Criteria |
| As a Player,  I want my character to be animated when moving, standing still and idle,  So that the overall game experience feels more natural. | Criterion 1.  **IF** the character is alive...  **WHEN** it begins/stops moving...  **THEN** a change in animations need to occur to indicate how the player is moving.  Criterion 2.  **IF** the character dies...  **THEN** there needs to be an animation showing to the player that they have died. |
| Priority: 3/5  Version: 1  Date: 17/11/21 |

### R12

|  |  |
| --- | --- |
| User Story: Continuing/Pausing the game R12 | Acceptance Criteria |
| As a Player,  I want to pause/continue the game in progress,  So that I can take breaks. | Criterion 1.  **IF** the character is alive…  **AND** the game isn’t paused…  **WHEN** the player clicks the “pause” button….  **THEN** pause the game.  Criterion 2.  **IF** the character is alive…  **AND** the game is paused…  **WHEN** the player clicks the “play” button….  **THEN** continue the game. |
| Priority: 3/5  Version: 1  Date: 17/11/21 |

### R13

|  |  |
| --- | --- |
| User Obstacle generation + R13 | Acceptance Criteria |
| As a Player,  I want obstacles to be generated in my path  So, the game becomes more challenging, and more pipe configuration are needed. | Criterion 1.  **IF** the player is alive and moving...  **THEN** new obstacles need to be generated continuously.  Criterion 2.  **IF** the player builds the pipe and hits an obstacle...  **WHEN** the character touches the obstacle...  **THEN** the player should die, and the game should be over. |
| Priority: 3/5  Version: 1  Date: 17/11/21 |

### R14

|  |  |
| --- | --- |
| User Story: Restart Game R14 | Acceptance Criteria |
| As a Player,  I want to restart the game,  So that I can quickly start the game over if I want to. | Criterion 1.  **IF** the character is alive...  **WHEN** I click the restart button...  **THEN** restart the game.  Criterion 2.  **IF** It’s game over screen...  **WHEN** I click the restart button...  **THEN** restart the game.  Criterion 3.  **IF** It’s pause screen...  **WHEN** I click the restart button...  **THEN** restart the game. |
| Priority: 2/5  Version: 1  Date: 17/11/21 |

### R15

|  |  |
| --- | --- |
| User Story: Go home R15 | Acceptance Criteria |
| As a Player,  I want to go to the home menu,  So that I can exit the game to show my high score. | Criterion 1.  **IF** the player is in pause or in game over screen...  **WHEN** I click the home button…  **THEN** take the player to the home menu. |
| Priority: 2/5  Version: 1  Date: 17/11/21 |

### R16

|  |  |
| --- | --- |
| User Story: Coin Collection R16 | Acceptance Criteria |
| As a Player,  I want to be able to collect coins to add a bonus to my score,  So that there is more incentive to move my pipe in certain directions. | Criterion 1.  **IF** the player is alive and moving...  **THEN** new coins need to be generated continuously.  Criterion 2.  **IF** the player builds the pipe to reach a coin...  **WHEN** the character touches the coin...  **THEN** the coin object should be removed, and a point bonus should be added to the score. |
| Priority: 2/5  Version: 1  Date: 17/11/21 |

### R17

|  |  |
| --- | --- |
| User Story: Special Pipes R17 | Acceptance Criteria |
| As a Player,  I want to be able to select pipes with special features, such as slowing down the game,  So that I have more time to choose my next move. | Criterion 1.  **IF** the player is alive and moving...  **THEN** a special pipe should be able to be selected if randomly generated, and it have the desired effect. |
| Priority: 1/5  Version: 1  Date: 17/11/21 |

### R18

|  |  |
| --- | --- |
| User Story: Beat Bonus R18 | Acceptance Criteria |
| As a Player,  I want to gain extra points by attaching my selected pipe piece in time with the music,  So that there is an added layer of difficulty but also a further way to increase my score. | Criterion 1.  **IF** the player is alive and moving...  **WHEN** the selected pipe is attached in time with the music...  **THEN** a point bonus should be added to the score. |
| Priority: 1/5  Version: 1  Date: 17/11/21 |

### R19

|  |  |
| --- | --- |
| User Story: Increasing Pipe Generation Speed R19 | Acceptance Criteria |
| As a Player,  I want the pipes to generate with increasing speed as I progress through the game,  So that there is less time to choose a correct pipe (increasing difficulty level) thus keeping the game engaging. | Criterion 1.  **IF** the player is alive and moving...  **THEN** incrementally increase the pipe generation speed. |
| Priority: 1/5  Version: 1  Date: 17/11/21 |

### R20

|  |  |
| --- | --- |
| User Story: Increase the Number of Pipes Generated R20 | Acceptance Criteria |
| As a Player...  I want more pipes to be generated at a time as I progress through the game,  So that it becomes less obvious which pipes to choose (increasing difficulty level) thus keeping the game engaging. | Criterion 1.  **IF** the player is alive and moving...  **THEN** incrementally increase the number of pipes generated at a time. |
| Priority: 1/5  Version: 1  Date: 17/11/21 |

### R21

|  |  |
| --- | --- |
| User Story: Move the camera to the right at an increasing velocity during gameplay R21 | Acceptance Criteria |
| As a Game Developer...  I want the player to have to keep up with the moving screen,  So that they have a goal and come across items in the game. | Criterion 1.  **IF** the player is alive…  **AND** the game isn’t paused….  **THEN** move the screen to the right at an increasing velocity. |
| Priority: 5/5  Version: 1  Date: 17/11/21 |

### R22

|  |  |
| --- | --- |
| User Story: Save the high score R22 | Acceptance Criteria |
| As a Player...  I want the game to remember and display my high score,  So that I can aim to beat it. | Criterion 1.  **IF** the player is alive…  **AND** the player has surpassed their previous high score  **THEN** save the new high score for display on the menu. |
| Priority: 3/5  Version: 1  Date: 17/11/21 |

### R23

|  |  |
| --- | --- |
| User Story: Exit the game from within the game R23 | Acceptance Criteria |
| As a Player...  I want to be able to close down the application from within the game,  So that I can easily close the application down without using my task manager. | Criterion 1.  **IF** the player is on the menu  **THEN** display an exit button that closes the application when clicked. |
| Priority: 3/5  Version: 1  Date: 17/11/21 |

## **User Story Tests**

The following test was completed at the end of Sprint 4. If the User Story’s acceptance criteria is fulfilled, the test is passed.

|  |  |  |
| --- | --- | --- |
| User Story | Test Result (acceptance criteria) | Reason for failure |
| R1 | Pass | - |
| R2 | Fail | No pipe highlighting for selected pipes. |
| R3 | Fail | Unable to add a pipe at all rotations that should be possible. |
| R4 | Pass | - |
| R5 | Pass | - |
| R6 | Fail | Can still rotate pipe pieces after death. |
| R7 | Fail | The pipe generation method has not been tuned to gameplay yet, so not enough pipes are generated. |
| R8 | Pass | - |
| R9 | Fail | The player’s high score is not saved so cannot be displayed. |
| R10 | Pass | - |
| R11 | Pass | - |
| R12 | Fail | No pause or resume button implemented. |
| R13 | Fail | No obstacle functionality implemented. |
| R14 | Fail | No restart buttons implemented. |
| R15 | Fail | No home buttons implemented. |
| R16 | Fail | No coin functionality implemented. |
| R17 | Fail | No special pipe functionality implemented. |
| R18 | Fail | No beat bonus functionality implemented. |
| R19 | Fail | Pipe generation does not increase in speed. |
| R20 | Fail | Number of pipes generated does not increase wit time. |
| R21 | Fail | Camera moves to right but not at increasing velocity. |
| R22 | Fail | High score is not saved. |
| R23 | Pass | - |

## **Use Cases**

Use Cases 1-16 were created this week. Please note that these Use Cases were created from the User Stories with more urgent priority. User Stories regarding additional features (e.g., special pipes) were left out as these were deemed unnecessary at this stage.

### Use Case 1

**UC1 - Version 1**

|  |  |  |
| --- | --- | --- |
| **UC1-1** | Use Case | Play the game (high-level) |
| **UC1-2** | Author | Leo Grant |
| **UC1-3** | Date | 18/11/21 |
| **UC1-4** | Purpose | For the user to play the game |
| **UC1-5** | Overview | Starts when player opens up the game file. The system displays a menu to the player. The player clicks the start button. The system starts the game. The player doesn’t achieve a high score else Alternative 1: The player achieves their high score. The system declares game over upon player death. |
| **UC1-6** | Cross Reference | R1, R10 |
| **UC1-7** | Actors | Player |
| **UC1-8** | Pre-Condition | UC1-Pre-1: The game has been installed successfully to a machine which is compatible with the game. |
| **UC1-9** | Post-Conditions | UC1-Post-1: The system is in a game over state. |

|  |  |  |
| --- | --- | --- |
| **Actor Actions**  1. Player loads up the game file.  3. Player clicks the start button.  5. Player plays the game. |  | **System Actions**  2. System loads home menu.  4. System starts the game.  6. System records player death has occurred.  7. System changes to game over state. |

|  |  |  |
| --- | --- | --- |
| **UC1-10** | Alternative Flow of Events: | * The player achieves their high score. Before Step 7, system should record the new high score. |
| **UC1-11** | Exceptional Flow of Events: | * None. |

### Use Case 2

**UC2 - Version 1**

|  |  |  |
| --- | --- | --- |
| **UC2-1** | Use Case | The player plays the game - Step 5 of UC1 - (low-level) |
| **UC2-2** | Author | Leo Grant |
| **UC2-3** | Date | 18/11/21 |
| **UC2-4** | Purpose | The player view should continuously move whilst the player is in game. |
| **UC2-5** | Overview | The system moves the camera view continuously to the right. The player clicks the pause button. System stops the camera moving. The player clicks the resume button. The system resumes the camera movement. The system eventually declares game over. The system stops the camera movement. |
| **UC2-6** | Cross Reference | R21 |
| **UC2-7** | Actors | Player |
| **UC2-8** | Pre-Condition | UC2-Pre-1: The game is playing.  UC2-Pre-1: The game is not paused. |
| **UC2-9** | Post-Conditions | UC2-Post-1: The player has played the game. |

|  |  |  |
| --- | --- | --- |
| **Actor Actions**  2. Player clicks pause button.  4. Player clicks resume button. |  | **System Actions**  1. System moves camera continuously to the right.  3. System stops camera movement.  5. System resumes camera movement.  6. System eventually declares game over.  7. System stops camera movement. |

|  |  |  |
| --- | --- | --- |
| **UC2-10** | Alternative Flow of Events: | * None. |
| **UC2-11** | Exceptional Flow of Events: | * None. |

### Use Case 3

**UC3 - Version 1**

|  |  |  |
| --- | --- | --- |
| **UC3-1** | Use Case | Selecting and changing selection of a pipe piece- Step 5 of UC1 - (medium-level) |
| **UC3-2** | Author | Leo Grant |
| **UC3-3** | Date | 18/11/21 |
| **UC3-4** | Purpose | For the user to choose which pipe piece to interact with. |
| **UC3-5** | Overview | Starts when the player starts or restarts the game. The system loads up the game. The player clicks on a pipe piece. The system records this pipe piece as only interactable pipe piece. The player clicks on a different pipe piece else Alternative 1: The player clicks on something that’s not selectable (pipe system or anything else not a pipe piece). The system records this pipe piece as only interactable pipe piece. |
| **UC3-6** | Cross Reference | R2 |
| **UC3-7** | Actors | Player |
| **UC3-8** | Pre-Condition | UC3-Pre-1: The game has just been started or restarted.  UC3-Pre-2: The game is not paused. |
| **UC3-9** | Post-Conditions | UC3-Post-1: A pipe piece is now interactable (player can add it to the pipe system or rotate it). |

|  |  |  |
| --- | --- | --- |
| **Actor Actions**  1. Player starts/restarts the game.  3. Player clicks on a pipe piece.  5. Player clicks a different pipe piece. |  | **System Actions**  2. System loads up the game.  4. System records this pipe piece as only interactable pipe piece.  6. System records this pipe piece as only interactable pipe piece. |

|  |  |  |
| --- | --- | --- |
| **UC3-10** | Alternative Flow of Events: | * The player clicks on something that’s not selectable (pipe system or anything else not a pipe piece). The system records that nothing is selected, wait at Step 2. |
| **UC3-11** | Exceptional Flow of Events: | * None. |

### Use Case 4

**UC4 - Version 1**

|  |  |  |
| --- | --- | --- |
| **UC4-1** | Use Case | Selecting and highlighting a pipe piece - Steps 3 to 6 of UC3 & Step 4 of UC7 – (low-level) |
| **UC4-2** | Author | Leo Grant |
| **UC4-3** | Date | 18/11/21 |
| **UC4-4** | Purpose | For the user to identify the selected pipe piece |
| **UC4-5** | Overview | Starts when the player clicks on a pipe piece. The system records this pipe piece as only interactable pipe piece. The system highlights the pipe piece to the player. The player clicks on a different pipe piece else Alternative 1: The player clicks on something that’s not selectable (pipe system or anything else not a pipe piece). The system records this new pipe piece as the only interactable pipe piece. The system de-activates the highlighting on the old, selected pipe piece, and highlights the new selected pipe piece. |
| **UC4-6** | Cross Reference | R2 |
| **UC4-7** | Actors | Player |
| **UC4-8** | Pre-Condition | UC4-Pre-1: The game has just been started or restarted.  UC4-Pre-2: The game is not paused.  UC4-Pre-3: The selected pipe pieces are all in the correct orientation to add them to the pipe system. |
| **UC4-9** | Post-Conditions | UC4-Post-1: The player has seen which pipe pieces he has selected and unselected via highlighting. |

|  |  |  |
| --- | --- | --- |
| **Actor Actions**  1. Player clicks on a pipe piece.  4. Player clicks on a different pipe piece.  7. Player presses return key. |  | **System Actions**  2. System records this pipe piece as only interactable pipe piece.  3. System highlights selected pipe piece.  5. System records this pipe piece as only interactable pipe piece.  6. System highlights selected pipe piece, and de-activates highlighting on previously selected pipe piece.  7. System adds the pipe piece to the pipe system.  8. System removes interactable quality of selected pipe piece (stops it being selected).  9. System removes the highlight of the just added pipe piece. |

|  |  |  |
| --- | --- | --- |
| **UC4-10** | Alternative Flow of Events: | * The player clicks on something that’s not selectable (pipe system or anything else not a pipe piece). The system records that nothing is selected and removes highlighting on that previously selected pipe piece. System waits for Step 1. |
| **UC4-11** | Exceptional Flow of Events: | * None. |

### Use Case 5

**UC5 - Version 1**

|  |  |  |
| --- | --- | --- |
| **UC5-1** | Use Case | Player finds new pipes to select (pipe generation) - Steps 1 to 4 of UC3 – (low-level) |
| **UC5-2** | Author | Marcus Williams |
| **UC5-3** | Date | 18/11/21 |
| **UC5-4** | Purpose | The player must find pipe pieces to select to add them to the pipe system and play the game. |
| **UC5-5** | Overview | Starts when the player starts or restarts the game. The system loads up the game. System instantiates pipe pieces into view for the player to select from. System moves camera to the right. System generates new pipe pieces off screen (for the camera to then move them into view). The player clicks on a generated pipe piece. The system records this pipe piece as only interactable pipe piece. The system eventually declares game over else Alternative 1: The game is paused. System stops pipe generation. |
| **UC5-6** | Cross Reference | R7 |
| **UC5-7** | Actors | Player |
| **UC5-8** | Pre-Condition | UC5-Pre-1: The player has the game file loaded up.  UC5-Pre-2: The system must have an array of possible pipe objects to select from. |
| **UC5-9** | Post-Conditions | UC5-Post-1: Pipe generation is stopped/paused.  UC5-Post-2: The player has selected a pipe piece. |

|  |  |  |
| --- | --- | --- |
| **Actor Actions**  1. Player starts/restarts the game.  6. Player clicks a pipe piece. |  | **System Actions**  2. System loads up the game.  3. System instantiates pipe pieces into view.  4. System moves camera to the right.  5. System generates new pipe pieces as the camera moves right.  7. System records this pipe piece as only interactable pipe piece.  8. System (eventually) declares game over.  9. System stops pipe generation. |

|  |  |  |
| --- | --- | --- |
| **UC5-10** | Alternative Flow of Events: | * The game is paused. System pauses any further pipe generation until game is resumed. |
| **UC5-11** | Exceptional Flow of Events: | * None. |

### Use Case 6

**UC6 - Version 1**

|  |  |  |
| --- | --- | --- |
| **UC6-1** | Use Case | Rotating the pipe pieces - Step 5 of UC1 - (medium-level) |
| **UC6-2** | Author | Marcus Williams |
| **UC6-3** | Date | 18/11/21 |
| **UC6-4** | Purpose | A player should be able to rotate pipes before adding them to the pipe system. |
| **UC6-5** | Overview | Starts when the player presses the right keypad else Alternative 1: Player presses left keypad. The system checks that a pipe piece is selected. A pipe piece is selected, else Alternative 2: A pipe piece is not selected. The system rotates the selected pipe piece clockwise 90 degrees. System updates information regarding that pipe’s rotation. |
| **UC6-6** | Cross Reference | R6 |
| **UC6-7** | Actors | Player |
| **UC6-8** | Pre-Condition | UC6-Pre-1: The game is playing.  UC6-Pre-2: The game is not paused.  UC6-Pre-3: A pipe piece has been selected. |
| **UC6-9** | Post-Conditions | UC6-Post-1: A pipe piece has been rotated has been rotated 90 degrees clockwise or anticlockwise. |

|  |  |  |
| --- | --- | --- |
| **Actor Actions**  1. Player presses right keypad. |  | **System Actions**  2. System validates there is a pipe piece selected.  3. System rotates that pipe piece 90 degrees clockwise.  4. System updates information about that pipe’s rotation. |

|  |  |  |
| --- | --- | --- |
| **UC6-10** | Alternative Flow of Events: | * The player presses left keypad. System performs same actions except Step 3 rotates pipe anti-clockwise 90 degrees. * A pipe piece is not selected. The system does nothing in response to the players keypad input. |
| **UC6-11** | Exceptional Flow of Events: | * Player presses both left and right keypad down at the same time. System ignores keypad input. |

### Use Case 7

**UC7 - Version 1**

|  |  |  |
| --- | --- | --- |
| **UC7-1** | Use Case | Adding the pipe pieces to the pipe system - Step 5 of UC1 - (medium-level) |
| **UC7-2** | Author | Leo Grant |
| **UC7-3** | Date | 18/11/21 |
| **UC7-4** | Purpose | A player should be able to add pipes to the pipe system to give the character something to move through. |
| **UC7-5** | Overview | Starts when the player presses the return key. The system checks that a pipe piece is selected. A pipe piece is selected else Alternative 1: A pipe piece isn’t selected. The system checks if it is in the correct orientation to add to the pipe system. The pipe piece is in the correct orientation else Alternative 2: The pipe piece is not in the correct orientation. The system adds the pipe piece to the pipe system. |
| **UC7-6** | Cross Reference | R3 |
| **UC7-7** | Actors | Player |
| **UC7-8** | Pre-Condition | UC7-Pre-1: The game is playing.  UC7-Pre-2: The game is not paused.  UC7-Pre-3: A pipe piece has been selected. |
| **UC7-9** | Post-Conditions | UC7-Post-1: A pipe piece has been added to the pipe system. |

|  |  |  |
| --- | --- | --- |
| **Actor Actions**  1. Player presses the return key. |  | **System Actions**  2. System validates there is a pipe piece selected.  3. System validates the pipe piece is in the correct orientation to fit onto the pipe system.  4. System adds the pipe piece to the end of the pipe system. |

|  |  |  |
| --- | --- | --- |
| **UC7-10** | Alternative Flow of Events: | * A pipe piece isn’t selected. The system does nothing in response to the players return key input. * Incorrect orientation of pipe piece for pipe system. The system does nothing in response to the players return key input. |
| **UC7-11** | Exceptional Flow of Events: | * Player presses both a left or right keypad input AND a return key input at the same time. System first rotates pipe, then attempts to add it to the system with new orientation. |

### Use Case 8

**UC8 - Version 1**

|  |  |  |
| --- | --- | --- |
| **UC8-1** | Use Case | Player makes character move – Continuation of Step 4 UC7 – (low-level) |
| **UC8-2** | Author | Leo Grant / Marcus Williams |
| **UC8-3** | Date | 18/11/21 |
| **UC8-4** | Purpose | The character should run to the end of the newly added pipe. |
| **UC8-5** | Overview | Starts when player presses return on a selected pipe piece. System adds a pipe to the pipe system. System moves character through the pipe to the end of the pipe system. System animates the character running. Character reaches the end of the pipe system else Alternative 1: the character dies before they reach the end of the pipe system. The character animation returns to an idle state. |
| **UC8-6** | Cross Reference | R3, R4, R11 |
| **UC8-7** | Actors | Player |
| **UC8-8** | Pre-Condition | UC8-Pre-1: The pipe piece selected is in the correct orientation to add it to the pipe system.  UC8-Pre-2: The game is in play.  UC8-Pre-3: The game isn’t paused.  UC8-Pre-4: The character isn’t moving. |
| **UC8-9** | Post-Conditions | UC8-Post-1: The character has run to the end of the newly added pipe in the pipe system. |

|  |  |  |
| --- | --- | --- |
| **Actor Actions**  1. Player presses return on selected pipe piece. |  | **System Actions**  2. System adds pipe piece to pipe system.  3. System moves character through the pipe, to the end of the pipe system, all the while animating the character running.  4. System returns character animation to an idle state. |

|  |  |  |
| --- | --- | --- |
| **UC8-10** | Alternative Flow of Events: | * The character dies. The system stops the character moving. The system plays a character death animation. |
| **UC8-11** | Exceptional Flow of Events: | * The system declares game over before the character reaches the end of the pipe system. The system should stop the character movement. |

### Use Case 9

**UC9 - Version 1**

|  |  |  |
| --- | --- | --- |
| **UC9-1** | Use Case | Pausing and resuming the game - Step 5 of UC1 - (medium-level) |
| **UC9-2** | Author | Zoe Broad |
| **UC9-3** | Date | 18/11/21 |
| **UC9-4** | Purpose | The player should be able to pause and resume the game. |
| **UC9-5** | Overview | Starts when the system loads the game. Player clicks on the pause button. System pauses the game and displays a pause screen. Player clicks resume button else Alternative 1: Player clicks the exit button else Alternative 2: Player clicks the restart button. The system resumes the game. |
| **UC9-6** | Cross Reference | R12, R14, R15 |
| **UC9-7** | Actors | Player |
| **UC9-8** | Pre-Condition | UC9-Pre-1: The character does not die before the player clicks the pause button. |
| **UC9-9** | Post-Conditions | UC9-Post-1: The game has resumed from the pause point. |

|  |  |  |
| --- | --- | --- |
| **Actor Actions**  2. The player clicks the pause button.  5. The player clicks the resume button. |  | **System Actions**  1. System loads up the game (gameplay)  3. The system pauses the game.  4. The system loads up the pause screen.  6. The system removes the pause screen.  7. The system resumes the game. |

|  |  |  |
| --- | --- | --- |
| **UC9-10** | Alternative Flow of Events: | * The player clicks the exit button. The system should take the player to the home menu. * The player clicks the restart button. The system restarts the game (start form Step 1). |
| **UC9-11** | Exceptional Flow of Events: | * A power cut means the game progress is lost and the player cannot select neither resume nor quit from the pause menu as it is no longer displayed. |

### Use Case 10

**UC10 - Version 1**

|  |  |  |
| --- | --- | --- |
| **UC10-1** | Use Case | Navigating to the home menu – UC1 Steps 1, 5, 7 - (medium-level) |
| **UC10-2** | Author | Matthew Fleischer |
| **UC10-3** | Date | 18/11/21 |
| **UC10-4** | Purpose | A player should be able to navigate to the home menu at any point in the game. |
| **UC10-5** | Overview | Starts when the character dies else Alternative 1: Starts when the game file is first loaded up, else Alternative 2: Starts when gameplay begins. System changes to a game over state. Player clicks the home button. System loads up the home menu. |
| **UC10-6** | Cross Reference | R10 |
| **U10-7** | Actors | Player |
| **UC10-8** | Pre-Condition | UC10-Pre-1: The character has just died. |
| **UC10-9** | Post-Conditions | UC10-Post-1: The home menu has been loaded up. |

|  |  |  |
| --- | --- | --- |
| **Actor Actions**  2. The player clicks on the home button. |  | **System Actions**  1. The system loads up the game over state.  3. System loads up the home menu. |

|  |  |  |
| --- | --- | --- |
| **UC10-10** | Alternative Flow of Events: | * The game file has just been loaded up. The system automatically displays the home menu. * The character is still alive and the game is in play. Follow UC9 with Alternative 1. |
| **UC10-11** | Exceptional Flow of Events: | * None. |

### Use Case 11

**UC11 - Version 1**

|  |  |  |
| --- | --- | --- |
| **UC11-1** | Use Case | Starting the game from the beginning – UC1 Steps 1, 5, 7 - (medium-level) |
| **UC11-2** | Author | Leo Grant |
| **UC11-3** | Date | 18/11/21 |
| **UC11-4** | Purpose | A player should be able to start the game from the beginning at any point in the game. |
| **UC11-5** | Overview | Starts when the character dies else Alternative 1: Starts when the game file is first loaded up, else Alternative 2: Starts when gameplay begins. System changes to a game over state. Player clicks the restart button. System loads up the game from the beginning. |
| **UC11-6** | Cross Reference | R1, R14 |
| **UC11-7** | Actors | Player |
| **UC11-8** | Pre-Condition | UC11-Pre-1: The character has just died. |
| **UC11-9** | Post-Conditions | UC11-Post-1: A new game has been loaded up. |

|  |  |  |
| --- | --- | --- |
| **Actor Actions**  2. The player clicks on the restart button. |  | **System Actions**  1. The system loads up the game over state.  3. System loads up a new game. |

|  |  |  |
| --- | --- | --- |
| **UC11-10** | Alternative Flow of Events: | * The game file has just been loaded up. The system automatically displays the home menu. Player clicks on start button, then move to Step 3. * The character is still alive and the game is in play. Follow UC9 with Alternative 2. |
| **UC11-11** | Exceptional Flow of Events: | * None. |

### Use Case 12

**UC12 - Version 1**

|  |  |  |
| --- | --- | --- |
| **UC12-1** | Use Case | Achieving a high score – UC1 Step 5 - (medium-level) |
| **UC12-2** | Author | Leo Grant |
| **UC12-3** | Date | 18/11/21 |
| **UC12-4** | Purpose | The system should recognise and store the player’s high score. |
| **UC12-5** | Overview | Starts when the player achieves their high score in game. The system validates that the score is the high score. The system saves the score as the new high score. |
| **UC12-6** | Cross Reference | R22 |
| **UC12-7** | Actors | Player |
| **UC12-8** | Pre-Condition | UC12-Pre-1: The game is playing.  UC12-Pre-2: The game is not paused. |
| **UC12-9** | Post-Conditions | UC12-Post-1: A new high score has been saved to the game. |

|  |  |  |
| --- | --- | --- |
| **Actor Actions**  1. Player achieves high score whilst playing the game. |  | **System Actions**  2. System validates that the score achieved is a high score.  3. System saves the current score as the high score. |

|  |  |  |
| --- | --- | --- |
| **UC12-10** | Alternative Flow of Events: | * None. |
| **UC12-11** | Exceptional Flow of Events: | * The player achieves their high score, and then the power cuts out. The system should have saved a reference to the new high score before the power cuts out. |

### Use Case 13

**UC13 - Version 1**

|  |  |  |
| --- | --- | --- |
| **UC15-1** | Use Case | Player loses the game – UC1 Step 5, 6 & 7- (medium-level) |
| **UC15-2** | Author | Zoe Broad |
| **UC15-3** | Date | 18/11/21 |
| **UC15-4** | Purpose | The player should be able to lose the game, otherwise it would become boring. |
| **UC15-5** | Overview | Starts when the system starts the game. System generates obstacles (similar to UC5 with pipes). The player adds a pipe to the system (see UC3,4,5,6, and 7). The system moves the character through the pipe (see UC8). The system is triggered when the character touches an obstacle else Alternative 1: The system is triggered when the character touches the left-hand side of the screen. The system declares a game over state. System stops generating new obstacles. |
| **UC15-6** | Cross Reference | R5, R13 |
| **UC15-7** | Actors | Player |
| **UC15-8** | Pre-Condition | UC15-Pre-1: The system is in the progress of starting the game. |
| **UC15-9** | Post-Conditions | UC15-Post-1: The system is in a game over state. |

|  |  |  |
| --- | --- | --- |
| **Actor Actions**  3. The player adds a pipe to the pipe system. |  | **System Actions**  1. The system starts a new game.  2. System starts obstacle generation.  4. The system adds a pipe to the pipe system.  5. The system moves the character through the pipe system.  6. The system is triggered when the character touches an obstacle.  7. The system changes to a game over state.  8. System stops obstacle generation. |

|  |  |  |
| --- | --- | --- |
| **UC15-10** | Alternative Flow of Events: | * The system is triggered when the character touches the left-hand side of the screen. Move to Step 6. |
| **UC15-11** | Exceptional Flow of Events: | * The system is triggered by an obstacle collision, and a screen collision. The system should move to Step 6. |

### Use Case 14

**UC14 - Version 1**

|  |  |  |
| --- | --- | --- |
| **UC14-1** | Use Case | Player loses game – Step 6 of UC15 – (low-level) |
| **UC14-2** | Author | Leo Grant |
| **UC14-3** | Date | 18/11/21 |
| **UC14-4** | Purpose | Upon character death, the game over options must be presented, and interaction with the lost game prohibited. |
| **UC14-5** | Overview | Starts when the character has just died. The system stops camera movement. The system stops pipe selection. The system stops pipe rotation (of existing selection). The system stops pipe adding (to the pipe system). The system loads up a game over menu. The player tries to interact with the pipes. The system prevents this. Alternative 1: The player clicks on the home menu button. |
| **UC14-6** | Cross Reference | R4, R6, R15 |
| **UC14-7** | Actors | Player |
| **UC14-8** | Pre-Condition | UC14-Pre-1: The character has just died.  UC14-Pre-2: The player had a pipe selected before they died.  UC14-Pre-3: The selected player pipe was in the correct orientation to be added to the pipe system. |
| **UC14-9** | Post-Conditions | UC14-Post-1: The system has loaded up the game over state. |

|  |  |  |
| --- | --- | --- |
| **Actor Actions**  6. The player presses the left or right keypad button.  7. The player presses the return key.  8. The player clicks on a different pipe piece. |  | **System Actions**  1. The system records the character has died.  2. The system stops camera movement.  3. The system stops character movement.  4. The system stops pipe selection, rotation, and addition to the pipe system.  5. The system loads up the game over menu.  6. No response  7. No response.  8. No response. |

|  |  |  |
| --- | --- | --- |
| **UC14-10** | Alternative Flow of Events: | * The player clicks on something that’s not selectable (pipe system or anything else not a pipe piece). The system records that nothing is selected, wait at Step 2. * The player clicks on the home menu button. The system loads up the home menu. |
| **UC14-11** | Exceptional Flow of Events: | * None. |

### Use Case 15

**UC15 - Version 1**

|  |  |  |
| --- | --- | --- |
| **UC15-1** | Use Case | Player increases their score – UC1 Step 5 - (medium-level) |
| **UC15-2** | Author | Leo Grant |
| **UC15-3** | Date | 18/11/21 |
| **UC15-4** | Purpose | The player should be able to increase their score to aim to achieve a high score. |
| **UC15-5** | Overview | Starts when the system loads up a new game. The player adds a pipe piece (see UC3,4,5,6, and 7). The system moves the character through the pipe (see UC8). The system increases the score of the player relative to the maximum right horizontal displacement reached by the character. The player adds another pipe. The system moves the character through this new pipe. The system is triggered by the collision of the player with a coin. The system adds extra points to the player’s score. |
| **UC15-6** | Cross Reference | R8, R16 |
| **UC15-7** | Actors | Player |
| **UC15-8** | Pre-Condition | UC15-Pre-1: The system is in the progress of starting the game. |
| **UC15-9** | Post-Conditions | UC15-Post-1: The player’s score has increased |

|  |  |  |
| --- | --- | --- |
| **Actor Actions**  2. The player adds a pipe to the pipe system.  6. The player adds another pipe to the pipe system (through the path of a coin). |  | **System Actions**  1. The system loads up a new game.  3. The system adds a pipe to the pipe system.  4. The system moves the character through the pipe system.  5. The score increases relative to the maximum right horizontal displacement reached by the character.  7. The system adds the pipe to the pipe system.  8. The system moves the character through the pipe system.  9. The system is triggered by the collision of the player with a coin, and adds extra points to the player’s score. |

|  |  |  |
| --- | --- | --- |
| **UC15-10** | Alternative Flow of Events: | * None. |
| **UC15-11** | Exceptional Flow of Events: | * None. |

### Use Case 16

**UC15 - Version 1**

|  |  |  |
| --- | --- | --- |
| **UC16-1** | Use Case | Player exits the game application – UC1 Step 5 - (medium-level) |
| **UC16-2** | Author | Leo Grant |
| **UC16-3** | Date | 18/11/21 |
| **UC16-4** | Purpose | The player should be able to exit the application from the home menu. |
| **UC16-5** | Overview | Starts when the system loads up the home menu. The player clicks on the exit button. The system closes down the application. |
| **UC16-6** | Cross Reference | R23 |
| **UC16-7** | Actors | Player |
| **UC16-8** | Pre-Condition | UC16-Pre-1: The system is in the progress of loading up the home menu. |
| **UC16-9** | Post-Conditions | UC16-Post-1: The game has been shut down. |

|  |  |  |
| --- | --- | --- |
| **Actor Actions**  2. Player clicks on the exit button. |  | **System Actions**  1. The system loads up home menu.  3. System closes the application. |

|  |  |  |
| --- | --- | --- |
| **UC16-10** | Alternative Flow of Events: | * None. |
| **UC16-11** | Exceptional Flow of Events: | * None. |

## **Use Case Tests**

The following test was completed at the end of Sprint 4. The test passes if the use case runs as defined.

|  |  |  |
| --- | --- | --- |
| Use Case | Test Result | Reason for failure |
| UC1 | Fail | The high score has not been recorded by the system. |
| UC2 | Pass |  |
| UC3 | Pass |  |
| UC4 | Fail | Pipe piece is not highlighted when selected. |
| UC5 | Pass |  |
| UC6 | Pass |  |
| UC7 | Fail | Not all correctly orientated pipe pieces may be added to the pipe system. |
| UC8 | Fail | Character still moves to the end of the pipe when dead. |
| UC9 | Fail | Pausing and resuming not implemented. |
| UC10 | Fail | No home button to take player to main menu from game over state. |
| UC11 | Fail | No way to restart the game from the beginning whilst in game. |
| UC12 | Fail | System does not validate or store high score. |
| UC13 | Pass | - |
| UC14 | Fail | Character movement, pipe selection, pipe rotation and pipe addition may all still occur after character death. |
| UC15 | Fail | Coin functionality not implemented. |
| UC16 | Pass | - |

## **CRC Cards**

All of the CRC cards below were created this week.

### character\_Controller

|  |  |
| --- | --- |
| **Class Name:** character\_Controller | |
| **Version:** 1 | |
| **Cross Reference:** UC1, UC8, UC13 | |
| **Responsibilities:**  Move character (i) & (ii)  Detect character death (game over)  Declare game over (ii) | **Collaborators:**  (i) pipe\_System  (ii) game\_State\_Controller  (iii) animation\_Controller |

Description:

1. pipe\_System adds movement positions to a queue for the character to move through.
2. Character movement may be paused if the game state is paused, or in game over. Also game\_State changed to game over upon character death.
3. The state of the character will be identified in the character\_Controller class using bools: isRunning, and isGameOver to identify when the character should animate running, death, and standing still.

### game\_State\_Controller

|  |  |
| --- | --- |
| **Class Name:** game\_State\_Controller | |
| **Version:** 1 | |
| **Cross Reference:** UC1, UC9, UC10, UC13, UC14 | |
| **Responsibilities:**  Set game\_State (“menu” (ii), “play” (i)(ii)(iii)(iv), “game\_over” (iii)(iv), “pause” (ii)(iii)(iv)) | **Collaborators:**  (i) camera\_Controller  (ii) pipes\_Interface  (iii) character\_Controller  (iv) game\_UI\_Controller  (v) obstacle\_Generation  (vi) coins\_Controller  (vii) score  (viii) pipe\_Generation |

Description:

1. The camera moves if the game\_State == “play”.
2. The player is stopped from selecting, rotating and adding pipe pieces to the system if game\_State != “play”.
3. The character stops moving if game\_State == “pause” or “game over”, and continues moving to its target position if game\_State == “play”.
4. The game\_UI\_Controller displays the game over UI when game\_State == “game\_over”. Additionally, the game\_UI\_Controller can set the game\_State to “pause” and then back to “play” or “menu” depending on user input. The pause screen should be blocked from access when game\_State == “game\_over.”
5. Obstacle\_Generation only create new obstacles in a game\_State of “play”.
6. coins\_Controller should only be generate new coins in a game\_State of “play”.
7. The score is only calculated and displayed to the user whilst the game\_State == “play”.
8. pipe\_Generation should only instantiate pipes whilst the game\_State == “play”.

### pipe\_System

|  |  |
| --- | --- |
| **Class Name:** pipe\_System | |
| **Version:** 1 | |
| **Cross Reference:** UC1, UC7, UC8 | |
| **Responsibilities:**  Hold the pipe system (specifically, all the checkpoint game objects through which the character travels).  Add pipes to the pipe system (i) & (ii)  Add pipe checkpoints to the character\_Controller movement queue.  Checks if submitted pipe piece can be added to the pipe system. (i) | **Collaborators:**  (i) pipes\_Interface  (ii) pipe\_Properties  (iii) character\_Controller |

Description:

1. The pipes\_Interface sends the pipe\_System a pipe game object to add to the system. The pipe\_System checks it may be added and adds it if possible.
2. The pipe\_System uses detail from the pipe\_Properties class on that pipe to check if the pipe can be added to the pipe system.
3. pipe\_System adds movement positions to a queue for the character to move through.

### pipes\_Interface

|  |  |
| --- | --- |
| **Class Name:** pipes\_Interface | |
| **Version:** 1 | |
| **Cross Reference:** UC3, UC6 & UC7 | |
| **Responsibilities:**  Select pipe pieces (iii)  Checks for input to rotate Pipe pieces (i) & (iii)  Checks for input to add pipes to the pipe system (ii) & (iii) | **Collaborators:**  (i) pipe\_Properties  (ii) pipe\_System  (iii) game\_State\_Controller |

Description:

1. If input found by the pipes\_Interface class is to rotate the selected pipe, send this message on to the specific pipe object’s pipe\_Properties to rotate it. Additionally, tell pipe\_properties to highlight the pipe piece or un-highlight it (depending on user selection).
2. If input found by the Pipes Interface class is to add the selected pipe to the pipe system, send this message on to the pipe\_System class to do so (if possible), with a reference to the specific pipe object in question.
3. A pipe may not be selected or de-selected if game\_State doesn’t equal “play”. This barrier to selection entry should also apply to rotation and addition of pipes.

### pipe\_Properties

|  |  |
| --- | --- |
| **Class Name:** pipe\_Properties | |
| **Version:** 1 | |
| **Cross Reference:** UC4, UC6 | |
| **Responsibilities:**  Stores information about pipe rotation. (ii)  Rotates the pipe piece. (i)  Highlights / un-highlights the pipe piece. (i) | **Collaborators:**  (i) pipes\_Interface  (ii) pipe\_System |

Description:

1. The pipes\_Interface will send a message to the pipe\_Properties script to rotate the game object it’s attached to, upon user input. The pipe\_Properties class will update its information about the rotation of the pipe. Additionally, the pipes\_Interface will tell the pipe\_Properties class to highlight or de-highlight itself, depending on user selection status in the pipes\_Interface class.

### game\_UI\_Controller

|  |  |
| --- | --- |
| **Class Name:** game\_UI\_Controller | |
| **Version:** 1 | |
| **Cross Reference:** UC1, UC9, UC10, UC11, UC16 | |
| **Responsibilities:**  Menu (ii)  Display high score on menu (i)  Start game (ii)  Exit game (ii)  Pause game (ii)  Continue game (ii)  Restart game (ii)  Return to menu (at game over and at pause) (ii)  Display current score in game | **Collaborators:**  (i) score  (ii) game\_State\_Controller |

Description:

1. The score class stores the high score which is accessed by game\_UI\_Controller to display this number on the menu.
2. The game state should be changed by the game\_UI\_Controller when the buttons on the menu (start, exit), whilst playing (pause, continue, restart) are pressed, and on the game over screen (restart, return to menu)

### score

|  |  |
| --- | --- |
| **Class Name:** score | |
| **Version:** 1 | |
| **Cross Reference:** UC1, UC12, UC15 | |
| **Responsibilities:**  Store high score  Calculate score during play (i)  Display score during play (i) | **Collaborators:**  (i) game\_State\_Controller |

Description:

1. The score is only calculated and displayed to the user whilst the game\_State == “play”.

### pipe\_Generation

|  |  |
| --- | --- |
| **Class Name:** pipe\_Generation | |
| **Version:** 1 | |
| **Cross Reference:** UC1, UC5 | |
| **Responsibilities:**  Instantiate a set of initial pipes when the game starts  Create new pipes as the game progresses (i)  Decide what pipes need to be added to make sure progress is not impossible | **Collaborators:**  (i) game\_State\_Controller |

Description:

1. Instantiates pipes whilst the game\_State == “play”.

### animation\_Controller

|  |  |
| --- | --- |
| **Class Name:** animation\_Controller | |
| **Version:** 1 | |
| **Cross Reference:** UC1, UC8 | |
| **Responsibilities:**  Run the appropriate character animation based on the state of the character (i) (I.e. running, standing still, climbing, or dead) | **Collaborators:**  (i) character\_Controller |

Description:

1. The state of the character will be identified in the character\_Controller class using bools: isRunning, and isGameOver to identify when the character should animate running, death, and standing still.

### coins\_Controller

|  |  |
| --- | --- |
| **Class Name:** coins\_Controller | |
| **Version:** 1 | |
| **Cross Reference:** UC1, UC15 | |
| **Responsibilities:**  Create new coins as the game is played (ii)  Delete coin objects if the character collides with them.  Update the score (i) | **Collaborators:**  (i) score  (ii) game\_State\_Controller |

Description:

1. The coin class should identify when it collides with the character, and send this info to the score class.
2. New coins should only be generated in a game\_State of “play”.

### obstacle\_Generation

|  |  |
| --- | --- |
| **Class Name:** obstacle\_Generation | |
| **Version:** 1 | |
| **Cross Reference:** UC1,UC13 | |
| **Responsibilities:**  Create new obstacles as the game progresses (i)  When the character collides with an obstacle the game should be over (i) & (ii) | **Collaborators:**  (i) game\_State\_Controller  (ii) character\_Controller |

Description:

1. Only create new obstacles in a game\_State of “play”.
2. If the character hits and obstacle, change game\_State to “game\_over”, and change isGameOver character bool to true for the death animation to run.

### camera\_Controller

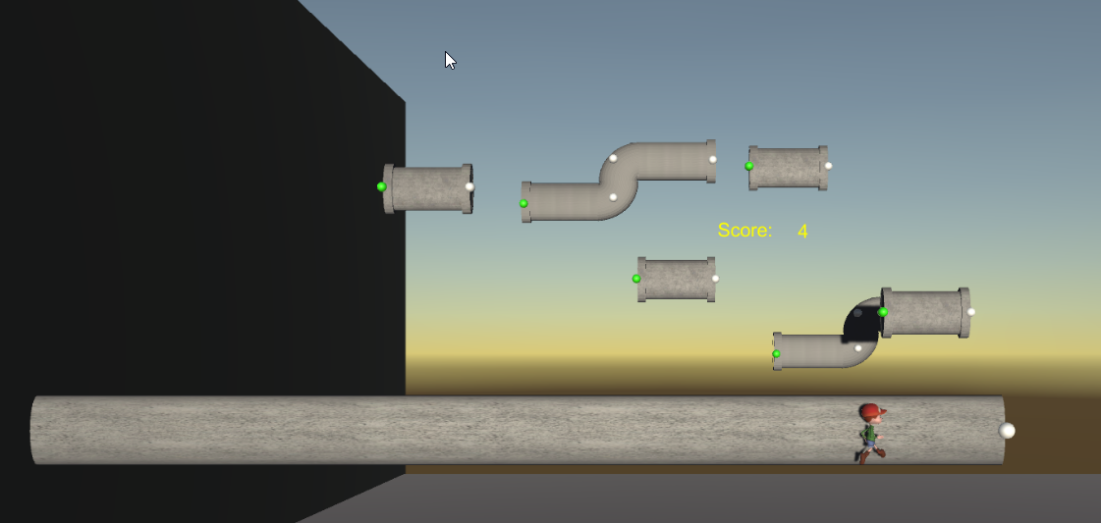
|  |  |
| --- | --- |
| **Class Name:** camera\_Controller | |
| **Version:** 1 | |
| **Cross Reference:** UC1,UC2 | |
| **Responsibilities:**  Move horizontally (left to right) at a steadily increasing speed (i) | **Collaborators:**  (i) game\_State\_Controller |

Description:

1. Only move right if the game\_State == “play”.

## **User interface design**

### Game Progress



### Dark Tunnel Style



### Dungeon entry style



### Mysterious Gates Style



## **Exception Handling**

The challenges we faced during this sprint’s tasks are outlined below, alongside their respective solutions:

|  |  |  |
| --- | --- | --- |
| Tasks | Challenges | Solutions |
| S4.1- T1 | Capturing every requirement for our project. | Use the customer meetings for guidance and allow for versioning and addition of user stories in the future. |
| S4.1- T2 | Ensuring that the Use Case was correctly presented. | The documentation examples provided in moodle was useful to model our approach. |
| S4.1- T3 | Capturing all functionality of a class. | Just getting something down was useful. We could then tidy up from there. |
| S4.1- T4 | Defining the roles. | Long discussion to outline the responsibilities of each role. |
| S4.1- T5 | Deciding where to generate pipes so they don’t overlap too much. | Create an empty game object to base the positioning of newly generated pipes from, give set positions of starting pipes and increase the time between pipe generations. |
| S4.1-T6 | N/A No challenge. | - |
| S4.1-T7 | N/A No challenge. | - |
| S4.1-T8 | N/A No challenge. | - |
| S4.2- T1 | 1)Remembering what I had originally created, and how it worked together.  2)Gradually adapting the mock-up to the new format. | 1)Reading the comments.  2)Regular testing that the game still compiles and runs as expected in the Unity Editor. |
| S4.2- T2 | Creating logic to identify if a pipe may be added on to the pipe system. | Incremental changes with an incremental progress aim (e.g., first focus on the logic for just one end of the pipe fitting on to the pipe system before considering the second). |
| S4.2- T3 | Positioning the score in the screen was difficult as it appeared in different positions to each dev. Team member. | Everyone use a fixed screen ratio when developing the game. |
| S4.2- T4 | Creating buttons, at first 3d text objects were used for aesthetic reasons but there were problems implementing the functionality. | The UI button object replaced it as it was much more intuitive, and while it had a predetermined look. Was much faster to work with. |
| S4.2- T5 | Similar issues to pipe generation. | Implement same process as for pipe generation. |
| S4.2- T6 | N/A No challenges. | N/A |
|  | Team member don’t have work transparency in each other’s work. | Shawn updates JIRA board every week with tasks so that everyone’s work is transparent to each other. |