

EWB PDF Report – Team 2, Semester 2

Peer Assessment 2

Evaluation		Group Number: 2		
Name	Contribution to team-working and motivation ¹	Contribution to the Poster and the Poster Video, PDF Report 2, and the Final Game Demo Video and Game Demo Schedule ^{1,2}	Contribution to methodically developed and functioning game code ^{1,2}	Peer Score (Range 85 – 115) You must fill this in as well as the other columns.
Peter Robinson	4	5	3	109
Isaac Edmonds	4	3	4	110
Nathan Watkins	4	2	5	112
Curtis McCartney	4	5	4	111
Antons Bogdanovs	4	4	2	107
Kal Worthington	4	4	2	108

¹Values for contribution: 1 = Minimal Contribution; 2 = Reasonable Contribution; 3 = Good Contribution; 4 = Very Good Contribution;
5 = Excellent Contribution

²This value should consider contributions in the round – direct contributions this semester to required deliverables, and contributions this semester that have made the deliverables possible.

Declaration		
"I declare that I have read the Queen's University regulations on plagiarism, and that any contribution I have made to the attached submission is my own original work, except for any elements that I have clearly attributed to third parties. I understand that this submission will be subject to an electronic test for plagiarism and will also be subject to the University's regulations concerning late submission if it is received after the deadline."		
Name	Date	Confirmation
Peter Robinson	14/03/2025	I agree to the terms of the declaration
Isaac Edmonds	14/03/2025	I agree to the terms of the declaration
Nathan Watkins	14/03/2025	I agree to the terms of the declaration
Curtis McCartney	14/03/2025	I agree to the terms of the declaration
Antons Bogdanovs	14/03/2025	I agree to the terms of the declaration
Kal Worthington	14/03/2025	I agree to the terms of the declaration

Personal Statements

<i>Personal statement of Peter Robinson:</i>	
The following were my most significant contributions to the Semester 2 Deliverable (100 words or less):	
With Isaac, developed and maintained the GameSystem class, linking various components for game functionality. Ensured the GameSystem behaved as expected by writing extensive JUnit test cases to achieve maximum coverage. Implemented backend logic for the tutorial, endgame and endgame achievements. With Kal, for the PDF report, wrote up the UI design detailing and reasoning why our game takes the current form. Recorded segments for poster and game demo video.	

<i>Personal statement of Isaac Edmonds:</i>	
The following were my most significant contributions to the Semester 2 Deliverable (100 words or less):	
Worked on a lot of the initial GameSystem class foundation with Peter. Was responsible for designing and updating the UML class diagram throughout semester 2 and created the Task class along with Curtis. I also designed multiple methods for use with Nathan's UI classes to deliver core features to the system, such as spending resources on progressing tasks, buying resources, and calculating the solution implementation. I performed testing on the Objective, Task, and SubTask classes and improved the way the system handled different types of resources.	

<i>Personal statement of Nathan Watkins:</i>	
The following were my most significant contributions to the Semester 2 Deliverable (100 words or less):	
Developed all the popups needed for the game such as the Journal, Shop and EndGame among others. Developed the front-end GUI for a fully functional game including a start screen and efficient resource loading. Helped with other back-end coding where needed. Also completed the Final Game Layout for the semester 2 PDF Report.	

Personal statement of Curtis McCartney:

The following were my most significant contributions to the Semester 2 Deliverable (100 words or less):

Over semester 2, I had to rework the square classes to comply with our new game idea, including new Task and Money Squares. With methods created by Isaac and Nathan, I was able to implement the ability to acquire, progress, discount and transfer Tasks, along with creating the generation of Money Squares across the board, enabling the main gameplay through the use of Popups. Finally, I have been solely responsible for the Acceptance Test Plan and the Surveys necessary for addressing opinions over our GUI and editing together the final Game Demo and Poster videos.

Personal statement of Antons Bogdanovs:

The following were my most significant contributions to the Semester 2 Deliverable (100 words or less):

During semester 2 of this assignment, I have spent most of my time working closely with Kal Worthington creating the Square class and its subclasses, writing the methods and ensuring compatibility with the rest of the program, and after this we worked together to write up and design the A1 Poster discussing the solution we created for the pothole problem in makers valley.

Personal statement of Kal Worthington:

The following were my most significant contributions to the Semester 2 Deliverable (100 words or less):

In Semester 2, I worked with Peter Robinson on the User Interface design section of the PDF report. Earlier, I collaborated with Antons Bogdanovs to develop the Square class and its subclasses, writing methods and ensuring they integrated smoothly with the program. Together, we also designed the A1 Poster, developing and presenting our solution to the pothole issue in Maker's Valley.

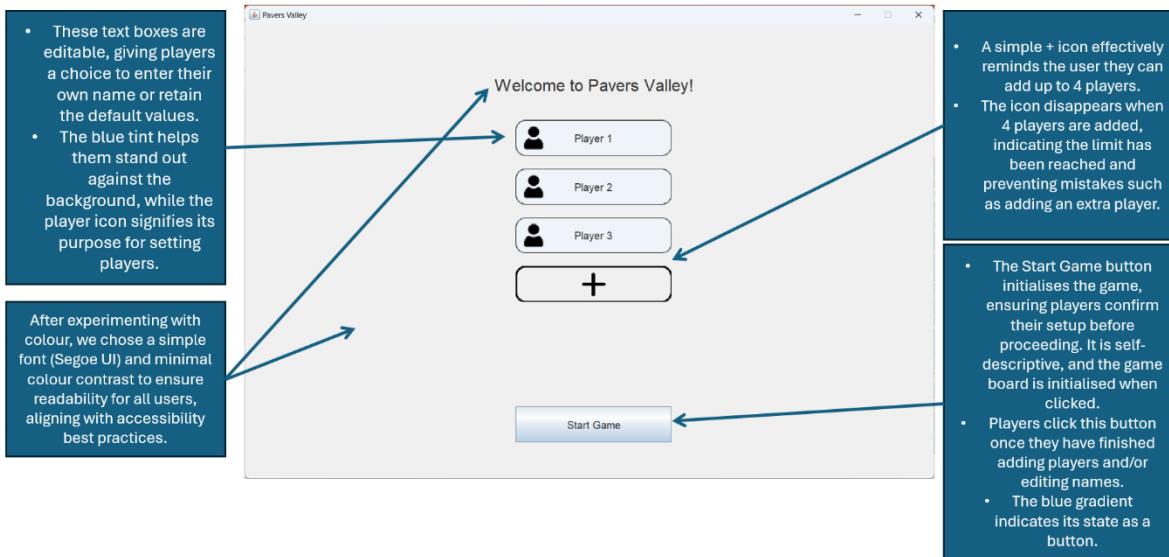
Design Documentation

User Interface Design

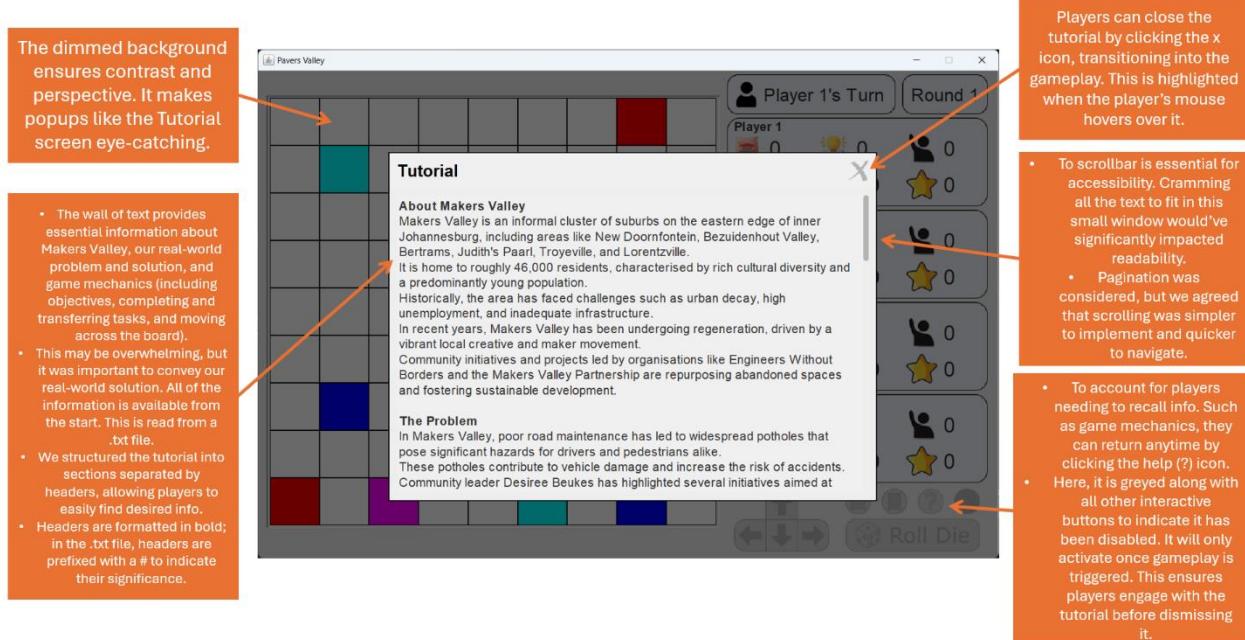
[Principal: P.R., K.W.] [Support: N.W.]

Note: Due to Java Swing limitations, rasterized images (PNG, JPG, etc.) used in the game will appear pixelated when the user's system scale is set to above 100%. A solution to this is to use vector images such as SVG, however this itself has limitations such as complex implementation and extra dependencies. Therefore, we chose to settle for the chance of having pixelated images instead of the difficult vector implementation.

Start Screen

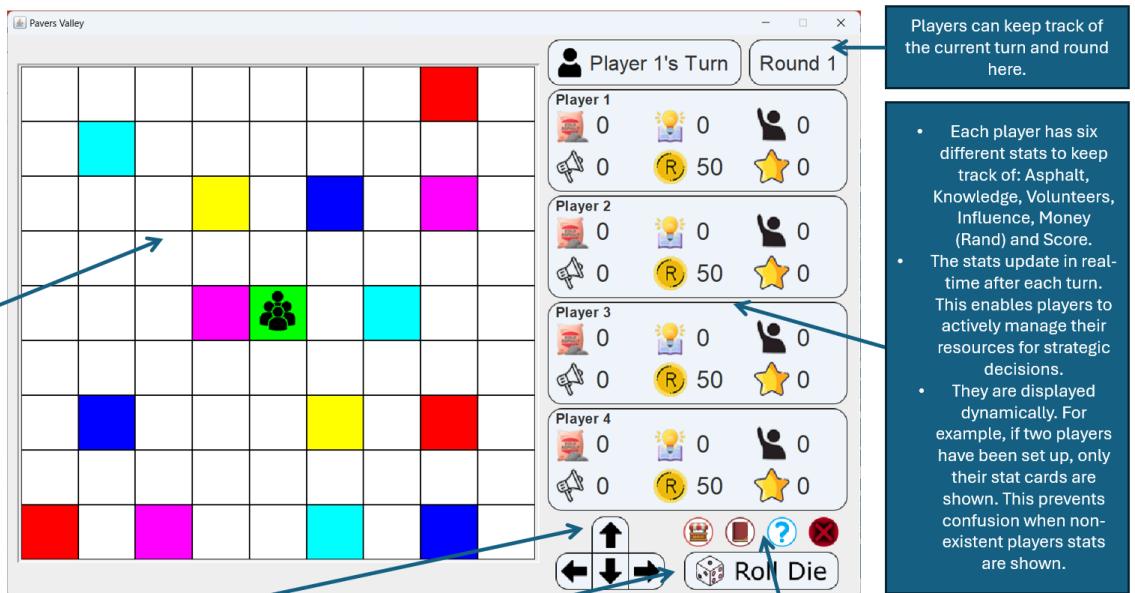


Tutorial



Main Window

- The board is a 9x9 grid. Earlier versions were 12x12, but it was agreed a smaller board would improve clarity.
- The green square, in the centre of the board, is a start square. Players pick up their resources here from the shop (explained later).
- The yellow squares are money squares, mimicking the colour of a gold coin. Players travel to these squares to receive funding, which is vital for maintaining the solution implementation.
- The other squares are task squares. Players travel to these squares to assume tasks. Each colour is related to a specific objective.
- There are 3 tasks for each objective. For instance, the red squares are tasks for the "Secure Grant" objective.
- Players can move onto any square and interact with it; an exact roll is not required, as that felt too restrictive.



- These buttons enable players to move across the board and make strategic decisions (e.g. claiming a task).
- Players click the "Roll Die" button and then move using the arrow buttons. The icons are simple and self-descriptive. The on-screen arrow buttons allow compatibility with touchscreen devices.

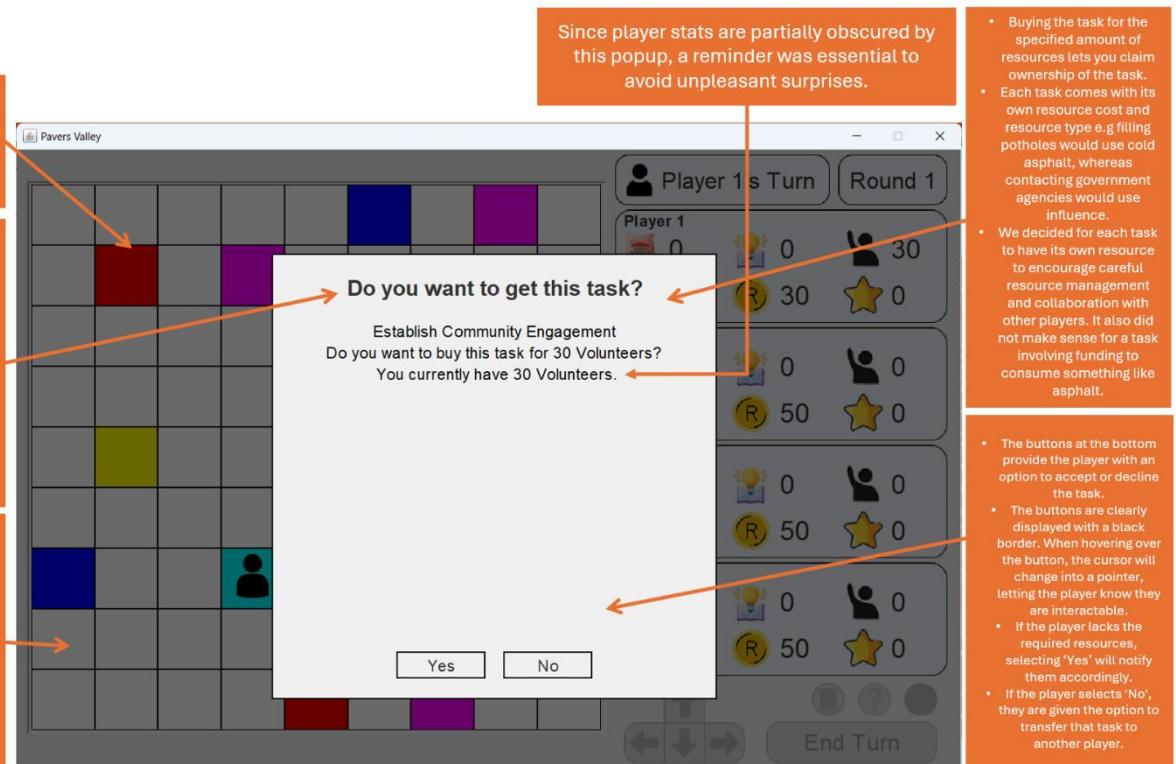
- These buttons also enable players to make decisions: access the shop to buy resources (if on the start square), journal to access, complete or transfer tasks, help (to recall info.) and quit the game (if they have given up).
- The icons are small but meaningful. The stall implies a shop, book implies a journal, the ? implies help and the red x icon implies quitting.
- As quitting is a consequential action, the UI prompts confirmation before triggering the bad ending. This reinforces error prevention.

Prompts/Popups

The dimmed background again ensures contrast and perspective. It makes the pop-up stand out.

- The sub header of the pop-up displays the name of the task.
- Each task comes with 3 subtasks, and the overarching goal of the task is implied by the colour of the square and can be viewed in the journal (explained later).
- These tasks are read in from a .json file.

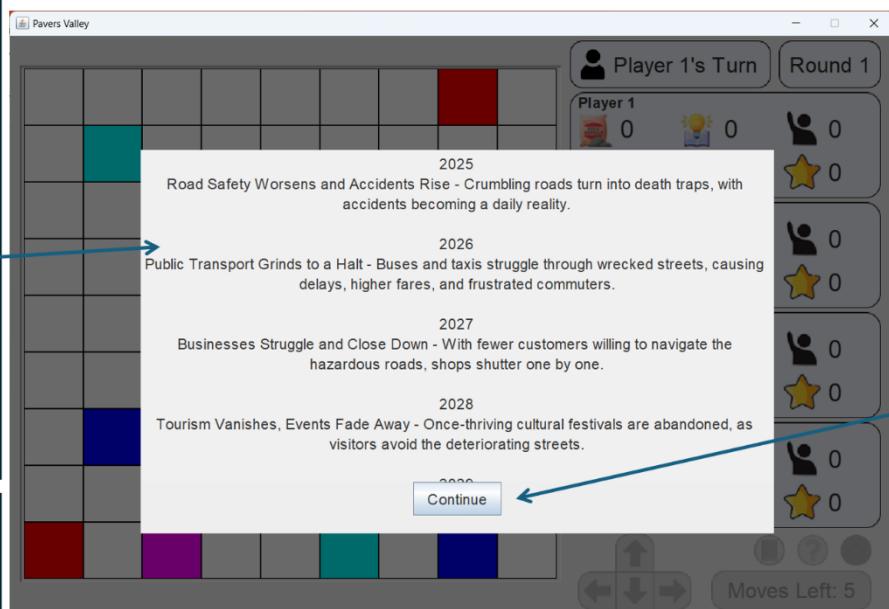
Other square types give different options e.g. the money square, which prompts the player to accept a random amount of money when landing on it. The money is presented with a story on how it was obtained e.g. IIPSA grant was successful.



Endgame Screen

- No matter the outcome of the game, a slow scrolling wall of text will appear as a pop-up upon the end of the game.
- If the players were successful in their objective of repairing the potholes, a positive dialogue is shown detailing the results of their teamwork.
- However, in this example, the players were unsuccessful in their goal and the consequences of their inadequacy are on full display.

This text acts as an epilogue, as specified in the core requirements.

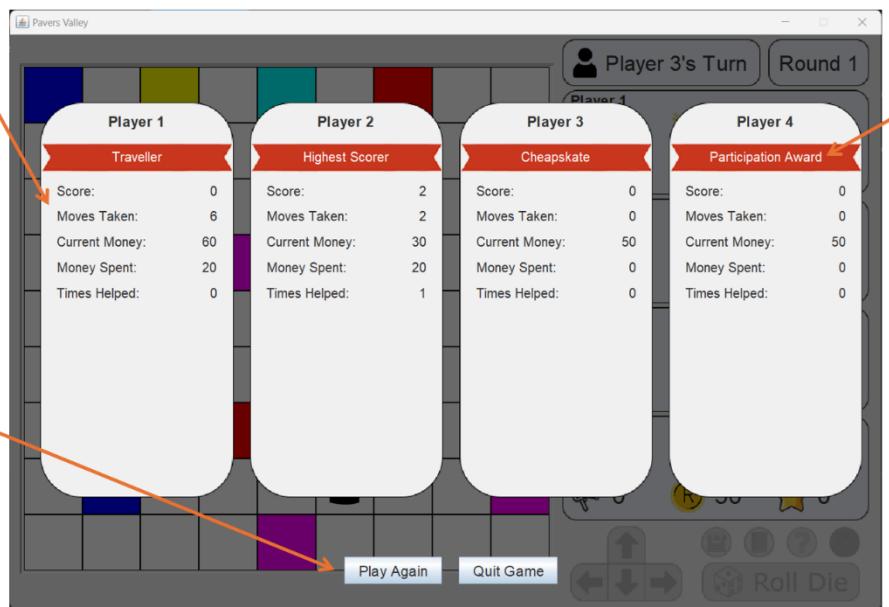


- This pop-up gives players a chance to relax at the end of the game and look back on their hard work (or lack thereof).
- The text is read in from a .txt file.
- The slow scroll speed allows players to read and digest the information in plenty of time.

- The button at the bottom clearly shows that it can be interacted with through the blue gradient, and it will highlight when a mouse hovers over it.
- This can be pressed at any time and will immediately advance players to the statistics section.

The final screen of the game is the statistics page. This displays each player's tracked stats at the end of the game.

The buttons at the bottom, clearly highlighted by their blue gradient, give the player the option to either terminate the game or play another round.



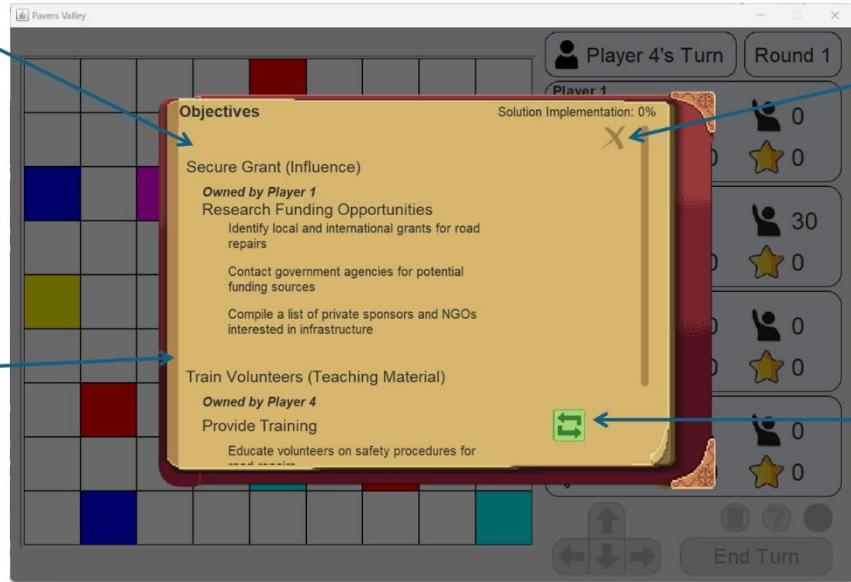
- This will display an award for each player based on their individual stats, where the title will be a pun/play on words based on the category.

- If a player shows no dominance in any stat, they will be given an arbitrary award (along the lines of a wooden spoon).
 - The title is highlighted by a red banner to improve readability and emphasize the award.
 - The ribbon shape also stands out against the rounded player summaries.

Journal

The journal shows all the tasks needed to be completed for the game to end successfully.

- The aesthetic of the journal is made to feel as if you are reading through a real journal.
- However, a scrollable element was used instead of pages to improve accessibility and readability.
- Each goal, task and subtask are separated clearly by tab indentation.
- The journal is initially empty. As players take ownership of tasks, they appear in the journal.
- The player who owns a task will have their name displayed alongside the task.



- The X in the top right corner lets the player exit the journal, as the journal can be accessed at any time. It is highlighted red when the mouse hovers over it.

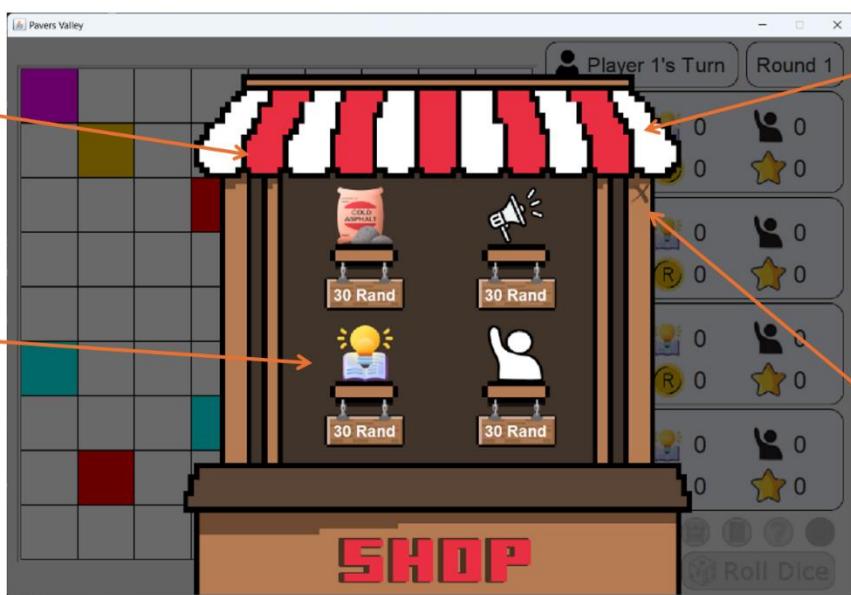
- Just above the X is the implementation tracker which tracks the percentage of subtasks completed. This tracker allows players to easily see how far along they are.

- As well as the task ownership cost, resources need to be dedicated to complete the task.
- If the player thinks a task will not be completed by them (they can't acquire enough of a specific resource or they are in a bad position on the board), they can use the green button to transfer ownership.
- The loop icon is used to imply a swap. This encourages collaboration between players.

Shop

The shop can be accessed when on the start square, which then shows an interface where the player can spend their Rand on various resources needed to complete tasks.

- When selected, each item will be purchased for the listed price underneath.
- Each symbol corresponds to a resource which will be added to a player's tracked resources after taking away their Rand.

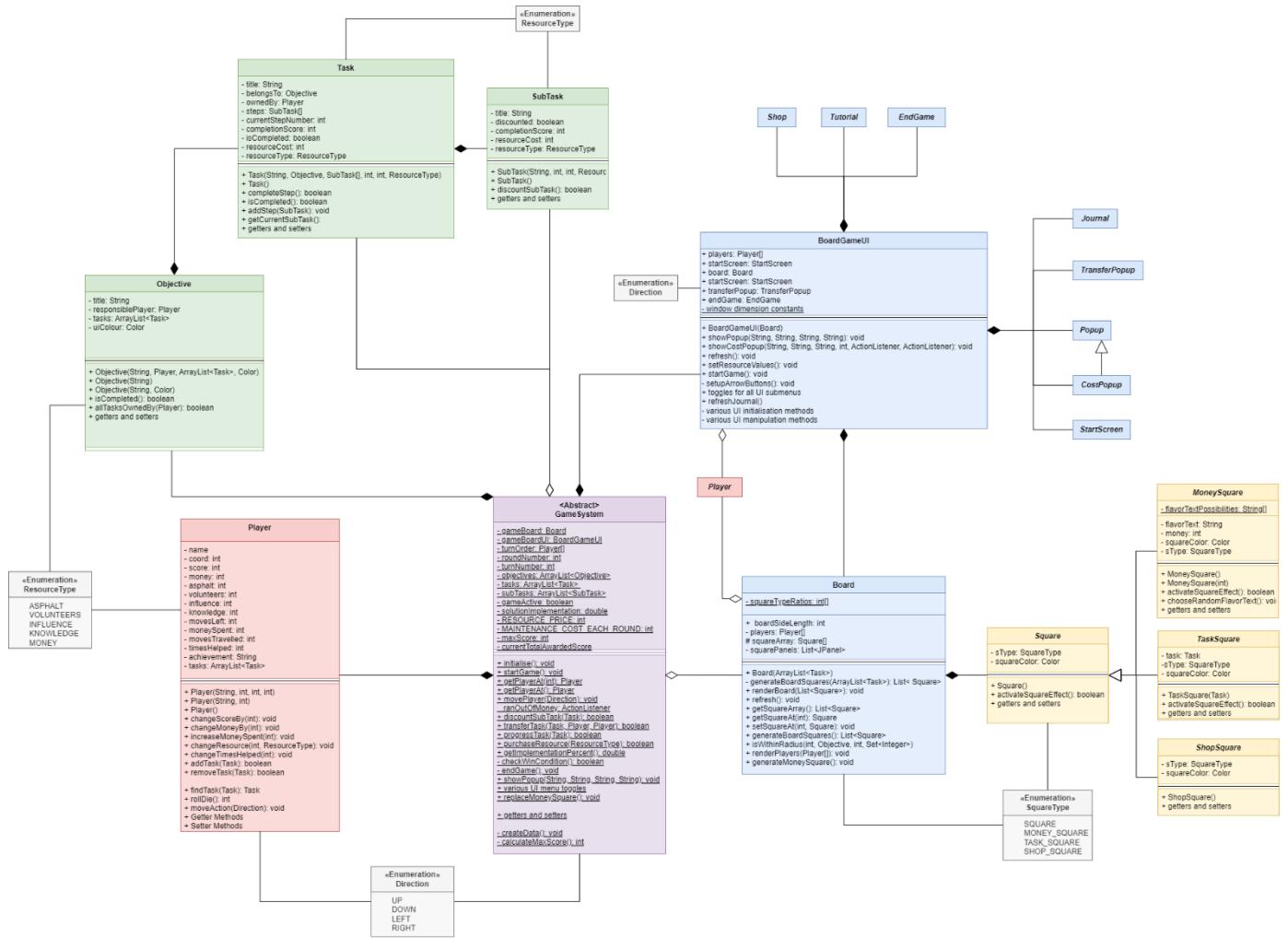


- The pixel art clearly represents a market stall, and the large title 'SHOP' at the bottom reminds the player of where they are.
- Each shelf stocks a different item and the price is listed on a hanging sign.

Again, there's a small X in the top right corner which highlights red when hovered over. This allows the player to close the shop menu and continue the game.

Class Diagram

[Principal: I.E.] [Support: P.R.]



The UML class diagram for the game is comprised of 5 main "parts": the GameSystem (purple), the Player class (red), the UI classes (blue), the objectives and classes directly related to completing the game (green) and finally the Square class and its subclasses that are used to make up the board (yellow). Enums are coloured in white.

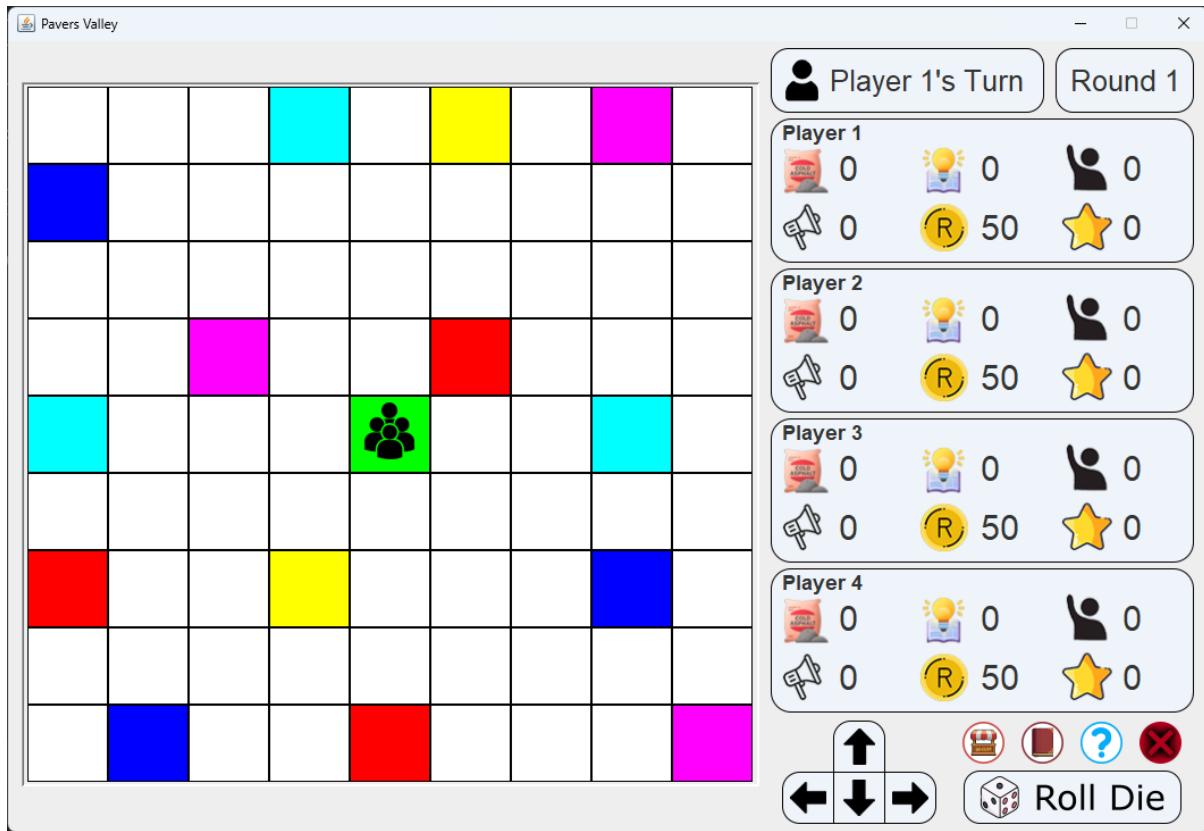
The abstract GameSystem class is the main central controller and manager of the game. It determines the game flow, keeps track of which player's turn it is and what round the game is currently on, as well as facilitating most of the communication between classes. GameSystem has associations with Objective, Task, and SubTask (which represents the 'steps' to complete each task) as the GameSystem is where each instance of these classes are stored.

The various classes associated with BoardGameUI are minor UI classes that serve a specific function of the game, such as showing the shop, the journal of completed tasks etc. They are not shown in detail for simplicity reasons.

The inheritance present in the square classes allows for polymorphism in the Board class, which makes the code simpler and the design easier to understand.

Final Game Layout

[Principal: N.W]



Our “Making it Better” game uses a 9x9 board populated with coloured squares. There is no square flow as players are free to move in the 4 cardinal directions.

Spawn/Shop Square (Green)	Task Squares (Red, Blue, Magenta and Cyan)	Money Square (Yellow)	Empty Square (White)
<p>Where all players start the game.</p> <p>You can view the shop from this square where you spend your money in exchange for needed resources.</p> <p>The opening tutorial will automatically play when the game starts.</p>	<p>Displays a popup when landed on, asking users if they would like to buy this task in exchange for resources, or if they want to offer it another player.</p> <p>Each colour represents a different objective.</p> <p>Blue – Secure Grant Red – Repair Potholes Magenta – Train Volunteers Cyan – Secure Longevity</p>	<p>Land on this task to receive a grant worth between 15 and 30 Rand, with a value of 20 having higher odds.</p> <p>This money can be spent in the shop in exchange for resources.</p>	<p>A sign of good work. When a task is completed, its square turns into an empty square.</p> <p>A money square can randomly spawn on empty squares.</p>

Implementation-Related Documentation

Acceptance Test Plan

[Principal: C.S.M.]

Basic Req\$	Use case	Description of Test	Test Initialisation	Test Inputs	Test Procedure	Expected Results	Passed?
1.1	UC01	Verify that when a when the names for all 4 players are input, the game begins with the correct resources for all.	Launch the game from the very beginning.	Enter names for all 4 Players that aren't default.	Start the Game. Enter names of 4 Players in their boxes. Press "Start Game" button and ensure resources are correctly given.	Once the game begins, each player's name will show up correctly, and all resources will be distributed correctly.	Yes
2.1	UC01	Verify that the tutorial pops up whenever the "Start Game" button is pressed.	Launch the game and press the "Start Game" button.	None.	Start the Game. Press "Start Game" button. Verify the tutorial pops up.	The tutorial will be displayed, where the stored tutorial text can be scrolled up and down to read all the text. [Re-test – see R1.]	No – The Tutorial is not scrollable.
3.1		Verify Players can stand on the Start Square to obtain resources.	Have the current Player standing on the Start Square, with enough money to buy each of the resources.	Click the "Shop" icon in the bottom right to open the shop. Click on each resource to buy them.	Have the current Player stand at the Start Square. Click the shop icon. Click on each resource to buy them.	Once each resource is clicked in the shop, they should be added to the current player's inventory, in exchange for money.	Yes
4.1		Verify that all Players can identify which subtasks are linked to each task and objective respectively.	Game Board is created with all Task Squares initialised.	Open journal.	Read tutorial which indicates what colour each Task Square corresponds to in relation to their Objective. Also open the journal to see what Subtasks are under each Task.	Each Task should be easily related to an objective by a player based on the colour of each Task Square. In the Journal, each Task and its corresponding Subtasks must file under the correct objectives.	Yes
5.1	UC05	Verify that all Players can roll a die in the pre-made order of ascending Player number, and that this movement moves that Player on the Board.	Start a game with at least two players.	Roll a die. Click on movement arrows. Click on end turn button. Repeat for all Players.	Roll a die, move the specified amount and end their turn. Then repeat this for all Players in game and ensure it loops back onto the beginning Player after the final player has gone.	Each Player should be able to roll one die and move the amount they roll, end their turn, and the turns should loop in the pre-made specified order of ascending Player number.	Yes
6.1		Verify that whenever a Player lands on a Task	Start game, initialise Players and roll a die so that the current player	Click on movement buttons to move	Click on movement buttons to bring the current Player to the designated Task Square. Ensure	Task Square will show a popup showing what Task it contains, and the ability to	Yes

		Square, the Player is prompted with an option.	can get to a Task Square.	into the necessary Square.	that once you land on this Square, a popup appears with relevant data and options for that current Player.	either accept the Task, or transfer it to someone else.	
6.2		Verify that whenever a Player lands on a Money Square, the Player is prompted with an option.	Start game, initialise Players and roll a die so that the current player can get to a Money Square.	Click on movement buttons to move into the necessary Square.	Click on movement buttons to bring the current Player to the designated Money Square. Ensure that once you land on this Square, a popup appears with relevant data and options for that current Player.	Money Square will show a popup showing a message on how you acquired X amount of money.	No – Incorrect popup shows. [Re-test – see R2.]
7.1		Verify that once you land on a Task Square, you can claim that Task as your own.	Start game, initialise Players and roll a die so that the current player can get to a Task Square.	Click on movement buttons to move into the Task Square. Click "Yes" button on popup to acquire the Task, spending the needed number of resources to obtain it. Then check in Journal to ensure you own that Task now.	Once the "Yes" button is pressed, if the Player has enough resources, then they shall acquire the Task. This will show in Journal. If they don't have enough resources, then they will not be able to acquire the Task.		Yes
8.1		Verify that once a Player lands on a Task Square owned by another Player, they'll have the option to dedicate some of their own resources to help the owner of the Task.	Start game, initialise Players, get a Player to claim a Task Square and roll a die so that the next player can get to the Task Square.	Click on movement buttons to move into the Task Square. Click "Yes" button on popup.	Click on movement buttons to move into the Task Square. Click "Yes" button on popup asking if you want to help complete the task. Then check if this has changed the cost of completing that Subtask within the Journal.	Once the "Yes" button is clicked, the price to complete that Subtask should be cut in half in the Journal. [Re-test – see R3.]	No – Resources are not actually dedicated towards Task, simply removed from Player when accepted.
9.1		Verify that all Players can dedicate resources towards completion of the Tasks that they own.	Have the current Player own a Task.	Click on the Journal button to open it. Click on the checkmark button three times.	Open the Journal and find the Task that has been claimed by the current Player, click on the corresponding "Progress Task" button shown using a green checkmark. Once this has been pressed three times, the Task should be marked as finished.	After the "Progress Task" button has been clicked, the Solution Implementation Meter should be incremented, and the turn should end. After the button has been clicked for the third time, it should mark the Task as completed.	Yes
10.1		Verify that all Players can identify who owns Tasks and the progress of each Task and Objective.	Players must own a random number of Tasks.	Click on the Journal button.	Click on the Journal button to show the Journal, from here, all Task and Objectives should be laid out, so all Players know what Tasks and	All Tasks and Objectives with an owner will display text showing the name of the Player that owns it.	Yes

					Objectives have been claimed by who.	Otherwise, if the Task or Objective is owned by nobody, then no text is shown.	
11.1		Verify that when all Tasks for an Objective are complete, that Objective will be marked as complete.	All Tasks but one must be completed for a particular Objective.	Click the "Progress Task" button on the final task in the Journal.	Finish the final Task for the Objective and then check the text for the Objective this Task falls under and see if its text has been crossed out.	The Objective text should be crossed through, showing that the Objective has been finished. The Solution Implementation Meter should also be incremented.	Yes
12.1	UC03	Verify that once all Objectives are completed, the game ends successfully, displaying a summary.	All Tasks in the system apart from one are completed.	Click the "Progress Task" button on the final task in the Journal.	Finish the final Task, end the current Players' turn, and observe the Post-Game Summary.	Once the final Task is completed, and the end turn button is pressed, the game will show an epilogue along with Player overall stats throughout the game.	No – Game is never finished even whenever all Objectives are completed. [Re-test – see R4.]
13.1	UC03	Verify that if any Player runs out of resources the game ends for all.	At least two Players have been input into the game. Where one Player is low on their money resource.	Click to buy a Resource in the shop.	Buy a resource so the current player gets to 0 money. Observe the Popup from the system.	Once a Player reaches 0 money, then the system should take the Players to the "Bad Ending", showing their final stats and an epilogue.	No – Game is never finished whenever Player runs out of all Resources. [Re-test – see R5.]
13.2	UC03	Verify that if any Player quits the game, the game ends for all.	At least two Players have been input into the game.	Click the "Quit" button.	Press the "Quit" button. Observe the Popup from the system.	Once the "Quit" button is pressed, then the system should take the Players to the "Bad Ending", showing their final stats and an epilogue.	Yes
14.1		Verify the endgame screen shows each Players' contributions and stats and an epilogue.	All Tasks in the system are completed.	Press the "End Turn" button.	Press the "End Turn" button. Observe the Popup from the System.	The endgame screen should show the stats of all Players and then an epilogue.	Yes
Value - Added Features		(No tests for va2 – va5 as they are already tested due to testing on past features).					
va1.1		Verify the GUI improves the navigation and readability of the game.	Launch the game.	Navigate throughout GUI.	Click throughout the game, testing every menu's implementation.	All menus should be shown correctly when needed.	No – due to failures from 2.1 and 6.2, not all menu's are being shown or being shown correctly. [Re-test – see R6.]

							being shown or being shown correctly. [Re-test – see R6.]
va6.1		Verify that Players can obtain money through "Money Squares".	Roll a die for the current Player.	Move onto a "Money Square".	Move the Player onto a Money Square. Navigate through the menu to receive a random amount of money.	The Player should be able to collect a random amount of Money between 2 pre-set thresholds.	No – Due to wrong menu being shown in core requirement 6.2, it is impossible to get to this menu to test this. [Re-test – see R7.]
Retests							
R1.1	UC01	Verify that the tutorial pops up whenever the "Start Game" button is pressed.	Launch the game and press the "Start Game" button.	None.	Start the Game. Press "Start Game" button. Verify the tutorial pops up.	The tutorial will be displayed, where the stored tutorial text can be scrolled up and down to read all the text.	Yes
R2.1		Verify that whenever a Player lands on a Money Square, the Player is prompted with an option.	Start game, initialise Players and roll a die so that the current player can get to a Money Square.	Click on movement buttons to move into the necessary Square.	Click on movement buttons to bring the current Player to the designated Money Square. Ensure that once you land on this Square, a popup appears with relevant data and options for that current Player.	Money Square will show a popup showing a message on how you acquired X amount of money.	Yes
R3.1		Verify that once a Player lands on a Task Square owned by another Player, they'll have the option to dedicate some of their own resources to help the owner of the Task.	Start game, initialise Players, get a Player to claim a Task Square and roll a die so that the next player can get to the Task Square.	Click on movement buttons to move into the Task Square. Click "Yes" button on popup asking if you want to help complete the task. Then check if this has changed the cost of completing that Subtask within the Journal.	Once the "Yes" button is clicked, the price to complete that Subtask should be cut in half in the Journal.		Yes
R4.1	UC03	Verify that once all Objectives are completed, the game ends successfully, displaying a summary.	All Tasks in the system apart from one are completed.	Click the "Progress Task" button on the final task in the Journal.	Finish the final Task, end the current Players' turn, and observe the Post-Game Summary.	Once the final Task is completed, and the end turn button is pressed, the game will show an epilogue along with Player overall stats throughout the game.	Yes
R5.1	UC03	Verify that if any Player runs out of resources the game ends for all.	At least two Players have been input into the game. Where one Player	Click to buy a Resource in the shop.	Buy a resource so the current player gets to 0 money. Observe the Popup from the system.	Once a Player reaches 0 money, then the system should take the Players to the	Yes

			<i>is low on their money resource.</i>			<i>"Bad Ending", showing their final stats and an epilogue.</i>	
R6.1		<i>Verify the GUI improves the navigation and readability of the game.</i>	<i>Launch the game.</i>	<i>Navigate throughout GUI.</i>	<i>Click throughout the game, testing every menu's implementation.</i>	<i>All menus should be shown correctly when needed.</i>	<i>Yes – Surveys (shown below) were sent out after fixes were made, resulting in a mostly positive review of navigation and readability of the system.</i>
R7.1		<i>Verify that Players can obtain money through "Money Squares".</i>	<i>Roll a die for the current Player.</i>	<i>Move onto a "Money Square".</i>	<i>Move the Player onto a Money Square. Navigate through the menu to receive a random amount of money.</i>	<i>The Player should be able to collect a random amount of Money between 2 pre-set thresholds.</i>	<i>No – Due to wrong menu being shown in core requirement 6.2, it is impossible to get to this menu to test this.</i>

Results of surveys

[Principal: C.S.M.]

[Survey for R6.1]

1. Using the GUI for the following purpose is easy to navigate.

[More Details](#)

■ Strongly Disagree ■ Disagree ■ Neutral ■ Agree ■ Strongly Agree

Adding Players



Reading the Tutorial



Popups are easy to read (e.g. Collecting a Task from a Task Square)



Using the Journal to work through Tasks



Using the Journal to understand what Objectives, Tasks, and Subtasks are related.



Using the Journal to see what Player owns each Task



Moving Across the Board to Task, Money and Shop Squares



Navigating through the End of Game Stats and Epilogue



100%

0%

100%

A total of 10 Surveys were completed for judging the GUI, most responses agree that the overall GUI is easy to read and navigate through.

Appendices

1. Team Minutes

January 13th, 2025

Minutes for CSC2058

Group 2 Date of this minute 13/01/2025 Location (Room No. and/or Teams): Teams

The following team members were present (in the same meeting room or on Teams) when these minutes were discussed:

Name (printed/typed)	In room (R); On teams (T).	Signature (agreed bitmap or initials)
Nathan Watkins	T	NW
Peter Robinson	T	PR
Kal Worthington	T	KW
Curtis McCartney	T	CSM
Isaac Edmonds	T	IE
Antons Bogdanovs	T	AB

Task Reporting

All:

- First meeting of the new semester so no tasks to report.

Actions Planned

Nathan, Kal, Curtis, Isaac, Antons:

- Fill in product backlog for meeting on Friday
- Talk to Ian regarding Semester 1 feedback and the project moving forward

Peter:

- Ensure the use cases fulfil the core requirements of the games.

Obstacles

All:

- O.K

Date of next minutes meeting: 17/01/2025

Location of next minutes meeting: (Room No. and/or Teams): CSB 01/020

January 17th , 2025

Minutes for CSC2058

**Group 2 Date of this minute 17/01/2025 Location (Room No. and/or Teams): CSB
01/020**

The following team members were present (in the same meeting room or on Teams) when these minutes were discussed:

Name (printed/typed)	In room (R); On teams (T).	Signature (agreed bitmap or initials)
Nathan Watkins	R	NW
Peter Robinson	R	PR
Kal Worthington	R	KW
Curtis McCartney	R	CSM
Isaac Edmonds	R	IE
Antons Bogdanovs	R	AB

Task Reporting

All:

- Made entries in the product backlog

Peter:

- Noted observations about the use case descriptions aligning with the core requirements (and any value-added features) as per Ian's feedback from semester 1

Actions Planned

Nathan:

- Start work on the Board class and player movement

Kal & Antons:

- Work on the Square class.

Curtis:

- Start work on the Player Class.

Curtis & Peter:

- Begin working on the BoardGameUI class using NetBeans.

Isaac:

- Create a document outlining the general flow and workings of the game

Isaac & Peter:

- Begin working on the GameSystem class.
- Implement barebones endRound and endTurn methods for GameSystem.

Obstacles

All:

- O.K

Date of next minutes meeting: 20/01/2025

Location of next minutes meeting: (Room No. and/or Teams): Teams

January 20th , 2025

Minutes for CSC2058

Group 2 Date of this minute 20/01/2025 Location (Room No. and/or Teams): Teams

The following team members were present (in the same meeting room or on Teams) when these minutes were discussed:

Name (printed/typed)	In room (R); On teams (T).	Signature (agreed bitmap or initials)
Nathan Watkins	T	NW
Peter Robinson	T	PR
Kal Worthington	T	KW
Curtis McCartney	T	CSM
Isaac Edmonds	T	IE
Antons Bogdanovs	T	AB

Task Reporting

Isaac:

- Create a document outlining the general flow and workings of the game

Nathan:

- Created the Board class and move player logic

Kal, Antons, Curtis, Peter:

- N/A

Actions Planned

Nathan & Curtis:

- Work on the BoardGameUI Class together on a meeting on 21st Jan.

Kal & Antons:

- Work on the Square class.

Curtis:

- Work on the Player Class

Isaac:

- Work on fleshing out the sprint backlog

Isaac & Peter:

- Begin working on the GameSystem class from tomorrow (21st January).
- Implement barebones endRound and endTurn methods for GameSystem.

Obstacles

All:

- O.K

Date of next minutes meeting: 24/01/2025

Location of next minutes meeting: (Room No. and/or Teams): CSB 01/020

January 24th , 2025

Minutes for CSC2058

Group 2 Date of this minute 24/01/2025 Location (Room No. and/or Teams): Teams

The following team members were present (in the same meeting room or on Teams) when these minutes were discussed:

Name (printed/typed)	In room (R); On teams (T).	Signature (agreed bitmap or initials)
Peter Robinson	T	PR
Curtis McCartney	T	CSM
Isaac Edmonds	T	IE
Antons Bogdanovs	T	AB

Task Reporting

Nathan & Kal:

- Absent from meeting likely due to Storm Eowyn

Curtis:

- Discussed updates to the GameBoardUI and GameBoard classes.

Isaac & Peter:

- Developed the GameSystem class further
- Created a static version of the GameSystem to demonstrate benefits of a static class

Antons:

- Discussed Square class and subclasses

Actions Planned (for next week)

Curtis & Nathan:

- Update the BoardGameUI and Player class to further align with the new GameSystem class.

Isaac & Peter:

- Further develop GameSystem class
- Start on the JobRole and Task class if enough time

Antons & Kal:

- Further develop the RoleSquare, TaskSquare and Square classes.

Obstacles

Curtis, Nathan, Kal, Antons:

- N/A

Isaac & Peter:

- Waiting for other classes to be completed to continue.

Everyone:

- Lasting damage from storm Eowyn.

Date of next minutes meeting: 27/01/2025

Location of next minutes meeting: (Room No. and/or Teams): Teams

January 27th , 2025

Minutes for CSC2058

Group 2 Date of this minute 27/01/2025 Location (Room No. and/or Teams): Teams

The following team members were present (in the same meeting room or on Teams) when these minutes were discussed:

Name (printed/typed)	In room (R); On teams (T).	Signature (agreed bitmap or initials)
Nathan Watkins	T	NW
Peter Robinson	T	PR
Kal Worthington	T	KW
Curtis McCartney	T	CSM
Isaac Edmonds	T	IE
Antons Bogdanovs	T	AB

Task Reporting

All (except Isaac):

- Nothing

Isaac:

- Created simple class definitions for JobRole and Task in the UML class diagram

Actions Planned

Peter & Isaac:

- Develop the GameSystem class further
- Incorporate JobRole and Task classes into GameSystem

Isaac:

- Develop rudimentary JobRole and Task classes

Curtis & Nathan:

- Pair-program more of the UI together.

Kal & Antons:

- Continue working on Square class

Obstacles

All:

- O.K.

Date of next minutes meeting: 31/01/2025

Location of next minutes meeting: (Room No. and/or Teams): CSB 01/020

January 31st , 2025

Minutes for CSC2058

**Group 2 Date of this minute 31/01/2025 Location (Room No. and/or Teams): CSB
01/020**

The following team members were present (in the same meeting room or on Teams) when these minutes were discussed:

Name (printed/typed)	In room (R); On teams (T).	Signature (agreed bitmap or initials)
Nathan Watkins	R	NW
Peter Robinson	R	PR
Kal Worthington	R	KW
Curtis McCartney	R	CSM
Isaac Edmonds	R	IE
Antons Bogdanovs	R	AB

Task Reporting

Nathan:

- Made the startScreen class
- Made popup class with Curtis

Kal & Antons:

- Continued work on Square and related classes

Curtis:

- Made popup class with Nathan.

Isaac & Peter:

- Continued implementation of GameSystem class.

Actions Planned

Everyone:

- None planned over weekend

Obstacles

All:

- O.K.

Date of next minutes meeting: 03/02/2025

Location of next minutes meeting: (Room No. and/or Teams): Teams

February 3rd , 2025

Minutes for CSC2058

Group 2 Date of this minute 03/02/2025 Location (Room No. and/or Teams): Teams

The following team members were present (in the same meeting room or on Teams) when these minutes were discussed:

Name (printed/typed)	In room (R); On teams (T).	Signature (agreed bitmap or initials)
Nathan Watkins	T	NW
Peter Robinson	T	PR
Kal Worthington	T	KW
Curtis McCartney	T	CSM
Isaac Edmonds	T	IE
Antons Bogdanovs	T	AB

Task Reporting

Nathan:

- Made the board use the square class instead of strings.
- Added icons for 1-4 players.

Everyone except Nathan:

- No actions were planned last meeting.

Actions Planned

Nathan:

- Start working on the Journal class

Peter:

- Start implementing end game method in GameSystem, handling different scenarios including when a player quits.

Kal & Antons:

- Work on Money Squares
- Think of ideas for tasks and resources
- Implement money into Player class

Curtis:

- Work on landing on squares activating square effects.
- Work on assigning tasks to Players.

Isaac:

- Update UML class diagram for the new system

Obstacles

All:

- O.K

Date of next minutes meeting: 07/02/2025

Location of next minutes meeting: (Room No. and/or Teams): CSB 01/020

February 7th , 2025

Minutes for CSC2058

**Group 2 Date of this minute 07/02/2025 Location (Room No. and/or Teams): CSB
01/020**

The following team members were present (in the same meeting room or on Teams) when these minutes were discussed:

Name (printed/typed)	In room (R); On teams (T).	Signature (agreed bitmap or initials)
Nathan Watkins	R	NW
Peter Robinson	R	PR
Kal Worthington	R	KW
Curtis McCartney	R	CSM
Isaac Edmonds	R	IE
Antons Bogdanovs	R	AB

Task Reporting

Nathan:

- Created the Journal class

Isaac:

- Updated UML diagram to better reflect the changes made to the function of the game

Peter:

- Worked on GameSystem, such as adding comments and removing redundant features.
- Implemented barebones functionality of the endGame() method. Messages are printed via the console and .txt files are used to read the future summary of events using File and Scanner.

Curtis:

- Worked on the implementation of taskSquares, assigning tasks, and how to activate a square.
- Updated player, Task, subtask, objective to align with new gameplay requirements.

Kal and Antons:

- Created Money Square Class and modified player class

Actions Planned

Nathan:

- Start work on the Shop class, including a graphic

Peter:

- Continue implementation on endGame, like displaying player scores and achievements.

Others:

- Nothing planned

Obstacles

All:

- O.K

Date of next minutes meeting: 17/02/2025

Location of next minutes meeting: (Room No. and/or Teams): Teams

February 17th , 2025

Minutes for CSC2058

Group 2 Date of this minute 17/02/2025 Location (Room No. and/or Teams): Teams

The following team members were present (in the same meeting room or on Teams) when these minutes were discussed:

Name (printed/typed)	In room (R); On teams (T).	Signature (agreed bitmap or initials)
Nathan Watkins	T	NW
Peter Robinson	T	PR
Kal Worthington	T	KW
Curtis McCartney	T	CSM
Isaac Edmonds	T	IE
Antons Bogdanovs	T	AB

Task Reporting

All except Kal & Nathan:

- Nothing (Reading week)

Nathan:

- Created the Shop class and a shop graphic

Kal

- Removed references to primaryOccupier from square

Actions Planned

Nathan:

- Implement task square onto the board.
- Allow players to close the shop.

Peter:

- Handle end game statistics such as player achievements and total points spent.
- If done, create write ups for the A1 poster.

Kal & Antons

- Work on the A1 poster

Curtis:

- Create GUI for giving Tasks and helping on tasks.

Isaac:

- Create methods in GameSystem for better functionality for UI buttons
- completeSubtask() method in gamesystem that subtracts resources from the player in control of that task and marks the subtask as completed

Obstacles

Isaac:

- Cannot work with resources until the updated Task, Objective, and SubTask classes are merged into the master branch

All:

- O.K

Date of next minutes meeting: 21/02/2025

Location of next minutes meeting: (Room No. and/or Teams): CSB 01/020

February 21st , 2025

Minutes for CSC2058

Group 2 Date of this minute 21/02/2025 Location (Room No. and/or Teams): Teams

The following team members were present (in the same meeting room or on Teams) when these minutes were discussed:

Name (printed/typed)	In room (R); On teams (T).	Signature (agreed bitmap or initials)
Nathan Watkins	R	NW
Peter Robinson	R	PR
Kal Worthington	R	KW
Curtis McCartney	R	CSM
Isaac Edmonds	R	IE
Antons Bogdanovs	R	AB

Task Reporting

Peter:

- Implemented end game statistics such as player achievements and total points spent.

Nathan:

- Created file input logic for tasks and implemented to journal,
- Made each task have its own popup

Kal and Antons:

- Created A1 poster guidance

Isaac:

- Nothing, due to other projects

Curtis:

- Created UI for receiving tasks and landing on already claimed TaskSquare.
- Merged back into master, making master now up to date with new Objectives/Tasks/SubTasks idea.

Actions Planned

Nathan:

- Design UI elements for transferring a task, tutorial and end game.

Peter:

- Write the description for the tutorial in a .txt file, detailing:
 - Background setting and problem
 - How the player rolls the die and moves
 - Objectives, tasks, and subtasks
 - How you interact with shop UI/journal UI
 - Transferring tasks to other players
 - Winning/Losing criteria

Kal & Antons

- Continue working on A1 poster

Curtis:

- Change Colour of each TaskSquare depending on the Objective it links to.
- Create implementation of UI for the completion of SubTasks/Tasks in Journal.

Isaac:

- Add functional progressTask() method (renamed from completeSubtask()) in the GameSystem and implement resource consumption functionality

Obstacles

Isaac & Curtis:

- Other modules' projects getting in the way

All:

- O.K

Date of next minutes meeting: 24/02/2025

Location of next minutes meeting: (Room No. and/or Teams): Teams

February 24th , 2025

Minutes for CSC2058

Group 2 Date of this minute 24/02/2025 Location (Room No. and/or Teams): Teams

The following team members were present (in the same meeting room or on Teams) when these minutes were discussed:

Name (printed/typed)	In room (R); On teams (T).	Signature (agreed bitmap or initials)
Nathan Watkins	T	NW
Peter Robinson	T	PR
Kal Worthington	T	KW
Curtis McCartney	T	CSM
Isaac Edmonds	T	IE
Antons Bogdanovs	T	AB

Task Reporting

Nathan:

- Added transfer and complete task buttons and owned by label to the Journal class.

Peter

- Wrote description for the tutorial. This is stored in a .txt file for future file reading.

All but Nathan & Peter:

- Nothing

Actions Planned

Nathan:

- Expand main screen to show more resource stats for each player.
- Create end game and tutorial screens/popups.

Peter:

- Write test cases for the GameSystem class.
- Implement a GameSystem method called playTutorial(), which will handle file reading for the tutorial .txt file (backend).

Kal and Anton:

- Continue work on the A1 poster

Curtis:

- Write test cases for the Player Class.
- Create backend and link to UI for transferring of Tasks.
- Change colour of each task depending on what Objective it belongs to.

Isaac:

- Implement functionality for players completing a task and subtask
- Unit test the new Objective, Task, and Subtask classes
- Implement global Solution Implementation percentage in GameSystem

Obstacles

All but Isaac & Peter:

- O.K

Peter:

- Unsure whether to implement unit or system testing for the game system; will research pros and cons of each before making a decision.

Isaac:

- Lost laptop charger over the weekend, cannot work on laptop until Wednesday when it can be picked up again.

Date of next minutes meeting: 28/02/2024

Location of next minutes meeting: (Room No. and/or Teams): CSB 01/020

February 28th , 2025

Minutes for CSC2058

**Group 2 Date of this minute 28/02/2025 Location (Room No. and/or Teams): CSB
01/020**

The following team members were present (in the same meeting room or on Teams) when these minutes were discussed:

Name (printed/typed)	In room (R); On teams (T).	Signature (agreed bitmap or initials)
Nathan Watkins	R	NW
Peter Robinson	R	PR
Kal Worthington	R	KW
Curtis McCartney	R	CSM
Isaac Edmonds	R	IE
Antons Bogdanovs	R	AB

Task Reporting

Nathan:

- Created the tutorial popup.
- Created the end game screen with statistics and an epilogue.

Peter:

- Created JUnit test cases for the GameSystem class.
- Implemented a GameSystem method called playTutorial(), which handles file reading for the tutorial .txt file (backend). Headings are prefixed with a # symbol.
- Transferred playTutorial() to new Tutorial class.

Kal & Antons:

- Wrote text for the A1 poster

Curtis:

- Board now uses only 12 TaskSquares that now relate to the Tasks held in System.
- Created the backend for Transferring Tasks between Players.

Isaac:

- Implemented progressTask() in GameSystem
- Created test cases for SubTask

Actions Planned

Nathan:

- Decrease the size of the board and expand ui to show more resources

Peter:

- Work on the demo video schedule.

Curtis:

- Work on the Acceptance Test Plan.
- Work on Test Cases for Board.
- Objectives being complete need to call the endGame() method.

Isaac:

- Finish up unit test cases for Objective, Task, Subtask
- Implement global activation percentage in GameSystem
- Finish up purchaseResources

Kal & Antons:

- Finish work on the poster

Obstacles

All:

- O.K

Date of next minutes meeting: 03/03/25

Location of next minutes meeting: (Room No. and/or Teams): Teams

March 3rd, 2025

Minutes for CSC2058

Group 2 Date of this minute 03/03/2025 Location (Room No. and/or Teams): Teams

The following team members were present (in the same meeting room or on Teams) when these minutes were discussed:

Name (printed/typed)	In room (R); On teams (T).	Signature (agreed bitmap or initials)
Nathan Watkins	T	NW
Peter Robinson	T	PR
Kal Worthington	T	KW
Curtis McCartney	T	CSM
Isaac Edmonds	T	IE
Antons Bogdanovs	T	AB

Task Reporting

Nathan:

- Started work on expanding the UI to show more resources.

Peter:

- Game Demo Schedule:** Identified core requirements and value-added features.

Kal, Curtis & Antons:

- No work to report.

Isaac:

- Completed progressTask(), purchaseResources(), and getImplementationPercent() methods in GameSystem
- Redid how resources are handled by adding a ResourceType enum
- Wrote JUnit test cases for Objective, Task, and SubTask

Actions Planned

Nathan:

- Finish the UI expansion to show more resources.
- Add more stats to the end game popup

Peter:

- Merge Isaac's changes into the GameSystem branch and verify test cases.
- Write script for the game demo video.

Curtis:

- Work on the Acceptance Test Plan.
- Work on Test Cases for Board.
- Objectives being complete need to call the endGame() method.

Isaac:

- Update UML diagram to current version of system
- Help other devs where needed

Antons and Kal:

- Continue Work on the A1 Poster.

Obstacles

Isaac:

- Looming project from another course module will displace productivity.

All:

- O.K

Date of next minutes meeting: 07/03/2025

Location of next minutes meeting: (Room No. and/or Teams): CSB 01/020

March 7th, 2025

Minutes for CSC2058

Group 2 Date of this minute 07/03/2025 Location (Room No. and/or Teams): CSB 01/020

The following team members were present (in the same meeting room or on Teams) when these minutes were discussed:

Name (printed/typed)	In room (R); On teams (T).	Signature (agreed bitmap or initials)
Nathan Watkins	R	NW
Peter Robinson	R	PR
Kal Worthington	R	KW
Curtis McCartney	R	CSM
Isaac Edmonds	R	IE
Antons Bogdanovs	R	AB

Task Reporting

Nathan:

- Added the solution implementation statistic to the Journal.
- Made visual changes to the Journal.

Peter:

- Merged Isaac's changes into the GameSystem branch. Wrote new test cases to verify changes.
- Devised script for the game demo video.

Kal & Antons:

- Poster

Curtis:

- Created most of the Acceptance Test Plan.
- Worked on Test Cases for Board.
- Objectives being complete now call the endGame() method.

Isaac:

- Nothing

Actions Planned

Nathan:

- Create a popup for purchasing tasks and resources.
- Create a quit game button.

Peter:

- **End game:** Integrate logic to trigger popups and display achievement titles.
- **GameSystem tests:** Fix testCheckWinConditionAllComplete to utilise proper list of objectives rather than a dummy list.

Curtis:

- Work more on the Acceptance Test Plan.
- Work more on Test Cases for Board.
- Make resource economy in game, with popups using resources.

Isaac:

- None planned

Antons and Kal:

- Polish and finish poster

Obstacles

All:

- O.K

Date of next minutes meeting: 10/03/25

Location of next minutes meeting: (Room No. and/or Teams): Teams

March 10th, 2025

Minutes for CSC2058

Group 2 Date of this minute 10/03/2025 Location (Room No. and/or Teams): Teams

The following team members were present (in the same meeting room or on Teams) when these minutes were discussed:

Name (printed/typed)	In room (R); On teams (T).	Signature (agreed bitmap or initials)
Nathan Watkins	T	NW
Peter Robinson	T	PR
Kal Worthington	T	KW
Curtis McCartney	T	CSM
Isaac Edmonds	T	IE
Antons Bogdanovs	T	AB

Task Reporting

Nathan:

- Made the quit game button.
- Made a popup for buying tasks and resources.

Peter:

- **End game:** Created logic to trigger popups and display achievement titles.
- **GameSystem tests:** Refined testCheckWinCondition (complete and incomplete) to utilise proper list of objectives rather than a dummy list.
- **PDF Report:** Planned outline for UI section.

Kal & Antons:

- Poster

Curtis:

- Created more of the Acceptance Test Plan.

Isaac:

- Nothing

Actions Planned

Nathan:

- Complete the final game layout for the PDF Report.

Peter:

- **Endgame:** If finished PDF report task and free, add extra achievements to ensure each player gets one, unique achievement.

Curtis:

- Finish Acceptance Test Plan, including creation and getting results of surveys.
- Create ability to use resources on Popups.

Isaac:

- Update UML diagram

Antons and Kal:

- Polish and finish poster

Kal & Peter:

- **PDF Report:** Write up UI design elements chosen, including the main window, start screen, shop, endgame and more.

Obstacles

All:

- Work for other modules diverting attention.

Date of next minutes meeting: 14/03/25

Location of next minutes meeting: (Room No. and/or Teams): CSB 01/020

March 14th, 2025

Minutes for CSC2058

**Group 2 Date of this minute 14/03/2025 Location (Room No. and/or Teams): CSB
01/020**

The following team members were present (in the same meeting room or on Teams) when these minutes were discussed:

Name (printed/typed)	In room (R); On teams (T).	Signature (agreed bitmap or initials)
Nathan Watkins	R	NW
Peter Robinson	R	PR
Kal Worthington	R	KW
Curtis McCartney	R	CSM
Isaac Edmonds	R	IE
Antons Bogdanovs	R	AB

Task Reporting

Nathan:

- Made the Final Game Layout for the PDF report.

Peter & Kal:

- PDF Report:** Devised UI design, including the main window, start screen, shop, endgame and more.

Kal & Antons:

- References for Poster

Curtis:

- Created more of the Acceptance Test Plan.
- Created ability to use resources on Popups.

Isaac:

- UML class diagram updated to newer version

Actions Planned

All:

- Record segments for poster video.

Peter:

- Finish endgame achievements logic.

Curtis:

- Edit together Poster Video.
- Finish off using resources on Popups.
- Fix multiple bugs in game.

Isaac:

- Finalise UML class diagram

Obstacles

All:

- Work for other modules diverting attention

Date of next minutes meeting: 18/03/25

Location of next minutes meeting: (Room No. and/or Teams): Teams

March 18th, 2025

Minutes for CSC2058

Group 2 Date of this minute 18/03/2025 Location (Room No. and/or Teams): Teams

The following team members were present (in the same meeting room or on Teams) when these minutes were discussed:

Name (printed/typed)	In room (R); On teams (T).	Signature (agreed bitmap or initials)
Nathan Watkins	T	NW
Peter Robinson	T	PR
Curtis McCartney	T	CSM
Isaac Edmonds	T	IE

Task Reporting

All:

- Recorded the Poster Video

Nathan:

- Cleaned up and formatted the game code

Peter:

- Endgame achievement logic implemented – players now get one achievement (including a default participation award) to ensure uniqueness.
- Updated UI Design to reflect new changes.
- Updated tutorial description to clarify task square colours and what resources are used for each objective. Added losing criteria for when a player runs out of resources (i.e. maintenance costs).
- **PDF Report**
 - Transferred UI Design PowerPoint slides to PDF report
 - Added completed team minutes to appendix 1
 - Refined UML diagram section
 - Added credits for poster, and poster and game demo videos
 - Devised suggestions for appendix 4 (Comments on the development processes selected or rejected)

Curtis:

- Edited together the Poster Video
- Finished off using resources on needed Popups.
- Fixed bugs in game.

Isaac:

- Finalised UML diagram

Actions Planned

All:

- Record the Game Demo Video

Nathan:

- Complete GitLab and JUnit Tests evidence appendices

Peter:

- Complete appendix 8 (Security and Assurance Considerations)
- If free, update existing and add new comments in GameSystem

Curtis:

- Complete Video Script.
- Edit together Game Demo Video

Isaac:

- Write up the Development functions accepted or rejected appendix on the PDF report

Obstacles

All:

- O.K

2. Credits: Poster

Principal: K.W., A.B.

3. Credits: Poster Video

Principal: C.S.M.

Support: K.W., A.B., P.R., I.E, N.W.

4. Credits: Final Game Demo Video

Principal: C.S.M., P.R.

Support: K.W., A.B., I.E, N.W.

5. Comments on the development processes selected or rejected

NetBeans was originally used in semester one for GUI development, but in semester two we decided to move forward with writing the code ourselves for more creative control and more concise UI classes.

Solar panels were originally considered as our real-world solution to the rolling blackouts across Johannesburg including Maker's Valley, but after consideration it was determined that Maker's Valley already has similar solutions in place.

Hydroponics Farming in urban environments was also considered as a real-world solution to working on food shortages with a restricted water supply but this proposal was also rejected due to the costs and scale of the operation necessary.

For our solution we decided to push for cold asphalt mix for cheaper and easier to source repair materials. Hot asphalt is longer lasting but requires heavy machinery and is much more expensive.

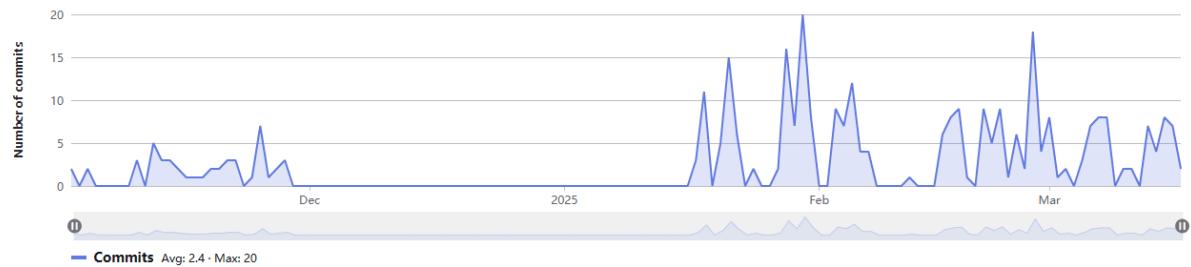
We initially agreed to having 2 phases in our game which would serve as different overall goals for players to complete. However, after a discussion with advisors, we decided to leave this idea for a simpler, one phase game.

We also proposed having role squares to hand out roles to people rather than objectives being completed, but this was determined to not meet the core requirements for the game.

6. GitLab Evidence

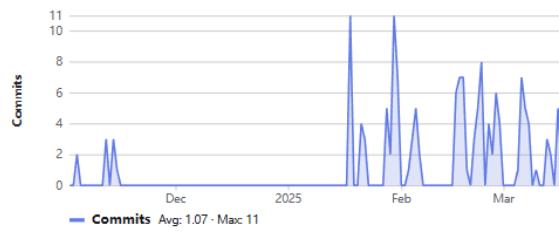
Commits to master

Excluding merge commits. Limited to 6,000 commits.



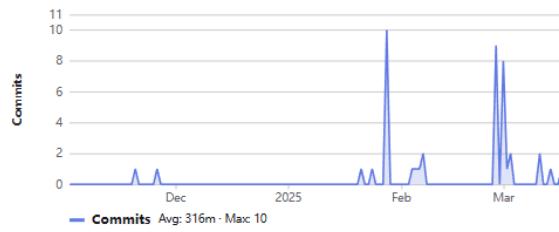
40401241

145 commits (nwatkins02@qub.ac.uk)



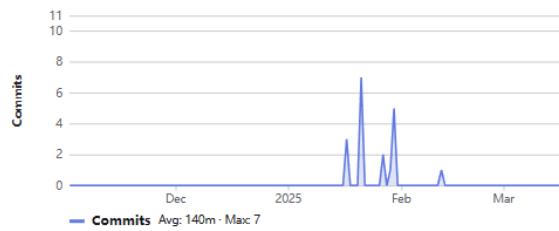
40405595

43 commits (ledmonds01@qub.ac.uk)



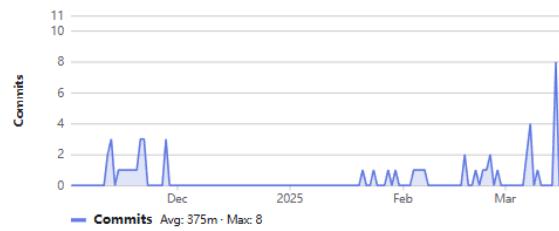
40400985

19 commits (kworthington01@qub.ac.uk)



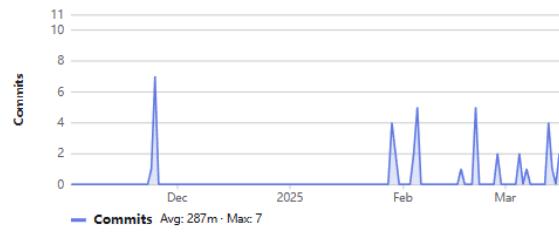
40400821

51 commits (probinson21@qub.ac.uk)



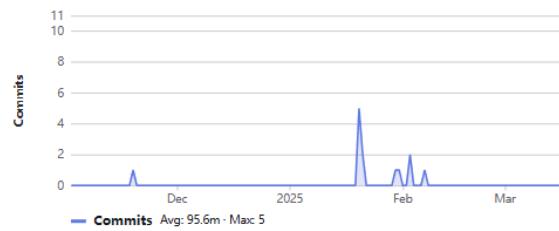
40405597

39 commits (cmccartney29@qub.ac.uk)



40407018

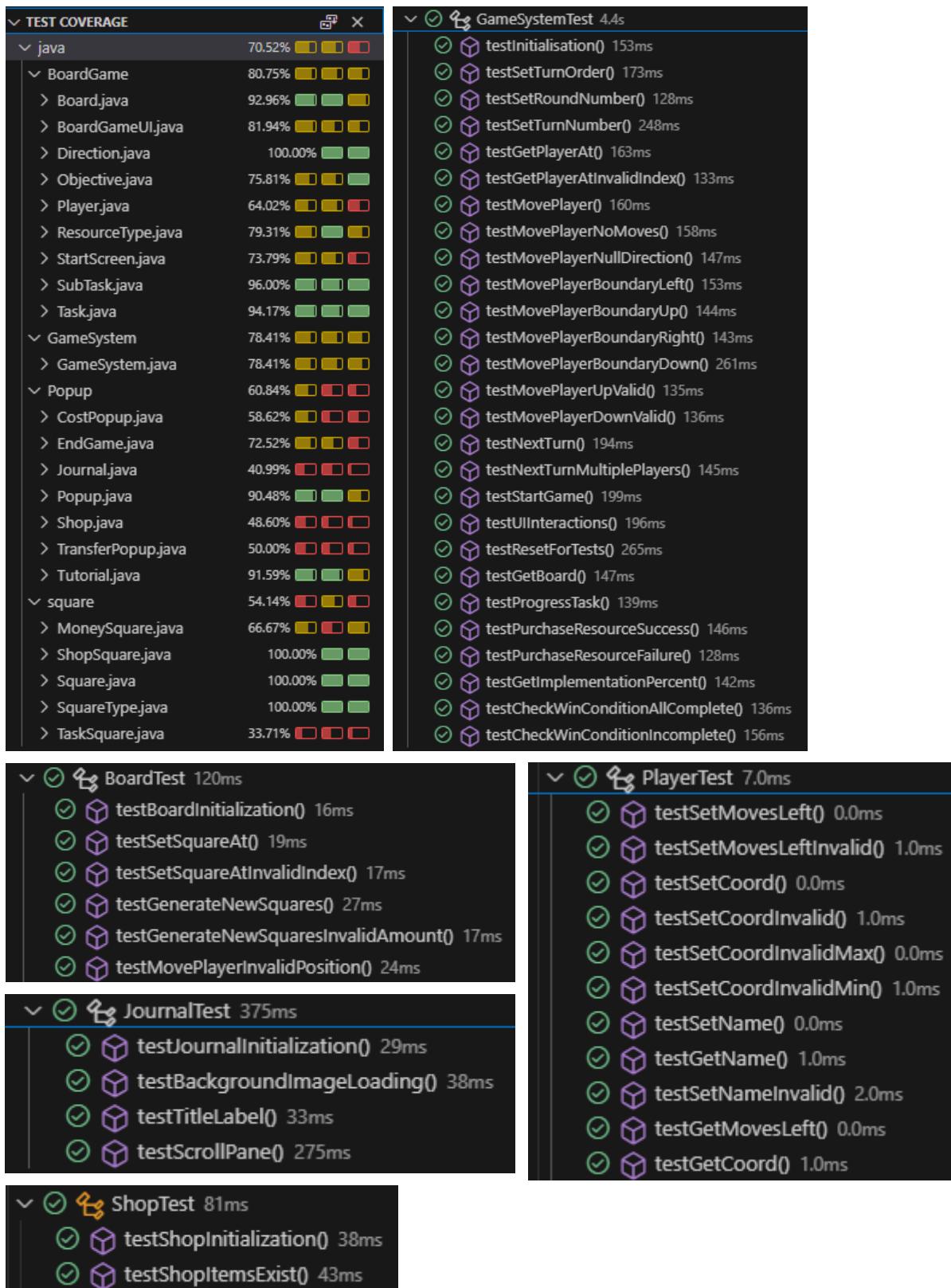
13 commits (abogdanovs01@qub.ac.uk)



 40405595 @40405595	19 hours ago
- Pushed to branch <code>master</code>	
5b411de0 · Replace UML_Class_Diagram.drawio	
 40401241 @40401241	1 day ago
- Pushed to branch <code>master</code>	
a1d26b5e · Fixed start screen graphic issue	
... and 5 more commits. Compare 9041dd16...a1d26b5e	
 40405597 @40405597	1 day ago
- Pushed to branch <code>master</code>	
9041dd16 · Got rid of unnecessary debug code.	
 40405597 @40405597	1 day ago
- Pushed to branch <code>master</code>	
0df5900d · Merge branch 'master' of https://gitlab.eeecs.qub.ac.uk/CSC2058-242...	
... and 1 more commit. Compare 5d9eca95...0df5900d	
 40400821 @40400821	2 days ago
- Deleted branch <code>game-system</code>	
 40400821 @40400821	2 days ago
- Pushed to branch <code>master</code>	
5d9eca95 · Merge branch 'game-system' into 'master'	
... and 10 more commits. Compare 67dce662...5d9eca95	
 40400821 @40400821	2 days ago
- Accepted merge request !18 "Game system"	
 40400821 @40400821	2 days ago
- Pushed to branch <code>master</code>	
67dce662 · Update tutorial.txt	
 40400821 @40400821	2 days ago
- Opened merge request !18 "Game system"	
 40400821 @40400821	2 days ago
- Pushed to branch <code>game-system</code>	
13afe94 · Refactor variable names to use uppercase constants for resource rew...	
... and 1 more commit. Compare a6e484bc...13afe94	
 40400821 @40400821	1 week ago
- Pushed to branch <code>master</code>	
08b3347f · Removed symbols causing rendering issues	
 40400821 @40400821	1 week ago
- Pushed to branch <code>game-system</code>	
0fd6b561 · Merge branch 'master' into game-system	
... and 2 more commits. Compare 94ce65b6...0fd6b561	
 40401241 @40401241	1 week ago
- Pushed to branch <code>UI-changes</code>	
259d9494 · Players can now only transfer tasks on their turn	
... and 1 more commit. Compare ed7393fd...259d9494	
 40401241 @40401241	1 week ago
- Pushed to branch <code>UI-changes</code>	
ed7393fd · Only shows objectives and tasks that are assigned	
 40401241 @40401241	1 week ago
- Pushed to branch <code>UI-changes</code>	
3b60b8a2 · Removed test code	
 40401241 @40401241	1 week ago
- Pushed to branch <code>UI-changes</code>	
02cb9715 · Fixed issue with popup buttons appearing white when clicked	
... and 1 more commit. Compare f7825dac...02cb9715	
 40401241 @40401241	1 week ago
- Pushed to branch <code>UI-changes</code>	
f7825dac · Fixed setting the visibility of the shop button	
... and 1 more commit. Compare 1d1c19cd...f7825dac	
 40401241 @40401241	1 week ago
- Pushed to branch <code>UI-changes</code>	
1d1c19cd · Buying resources now reflects on the ui	
 40401241 @40401241	1 week ago
- Pushed to branch <code>UI-changes</code>	
66385c06 · Refactored BoardGameUI and added solutionImplementation to the Journal	

	40405597 @40405597	2 weeks ago
↳ Pushed to branch board-rework		
cb4bba74 · Updated transferring popup to not allow transfer to yourself when d...		
	40400821 @40400821	2 weeks ago
↳ Pushed to branch game-system		
ab1b81a9 · New test cases and methods associated with testing		
	40401241 @40401241	2 weeks ago
↳ Pushed to branch Tutorial		
3f09fd5b · Added javadocs to tutorial		
	40401241 @40401241	2 weeks ago
↳ Pushed to branch Tutorial		
4d06ac7e · Added the close button and title to the tutorial		
	40401241 @40401241	2 weeks ago
↳ Pushed new branch Tutorial		
	40405597 @40405597	2 weeks ago
↳ Pushed to branch board-rework		
77116108 · Board Rework with 12 TaskSquares, Tasks in each Square, and new col...		
	40405597 @40405597	2 weeks ago
↳ Pushed new branch board-rework		
	40405595 @40405595	2 weeks ago
↳ Pushed to branch ResourceConsumption		
32718636 · Collapsed 4 resources down into one resource ... and 1 more commit. Compare f616dd8a... 32718636		
	40400821 @40400821	2 weeks ago
↳ Pushed to branch <code>master</code>		
731364fc · Merge branch 'game-system' ... and 13 more commits. Compare f5d956e6... 731364fc		
	40401241 @40401241	2 weeks ago
↳ Deleted branch Popups		
	40400985 @40400985	1 month ago
↳ Pushed to branch Money		
c22d447e · Merge branch 'master' into 'Money' ... and 11 more commits. Compare 797a4ca6... c22d447e		
	40400985 @40400985	1 month ago
↳ Opened merge request !9 "Money"		
	40400985 @40400985	1 month ago
↳ Pushed to branch Money		
797a4ca6 · removed references to primary occupier and added javadoc comments t...		
	40405595 @40405595	1 month ago
↳ Pushed to branch <code>master</code>		
a6e4523b · Added additional methods to the GameSystem		
	40400821 @40400821	1 month ago
↳ Pushed to branch game-system		
12968c9c · Small update to endGame()		
	40405595 @40405595	1 month ago
↳ Pushed to branch <code>master</code>		
a7e1059e · Renamed some methods in Objective/SubTask and created the SquareType...		
	40407018 @40407018	1 month ago
↳ Pushed to branch Money		
bc1edd46 · syntax		
	40400821 @40400821	1 month ago
↳ Pushed to branch game-system		
b2f1d1e4 · Updated endGame() method ... and 6 more commits. Compare 97483c0d... b2f1d1e4		
	40401241 @40401241	1 month ago
↳ Pushed to branch <code>master</code>		
c153b8b9 · Changed version to 17 in pom.xml		
	40401241 @40401241	1 month ago
↳ Pushed to branch <code>master</code>		
c9b5d819 · Made project compatible with Java 17		
	40405595 @40405595	1 month ago
↳ Pushed to branch <code>master</code>		
d9b5fb59 · Created Popup, StartScreen, and Journal. Also added Direction enum		

7. JUnit Test Runs



8. Security and Assurance Considerations

Inadequate Repairs

To prevent inadequate pothole repairs using unauthorised materials (e.g. stones or bricks), a moderator trained in pothole repair will assess photo submissions from volunteers. This moderator, potentially trained through our initiative, follows a standardised verification process to determine whether the repair meets required standards. Security measures can include:

- Geo-tagging and timestamping photo submissions to confirm authenticity.
- AI-assisted validation, flagging inconsistencies such as duplicated images.
- A rating system allowing the community to provide feedback on completed repairs.

If a repair meets standards, it is approved, allowing volunteers to receive rewards. If further work is needed, responsible volunteers will be notified for correction. To prevent submission fraud, safeguards like sequential progress tracking (e.g. requiring before, during, and after photos) can be enforced.

Stock Management and Theft Prevention

A stock management system will track inventory usage, preventing theft and unauthorised use of essential repair equipment. Volunteers collecting items from the Maker's Valley MVP studio must log their equipment usage, including:

- Tool name and serial number (e.g. asphalt saws, bags of cold asphalt).
- User identity and timestamp (ensuring accountability).
- Repair job location and expected return time (to track allocation and prevent hoarding).

Admins can monitor stock levels in real-time, including:

- Who last used an item and for what purpose.
- When equipment was taken and returned.
- Maintenance status, ensuring tools remain functional.

Security and Access Control Measures

To prevent unauthorised access and data tampering, the system will implement:

- Role-based access control (RBAC) to restrict admin features to authorised personnel.
- Multi-factor authentication (MFA) for inventory management.
- Alerts for unusual activity, such as excessive withdrawals by a single user.

9. Third-party software toolkits used

- ChatGPT/Copilot
- GitKraken/GitLens
- GitHub Desktop
- Visual Studio Code
- IntelliJ IDEA
- Eclipse IDE
- NetBeans IDE