

# **FIT3077 / S1 2025**

## **Sprint 1 Submission**

### **Group Members:**

Suhani / 33840075

Gracia / 33840008

Jocelyn / 33732973

## Table of Contents

<b>Part 1: Team Information and Technology Prototypes.....</b>	<b>2</b>
Team Membership.....	2
Team Schedule.....	3
Technology Stack and Justification.....	3
<b>Part 2: User Stories.....</b>	<b>5</b>
<b>Part 3: Domain Model.....</b>	<b>11</b>
<b>Part 4: Basic UI Design.....</b>	<b>16</b>
<b>Contributor Analytics.....</b>	<b>20</b>

# Part 1: Team Information and Technology Prototypes



Team Name: **ByteSized**

Moodle Team Name: **CL\_Tuesday04pm\_Team050**

## Team Membership

A1 Team Leader → Suhani Gadgil

Name	Email
Suhani	sgad0010@student.monash.edu
Gracia	gyap0005@student.monash.edu
Jocelyn	jtan0449@student.monash.edu

## Technical and Professional Strengths

	Technical	Professional
Suhani	<b>Programming Languages:</b> Java, HTML/CSS, Python (a little)  <b>Design Tools:</b> Figma, Adobe Illustrator, Photoshop, InDesign  <b>Technologies &amp; Tools:</b> Git, VS Code, Android Studio, Trello  <b>Operating Systems:</b> Windows	<b>Collaborative Problem-Solving:</b> Working with teams, using feedback, and cross-disciplinary thinking  <b>Clear Communication:</b> Strong verbal and written communication skills  <b>Resilience and Adaptability:</b> Staying calm under pressure,

	<b>Other Technical Skills:</b> User-Centered Design, Prototyping, Agile Methodologies	adjusting to new challenges
Gracia	<b>Programming Languages:</b> Python, Java, JavaScript, HTML/CSS <b>Design Tools:</b> Figma, Adobe Illustrator, Photoshop, InDesign, Lucidchart <b>Technologies &amp; Tools:</b> Git, VS Code, Android Studio, Trello <b>Other Technical Skills:</b> Project Planning, UX/UI Design, Visual Communication, Documentation and Writing	<b>Collaboration and Teamwork:</b> Working with the team, resolving conflicts, and coordinating efforts <b>Time Management:</b> Efficient task prioritisation and adherence to deadlines <b>Adaptability:</b> Integrating feedback and adjusting to constraints
Jocelyn	<b>Programming languages:</b> Python, Java, JavaScript, HTML/CSS, R <b>Design Tools:</b> Figma, Photoshop, Lucidchart <b>Technologies &amp; Tools:</b> Git, VS Code, Android Studio <b>Other technical skills:</b> Agile & Scrum Methodologies, UI/UX Principles, Requirement Analysis	<b>Communication &amp; Collaboration:</b> Effectively working in teams by actively participating in discussions <b>Problem-Solving:</b> Analysing, identifying and solving challenges that come along the way <b>Attention to Detail:</b> Ensuring accuracy and quality in all tasks

### Fun Fact

Suhani: I used to live in Africa

Gracia: I lived in Queensland

Jocelyn: I don't like cheese

## Team Schedule

### **Weekly meeting:**

Tuesday (between 12 pm - 4 pm) - in person

[Zoom link](#)

### **Workload distribution:**

[Workload Distribution Board](#)

## Technology Stack and Justification

Programming language: **Java**

As a team, we have more experience with Java, having used it since our first year of university and us generally just being more comfortable with it. We believe it is easier to apply OOP concepts in Java, which allows us to develop more efficiently and maintain higher code quality. We also feel that using this language will give us more flexibility to explore advanced features compared to Python.

[Link to Git](#)

[Link to prototypes](#)

## Part 2: User Stories

User Story	INVEST Criteria
As a player, I want to be able to invite a friend to an online match so that I can easily play with them in a private game.	<p><b>I:</b> This feature can be developed separately from other multiplayer mechanics, such as matchmaking or ranking systems.</p> <p><b>N:</b> The specifics of how invitations are sent (e.g., via a game-generated code, direct friend list, or external link) can be refined based on user needs.</p> <p><b>V:</b> Enhances the multiplayer experience by allowing friends to easily connect and play together, making the game more social and engaging.</p> <p><b>E:</b> The feature's scope is clear and can be broken down into components such as invitation sending, friend acceptance, and lobby creation, making it easier to estimate development time.</p> <p><b>S:</b> The functionality is focused and can be implemented incrementally, such as starting with basic friend invites before expanding to additional features like chat or team modes.</p> <p><b>T:</b> The system can be verified by checking that players successfully send and receive invitations and can connect to a match without issues.</p>
As a player, I want an interactive tutorial that clearly explains movement and win conditions so that I can quickly and effectively learn how to play the game.	<p><b>I:</b> The tutorial can be developed separately from other gameplay mechanics and UI elements, making it modular.</p> <p><b>N:</b> The specifics of the tutorial (e.g., step-by-step guidance, hands-on practice, voice-over instructions) can be adjusted based on user feedback and feasibility.</p> <p><b>V:</b> Helps new players understand the game mechanics, leading to a smoother onboarding experience and increased player retention.</p> <p><b>E:</b> The tutorial's scope is well-defined, allowing developers to estimate the time needed to create step-by-step guidance, interactive elements, and completion tracking.</p> <p><b>S:</b> The feature is focused on one core aspect: teaching the game. It can be developed incrementally, starting with basic movement and then adding more advanced strategies.</p> <p><b>T:</b> The tutorial can be tested by tracking player completion rates, engagement levels, and whether users understand the game better after going through it.</p>
As a player, I want a hint system that provides strategic suggestions when I am stuck so that I can receive guidance and	<p><b>I:</b> The hint system can be implemented separately from core gameplay mechanics, ensuring it does not interfere with game functionality.</p> <p><b>N:</b> The specifics of the hint system (e.g., displaying subtle hints vs. direct suggestions, cooldowns, or limited uses per game) can be adjusted based on user needs and developer constraints.</p>

continue playing the game.	<p><b>V:</b> Enhances the player experience by reducing frustration, encouraging learning, and improving engagement, especially for newer players.</p> <p><b>E:</b> The scope is well-defined, making it easy for developers to estimate the time and effort required to design and integrate hint logic.</p> <p><b>S:</b> The feature is focused solely on providing hints, making it a manageable, standalone addition that can be implemented incrementally.</p> <p><b>T:</b> The effectiveness of the hint system can be tested through user feedback, tracking hint usage rates, and measuring player progression improvements.</p>
As a player, I want there to be an option of dark mode so that the visuals are easier on my eyes if I am playing at night.	<p><b>I:</b> The dark mode feature can be developed separately from gameplay mechanics.</p> <p><b>N:</b> The details of how hints are provided (e.g., text-based suggestions, visual indicators, or AI-driven recommendations) can be adjusted based on user needs.</p> <p><b>V:</b> Provides accessibility and improves user comfort.</p> <p><b>E:</b> Straightforward implementation (UI theme switch), easy to estimate.</p> <p><b>S:</b> A simple UI feature that is manageable within a sprint.</p> <p><b>T:</b> Users should be able to switch themes without issues.</p>
As a casual player, I want an undo button so that I can fix accidental moves.	<p><b>I:</b> Can be implemented without affecting other mechanics.</p> <p><b>N:</b> The number of undos allowed can be adjusted.</p> <p><b>V:</b> Improves user experience, especially for casual players.</p> <p><b>E:</b> Clearly defined functionality, making it easy to estimate effort.</p> <p><b>S:</b> A simple feature that can be implemented in a sprint.</p> <p><b>T:</b> Players should be able to undo and redo moves as intended.</p>
As a player, I want a hint system that suggests the best possible next move based on my current game state so that I can improve my strategy and make more informed decisions.	<p><b>I:</b> The hint system can be developed and implemented separately from other features without affecting the core gameplay mechanics.</p> <p><b>N:</b> The level of hints (e.g., basic suggestions vs. advanced strategy tips) can be discussed and refined based on user feedback.</p> <p><b>V:</b> Helps players enhance their strategic thinking, making the game more engaging and educational.</p> <p><b>E:</b> The complexity of the hint system (basic suggestions vs. AI-driven strategy recommendations) can be assessed, allowing for realistic planning and development.</p> <p><b>S:</b> A well-defined feature that can be developed and tested in a reasonable timeframe without requiring extensive changes to the existing game mechanics.</p> <p><b>T:</b> The system can be verified by checking if hints are relevant, if they provide useful guidance, and if they contribute to an improved player experience.</p>

<p>As a player, I want a dark mode setting that adjusts colours and contrast dynamically so that it reduces eye strain and provides a more comfortable gaming experience at night.</p>	<p><b>I:</b> Dark mode can be implemented separately from other features without affecting core gameplay.  <b>N:</b> The specifics of dark mode (e.g., automatic toggle, manual toggle, different contrast settings) can be discussed and adjusted based on user feedback.  <b>V:</b> A well-designed UI improves the game experience.  <b>E:</b> The scope is clear and can be accurately estimated (modifying the colour scheme and adding a toggle feature).  <b>S:</b> A self-contained feature that can be completed in a single sprint.  <b>T:</b> This can be verified by checking if users can switch to dark mode, if contrast adjustments are appropriate, and if it improves readability and visual comfort.</p>
<p>As a developer, I want tooltips on hover so that players can learn controls quickly.</p>	<p><b>I:</b> It does not affect other mechanics outside of online play.  <b>N:</b> The length of the timer and penalty rules can be discussed.  <b>V:</b> Prevents slow matches and improves pacing.  <b>E:</b> Clearly defined functionality.  <b>S:</b> A simple feature that can be implemented in a sprint.  <b>T:</b> Can verify if the timer properly limits turns.</p>
<p>As a worker on board, I want to climb to the third level so that my player wins the game.</p>	<p><b>I:</b> A player's movement and win condition don't depend on external factors.  <b>N:</b> Movement rules can be adjusted to change win conditions.  <b>V:</b> Reaching the third level is the game's win condition.  <b>E:</b> Board analysis can determine whether a player has a viable path to the top.  <b>S:</b> The action is a single-turn action.  <b>T:</b> The game will confirm the win after a worker reaches the third level</p>
<p>As a worker on board, I want to construct a dome on top of a third-level building so that I can prevent my opponent from winning.</p>	<p><b>I:</b> A player can block an opponent's move without requiring another piece to intervene.  <b>N:</b> The rule about dome placement could be altered to allow more flexibility in the game.  <b>V:</b> Dome placement can prevent opponents from reaching the third level, making it a key defensive tactic.  <b>E:</b> The success of the action is clear if a dome exists on the top of a building.  <b>S:</b> Placing a dome is a single-turn action without any additional steps.  <b>T:</b> The board state will indicate if the dome has been built.</p>
<p>As a player with colour blindness, I want the game to use colour-blind friendly palettes so that I can</p>	<p><b>I:</b> A player's ability to recognise elements of the game only depends on themselves.  <b>N:</b> The method of making the game colour-blind friendly can include patterns or higher contrast.  <b>V:</b> Without colour-friendly adjustments, players might struggle to distinguish game components.</p>

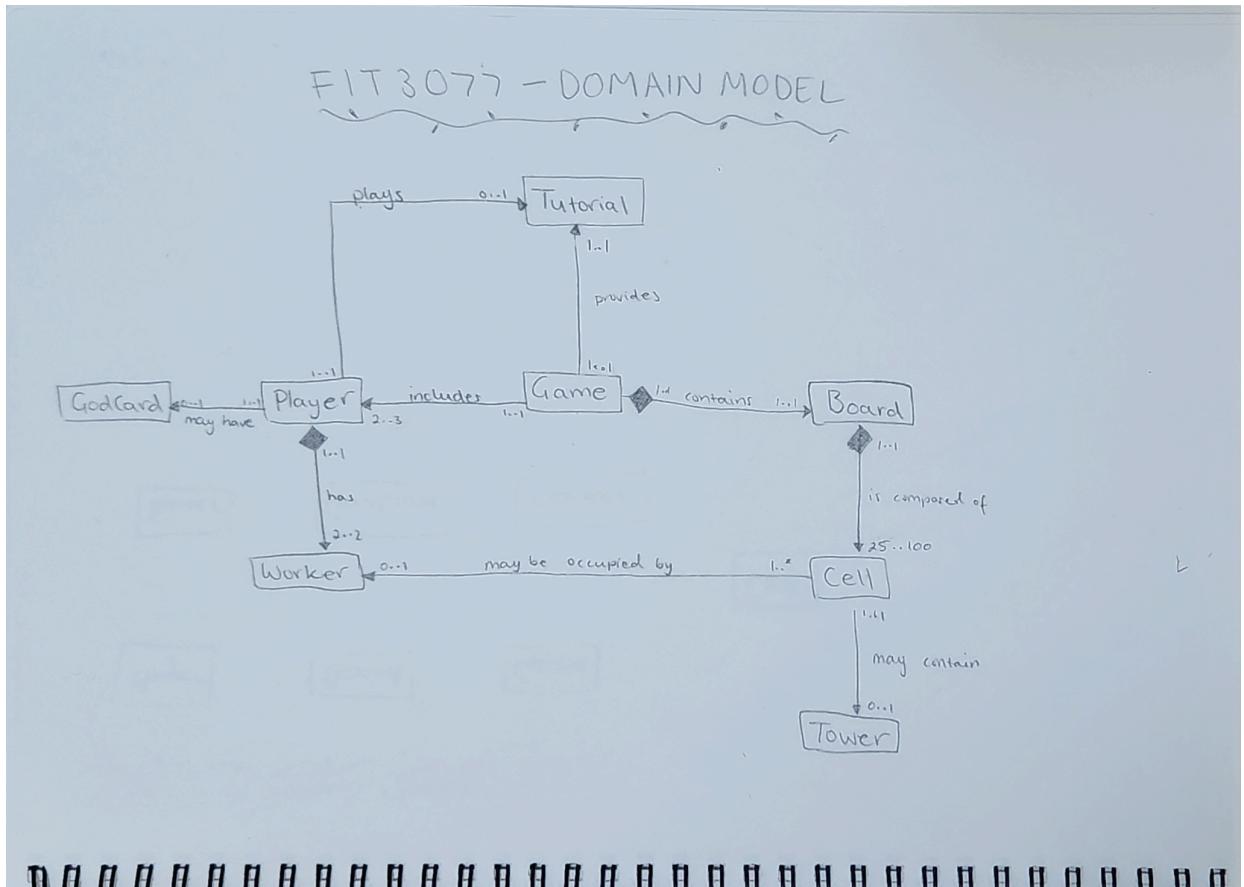
differentiate the different elements of the game.	<p><b>E:</b> User testing with colour-blind individuals can determine if the game is catered to them.</p> <p><b>S:</b> Adding or changing colours is a small design change.</p> <p><b>T:</b> Players with colour blindness can give feedback on the game.</p>
As a player with a God Card, I want a clear description of my card's ability so that I can make the best use of my ability without any confusion.	<p><b>I:</b> Understanding the ability of a God Card doesn't require external clarification.</p> <p><b>N:</b> The description of each God Card could be improved to improve clarity.</p> <p><b>V:</b> Without a clear description of a card's ability, the player might misinterpret the card's power.</p> <p><b>E:</b> User feedback could be used to assess whether each ability is described well.</p> <p><b>S:</b> Updating the descriptions of the God Cards is a minor change.</p> <p><b>T:</b> Rule check can verify whether the descriptions are clear enough.</p>
As a player, I want a user-friendly interface with intuitive controls so that I can navigate the game easily and avoid misplacing moves.	<p><b>I:</b> The ability to interact with the interface doesn't rely on hidden or unclear controls.</p> <p><b>N:</b> The game interface could be adjusted.</p> <p><b>V:</b> An intuitive layout helps prevent misclicks, improving game experience.</p> <p><b>E:</b> User testing is conducted to measure how quickly new players learn the interface.</p> <p><b>S:</b> UI adjustments are small changes that don't affect game mechanics</p> <p><b>T:</b> Usability testing can confirm if players are able to avoid common errors.</p>
As a player with the Pan God Card, I want my worker to instantly win the game if they move down two or more levels in a single turn, so that I win the game.	<p><b>I:</b> The God Card's ability is unaffected by the opponent's moves.</p> <p><b>N:</b> The win condition could be modified.</p> <p><b>V:</b> Each God Card has different powers, creating a unique way to win.</p> <p><b>E:</b> Checking the board confirms whether the worker moved down at least two levels to trigger the win.</p> <p><b>S:</b> Winning by stepping down two or more levels happens in a single movement.</p> <p><b>T:</b> A rule check can confirm if a move satisfies the win condition.</p>
As a player, I want a timer for each move so that the game progresses smoothly without unnecessary waiting.	<p><b>I:</b> The addition of a timer doesn't affect the game's flow.</p> <p><b>N:</b> The duration of the timer can be adjusted for different needs.</p> <p><b>V:</b> A timer keeps the game engaging and prevents long turns.</p> <p><b>E:</b> Tracking the average move time will indicate if the timer shortens playing time.</p> <p><b>S:</b> Implementing a timer is a simple change.</p> <p><b>T:</b> Observation can be done to see whether players adhere to the time limit.</p>

<p>As a player, I want the game to have visually engaging components and clear iconography so that playing the game feels immersive and enjoyable.</p>	<p><b>I:</b> The game's overall design doesn't affect core mechanics.  <b>N:</b> The game's design could be modified.  <b>V:</b> A visually appealing game can enhance immersion.  <b>E:</b> Feedback from players can determine if the game's design contributes to their playing experience.  <b>S:</b> Design improvements can be made incrementally.  <b>T:</b> User surveys can be conducted to confirm players' satisfaction with the game design.</p>
<p>As a player using a touch device, I want to be able to tap and drag my worker to move and build so that the controls feel intuitive on mobile.</p>	<p><b>I:</b> This doesn't rely on other game mechanics, only affecting mobile controls.  <b>N:</b> The implementation details can be discussed.  <b>V:</b> Accessibility and usability are improved for mobile players.  <b>E:</b> Effort can be estimated by analysing UI interactions.  <b>S:</b> Focus is solely on movement and building, not other gameplay mechanics.  <b>T:</b> Testing can verify that workers move and build correctly when tapped/dragged.</p>
<p>As a player with a hearing impairment, I want all important audio cues to have visual indicators so that I don't miss critical game events.</p>	<p><b>I:</b> This doesn't interfere with core gameplay mechanics or the game's rules.  <b>N:</b> The type of visual indicators can be adjusted based on feasibility.  <b>V:</b> It ensures inclusivity and accessibility.  <b>E:</b> The work required to add visual cues can be estimated.  <b>S:</b> The scope is limited to translating existing audio cues into visual indicators.  <b>T:</b> Sound will be disabled to verify if all relevant events have visual feedback.</p>
<p>As a player with the Athena God Card, I want to prevent my opponent's workers from moving up if my worker moves up this turn so that I can strategically limit their movement.</p>	<p><b>I:</b> This only applies when Athena is in play and doesn't affect other game mechanics.  <b>N:</b> The specific implementation can be adjusted.  <b>V:</b> Modifying movement dynamics adds depth and strategy to the game.  <b>E:</b> Effort can be estimated by reviewing movement logic and conditionally restricting moves.  <b>S:</b> This only affects a single game rule related to movement when Athena is active.  <b>T:</b> Testing can verify that opponents' workers cannot move up when Athena's effect is triggered.</p>
<p>As a player, I want to customise my worker's appearance so that I can personalise my</p>	<p><b>I:</b> This doesn't affect gameplay mechanics and is purely visual.  <b>N:</b> The types of customisation can be adjusted.  <b>V:</b> Allowing personalisation enhances player engagement.  <b>E:</b> Effort can be estimated based on available assets and UI requirements.</p>

gameplay experience.	<p><b>S:</b> This is limited to modifying character models and UI customisation options.</p> <p><b>T:</b> Testing includes selecting different customisation options and confirming whether they appear correctly.</p>
As a player, I want an option to view move history so that I can analyse previous moves and improve my strategy.	<p><b>I:</b> This doesn't interfere with gameplay, only provides a log of past actions.</p> <p><b>N:</b> The format of the move history can be adjusted.</p> <p><b>V:</b> This helps players review strategies and learn from past games.</p> <p><b>E:</b> Effort can be estimated based on how game state changes are recorded.</p> <p><b>S:</b> This only focuses on displayed past moves, not additional analytics.</p> <p><b>T:</b> Testing will involve playing the game and checking the move history to ensure accuracy.</p>
As a player with a Demeter God Card, I want my worker to build twice (but not on the same space) so that I can create new strategic opportunities.	<p><b>I:</b> This only applies when Demeter is in play and doesn't affect other god powers.</p> <p><b>N:</b> The UI feedback (highlighting buildable spaces) can be refined.</p> <p><b>V:</b> Allowing controlled extra builds enhances strategic gameplay.</p> <p><b>E:</b> Impact on the build logic and UI changes can be assessed.</p> <p><b>S:</b> This only affects the building phase.</p> <p><b>T:</b> Testing can verify that a player using Demeter can build twice, but not on the same space.</p>
As a player with a Minotaur God Card, I want my worker to push an opponent's worker if they are in my movement path so that I can manipulate the board.	<p><b>I:</b> This only affects movement and doesn't modify other mechanics.</p> <p><b>N:</b> The push mechanics can be refined.</p> <p><b>V:</b> This adds a unique gameplay mechanic and increases strategic depth.</p> <p><b>E:</b> Effort can be estimated based on modifying the movement logic.</p> <p><b>S:</b> This focuses on push mechanics without affecting building or winning conditions.</p> <p><b>T:</b> Testing can verify that a Minotaur worker can push another worker correctly.</p>
As a player who disconnects, I want the ability to rejoin my match within a set time limit so that I don't lose unfairly due to connection issues.	<p><b>I:</b> This affects session management, but doesn't alter gameplay mechanics.</p> <p><b>N:</b> The time limit and reconnection method can be fine-tuned.</p> <p><b>V:</b> This prevents unfair losses due to accidental disconnections.</p> <p><b>E:</b> Effort can be estimated based on session persistence and reconnection logic.</p> <p><b>S:</b> This is limited to session tracking and restoring player state.</p> <p><b>T:</b> Testing involves disconnecting and reconnecting within the time limit to confirm that the game resumes properly.</p>

# Part 3: Domain Model

Final version:



## Detailed justifications

When designing the domain model, we made the assumption that even though players can customise their board size, there is a limit to how big the board can be so that it still fits on the screen. The smallest board is 5x5 in size and can be customised up to 10x10. Another modification that we made was the addition of a tutorial that will be shown at the beginning of the game.

### 1. Game

This entity exists because it is the main driver of the program. It coordinates the gameplay, players, and board.

Relationships:

- Contains 1 board: For each game, the players play on one and only one board.  
This is a composition relationship (filled diamond), because the board can't be created independently. It depends on the game.
- Includes 2 to 3 players: The game is usually played by 2 players, but a third player can be added as an extension.
- Provides 0 to 1 tutorial: Players can choose to play the tutorial to learn about the game, or they can skip it.

## 2. Player

This entity exists because players are an essential part of the game. It represents each player of the game.

Relationships:

- Has 2 workers: Each player is allowed to place exactly two workers on the board.  
This is a composition relationship (filled diamond), because the workers are created specifically for the players and don't exist independently.
- May have 0 to 1 God Card: God Cards are optional since not all game modes enable them. When used, each player can have only one God Card.
- Plays 0 to 1 tutorial: Players can choose to play the tutorial to learn about the game, or they can skip it.

## 3. Board

This entity exists because the board provides the playing field on which players place their workers and build towers.

Relationship:

- Associated with *Cell*: It is composed of 25 cells (5x5) minimum and a maximum of 100 cells (10x10). The board is made up of multiple individual cells. While the standard game uses a 5x5 grid (25 cells), an extension will be implemented to allow players to customise the board size. Hence, the board consists of at least 25 cells and can have as many as 100 cells.
  - This is a composition relationship (filled diamond) because cells are part of the board and can't exist independently.

## 4. Cell

This entity represents a single position on the board where workers move and build. It's needed to track whether the cell is occupied and the tower level built on it.

Relationships:

- Associated with *Worker*: A cell can either be empty or occupied by a worker.
- Associated with *Tower*: A cell may have a tower, but not all cells do.

## 5. Tower

This represents the buildings constructed by players during the game, which consist of 1 to 3 levels and may be capped with a dome.

Relationship:

- Associated with *Cell*: A tower is always placed on a specific cell.

## 6. Worker

This represents the player-controlled piece that moves and builds on the tower. 2 workers belong to each player.

#### Relationships:

- Belongs to *Player*: Each worker is controlled by a single player.
- Occupies *Cell*: A worker is always on a specific cell.

#### 7. GodCard

This entity is included as it provides special abilities that modify the standard gameplay rules. Each ability is different depending on which card is chosen. There are 32 cards to choose from.

#### Relationship:

- Associated with *Player*: A player may pull 1 God Card, but the opponent will not pull the same card.

#### 8. Tutorial

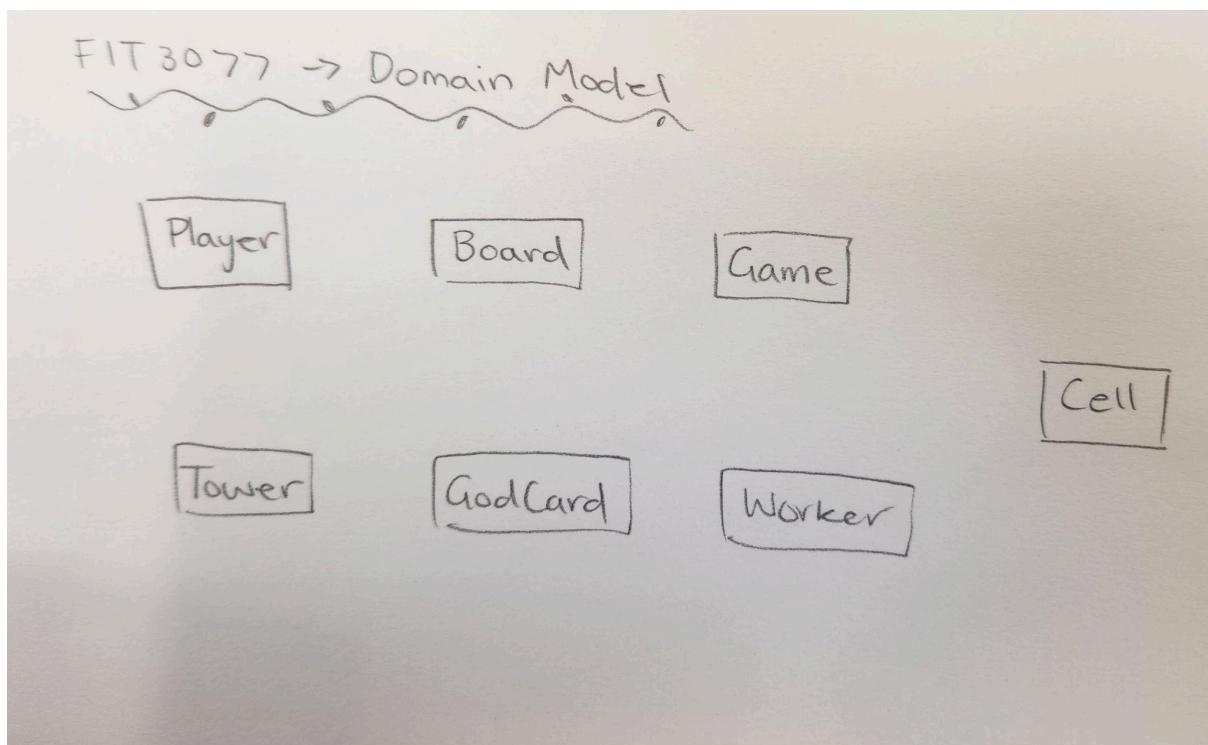
The team has decided to model the domain with this entity as an extension because we believe it is a great assistance to most new players in understanding the game. We, as a team, struggled to understand the gameplay when exploring Santorini, as we felt that the instructions given were not very clear. This entity provides interactive guidance to new players, explaining movement, building, and win conditions through step-by-step instructions if needed.

#### Relationship:

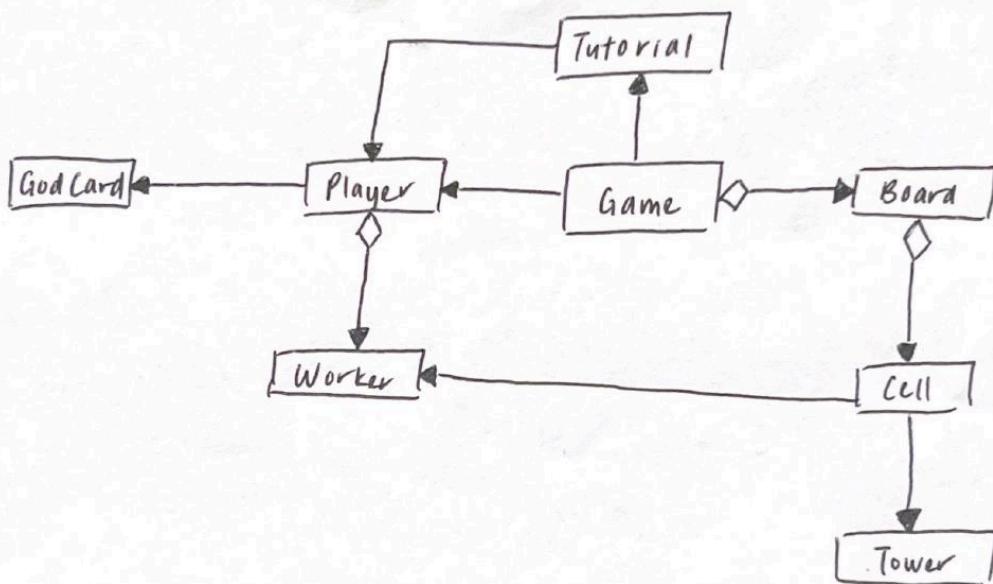
- Associated with *Game*: The tutorial is a feature of the game, but can be skipped.

#### Previous domain models:

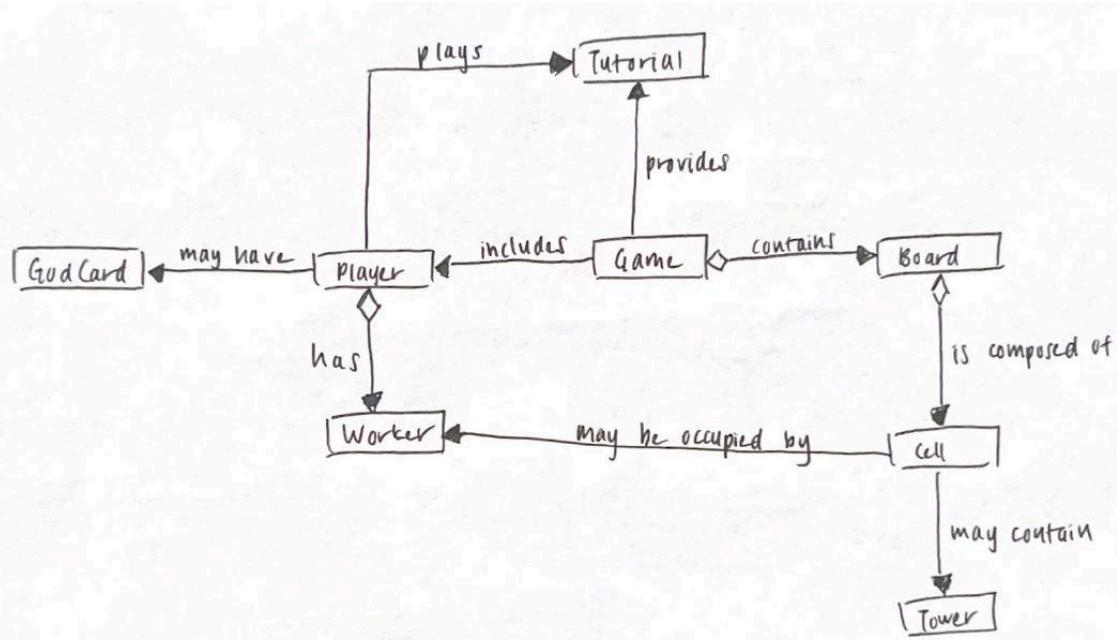
##### 1. V1:



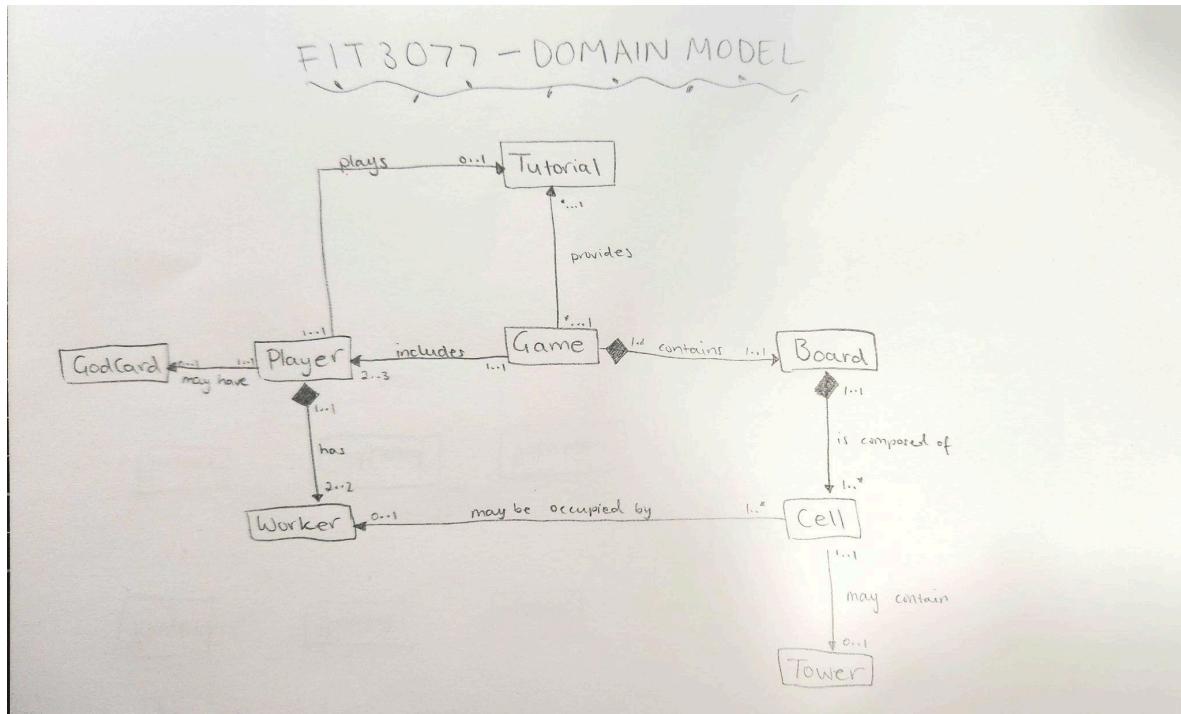
2. V2:



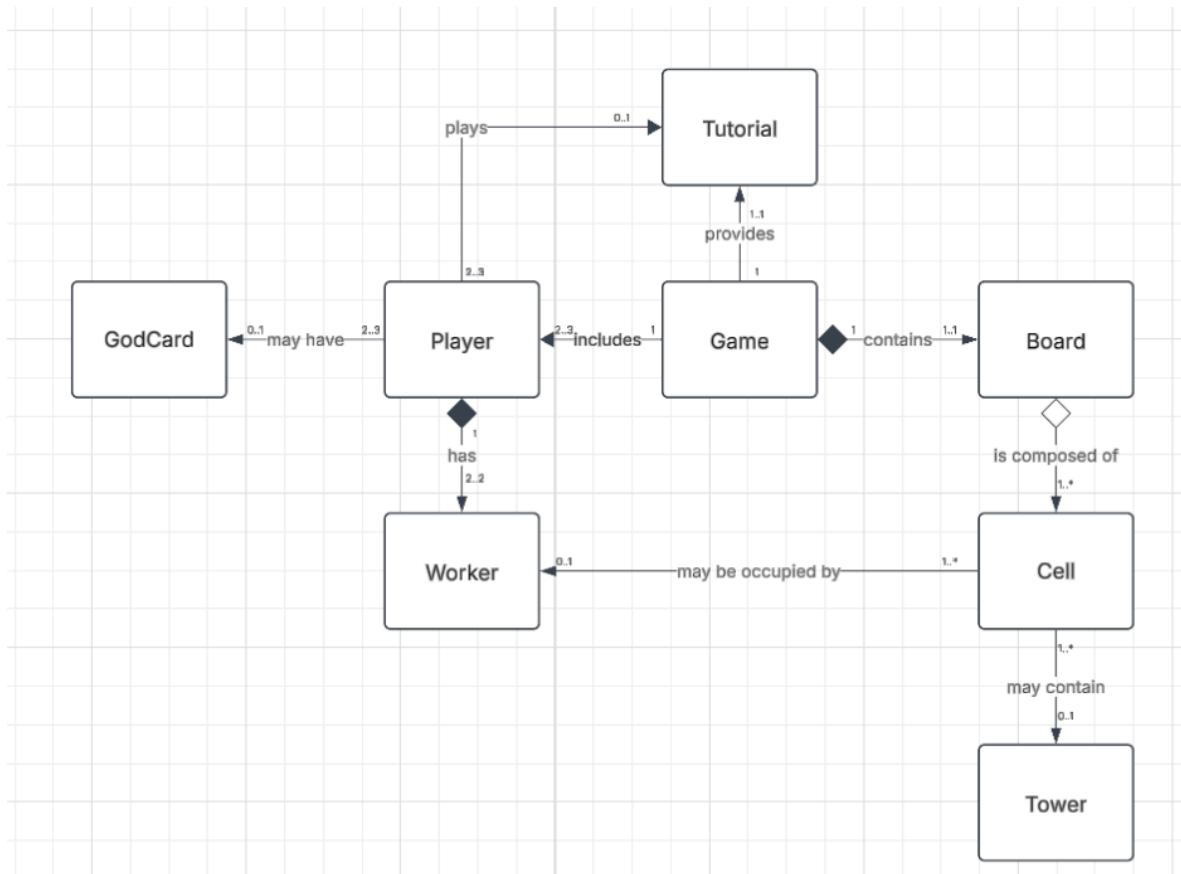
3. V3:



#### 4. V4:



Trello version:



# Part 4: Basic UI Design

## Team Figma Board

### Lo-fi prototype:

<https://www.figma.com/proto/CVCTFGnnr7mCRtKErLSUR4/Santorini-Prototype?node-id=0-1&t=MriXu2dDzv3PlpmZ-1>

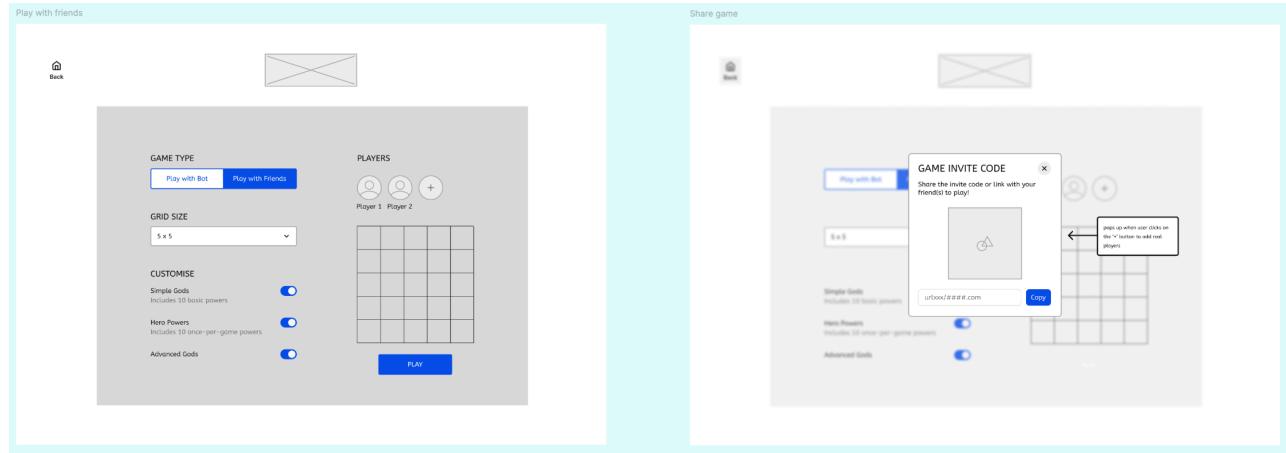
### Screenshots:

#### 1. Homepage



#### 2. Game Customisation

The image displays two side-by-side screens of a game customization interface. Both screens feature a header with a back button and a grid preview icon. The left screen, titled "Game customisation", contains sections for "GAME TYPE" (with "Play with Bot" selected), "GRID SIZE" (set to "5 x 5"), "CUSTOMISE" (with "Simple Gods" and "Hero Powers" options), and a "PLAY" button. The right screen, titled "Game customisation - dropdown options", shows an expanded "GRID SIZE" dropdown menu with options like "5 x 5", "6 x 6", "7 x 7", etc., and a "PLAY" button. A callout box points to the "PLAY" button on the right screen, stating: "option to add more players, which in this case are bots, which are simple and can't be removed if they ever lose a player". Another callout box points to the "PLAY" button on the right screen, stating: "preview of the board to above user the grid size they've chosen, and it's always updated when user picks a different grid size".



### 3. Tutorial

**Tutorial**

**Tutorial - Place worker**

**Tutorial - Move**

**Tutorial - Move**

**Tower - Build**

**Tower Build**

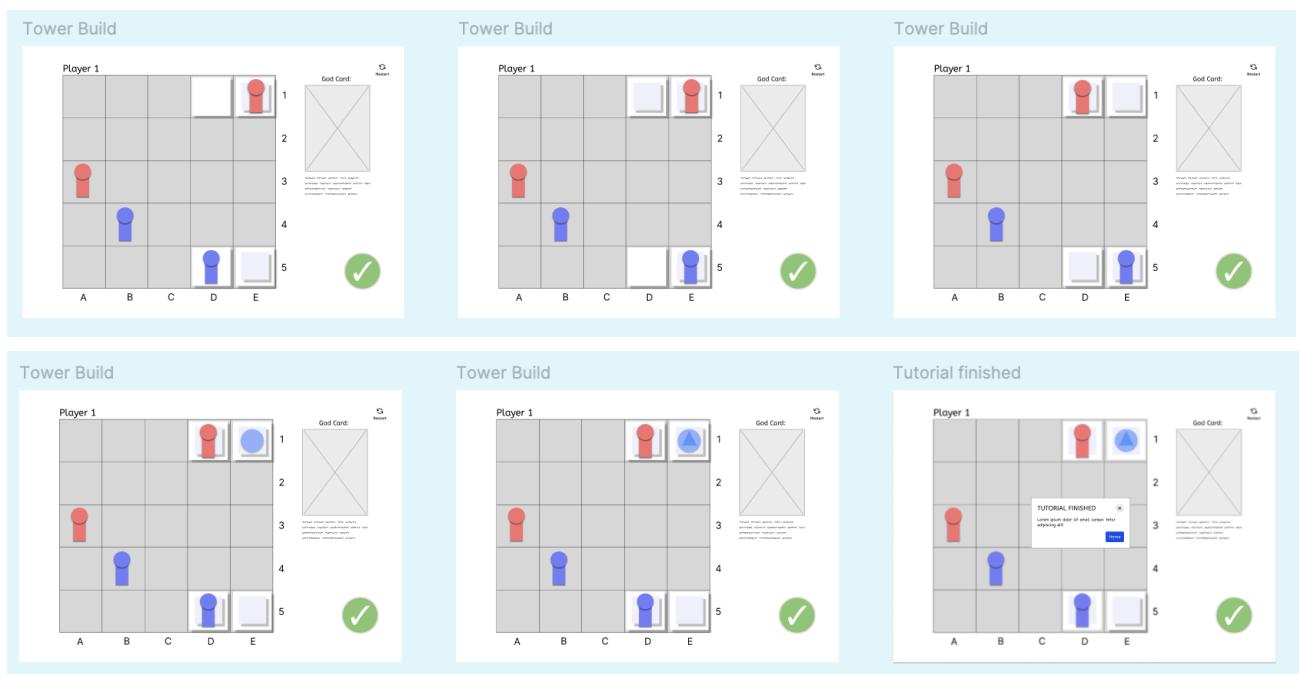
**Tower Build**

**Tower Build**

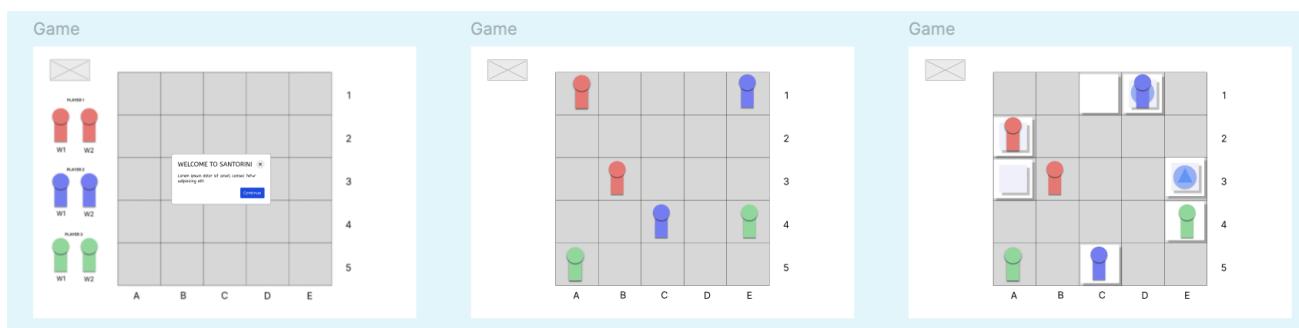
**Tower Build**

The tutorial consists of several steps illustrating game mechanics:

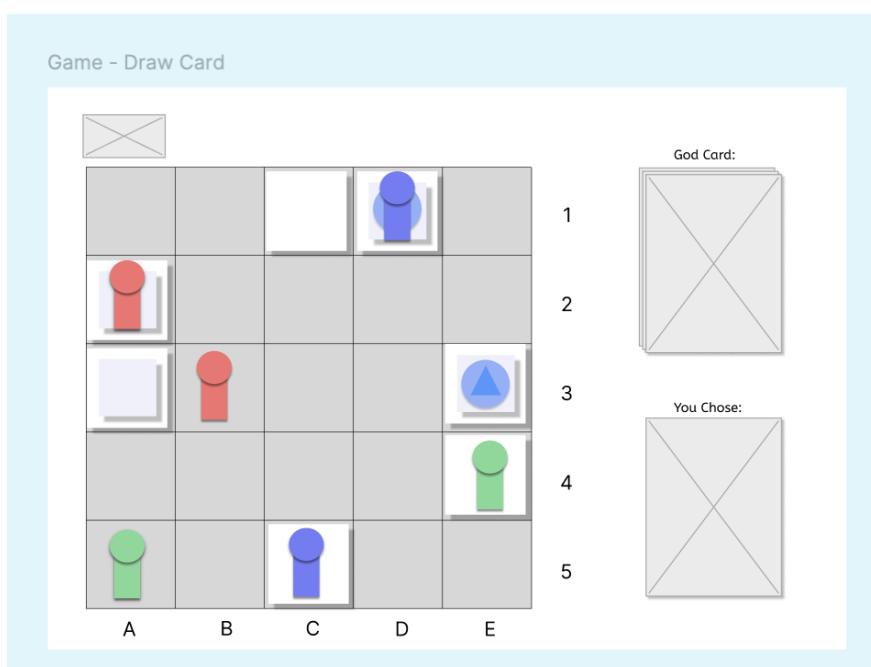
- Tutorial:** Welcome screen with instructions: "WELCOME TO SANTORINI! Learn basic rules of santorini before playing on".
- Tutorial - Place worker:** Player 1 places workers on the board. A tooltip says: "Place your workers! You can place workers on empty squares of the grid. Click on the button when you're done".
- Tutorial - Move:** Player 1 moves workers. A tooltip says: "Move your workers! You can move workers from one square to another if there's an empty square next to it. Click on the button when you're done".
- Tutorial - Move:** Another move step for Player 1.
- Tower - Build:** Player 1 builds a tower on square D1. A tooltip says: "Build a tower! You can build towers on empty squares of the grid. Click on the button when you're done".
- Tower Build:** Another tower build step for Player 1.
- Tower Build:** Another tower build step for Player 1.
- Tower Build:** Another tower build step for Player 1, showing a tooltip about God Cards.



#### 4. Game



#### 5. Draw Card

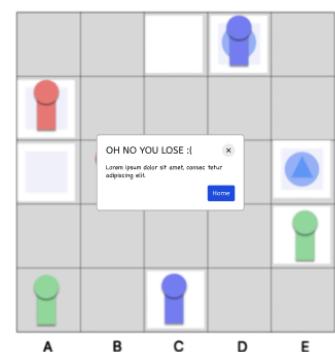


## 6. Game End

Win



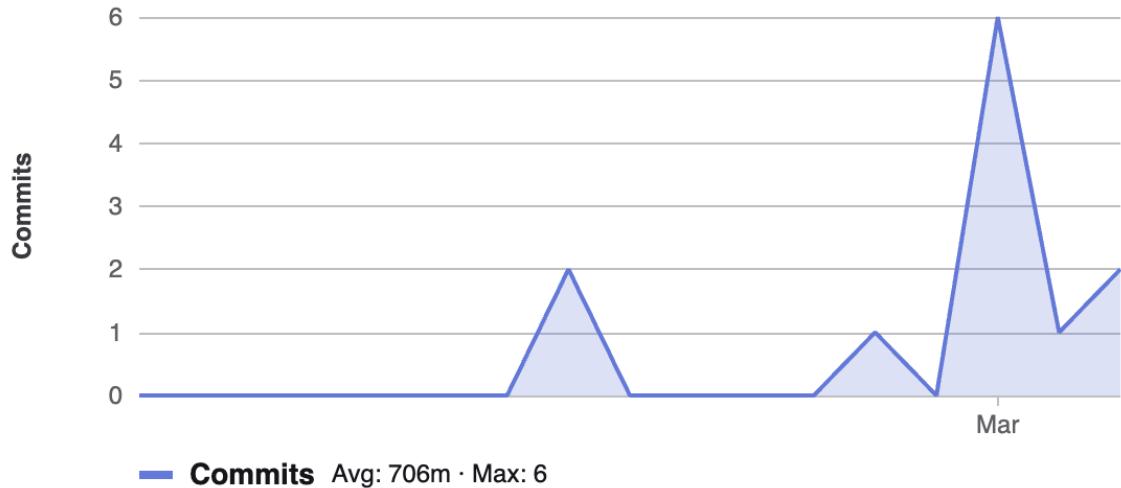
Lose



# Contributor Analytics

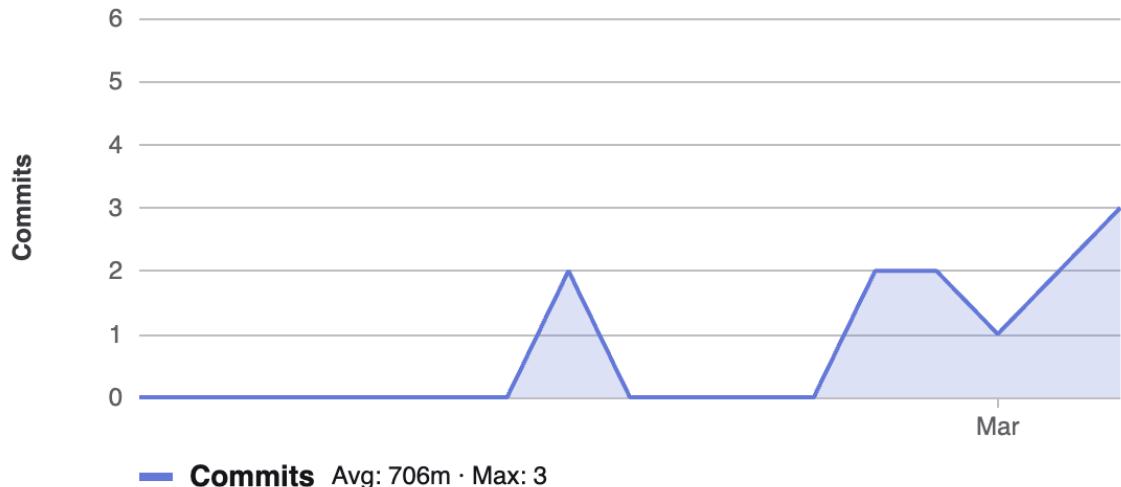
## sgad0010

12 commits (sgad0010@student.monash.edu)



## gyap0005

12 commits (gyap0005@student.monash.edu)



## Jocelyn Florentin Tanujaya

13 commits (jtan0449@student.monash.edu)

