

## User Stories

1. As a **player**, I want to move my worker to an empty adjacent space so that I can position them strategically.
  - Independent: This story focuses solely on worker movement mechanics and does not depend on other features (e.g., building blocks).
  - Negotiable: The definition of "**adjacent space**" (e.g., diagonal movement, edge cases) can be refined during implementation.
  - Valuable: It has a direct effect on the player's strategic planning. By giving the player more control over their actions, it improves their experience.
  - Estimable: Estimated by assessing movement logic complexity (e.g., checking for empty spaces, ensuring moves are within bounds).
  - Small: It's small enough because movement is a single action per turn and can be implemented within one iteration.
  - Testable: Testers can verify valid moves (e.g., no movement to occupied spaces or out-of-bounds).
2. As a **player**, I want to manually place my workers on unoccupied spaces during game setup so that I can strategically choose their starting positions.
  - Independent: This story focuses solely on the manual placement mechanic and does not depend on other features (e.g., movement, or building)
  - Negotiable: The starting placement rules can be refined during implementation (e.g., minimum distance from opponent's workers, etc)
  - Valuable: Allowing players to manually place workers is critical for strategy/planning,
  - Estimable: Estimated by assessing UI design for worker placement and validation logic.
  - Small: It's a one time action during setup and can be implemented within a single iteration
  - Testable: Testers can verify workers are placed on valid, unoccupied spaces and that overlapping placements are blocked.
3. As a **player**, I want to pick the color of my workers so that I easily differentiate them apart from my opponents.
  - Independent: This story focuses on color selection during setup
  - Negotiable: The color options can vary (e.g., predefined color selection vs custom colors).
  - Valuable: Critical for clarity of gameplay, where it prevents confusion between players
  - Estimable: Estimated by the UI for color picker, applying colors to workers, and validation logic (where no players share the same color)
  - Small: Limited to the setup phase and confined to a single UI component (color picker). No impact on core gameplay logic.
  - Testable: Verify players cannot select colors that have been picked by other players and that workers retain their chosen colors throughout the game.

4. As a **player**, I want to achieve victory by moving one of my workers on top of the third level of a building, so that I can fulfill the win condition of the game.
  - Independent: This story focuses on detecting wins when a worker reaches the third level of a building.
  - Negotiable: The visual feedback and timing of the victory confirmation can be refined during development.
  - Valuable: Provides clear feedback when a player wins, reinforcing the core objective of the game.
  - Estimable: Estimated by checking the level of the tile a worker moves to and triggering the win state when it is the third level.
  - Small: Limited to a win state condition check and a corresponding UI update.
  - Testable: Confirm that the win state is triggered immediately when a worker moves onto a third-level tile, and that the game transitions to the victory screen accordingly.
5. As a **player**, I want to win the game if all my opponent's workers are not able to perform any move or build action during their turn, so that I can claim victory when their actions are completely restricted.
  - Independent: This story focuses on detecting when an opponent has no legal move or build options, without altering other core gameplay mechanics.
  - Negotiable: The specific condition checks that define "unable to move or build" (e.g., a one-turn check or continuous state) can be refined during development.
  - Valuable: Provides an alternative win condition that rewards strategic play by enabling players to secure victory by immobilizing their opponents.
  - Estimable: Estimated by checking at the start of an opponent's turn to determine if they have any valid movement or building options and triggering the lose state if none exist.
  - Small: Limited to a lose state condition check during turn evaluation and a corresponding UI update.
  - Testable: Confirm that if a player's turn begins with no legal moves or building actions available, the game immediately triggers the lose state for the current player.
6. As a **player**, I want to assign turn order by default based on age (e.g., the oldest or youngest goes first), with the option to manually select or roll a die to decide who gets the first turn, so that I can quickly and fairly set up the game.
  - Independent: This feature is self-contained and does not rely on other features. It only affects the set up
  - Negotiable: The method of selecting the first player (UI options, automatic vs. manual selection) can be modified. The way to achieve this from a coding perspective is flexible.
  - Valuable: Gives flexibility in game setup, a small but important quality of life
  - Estimable: Requires UI work and logic implementation.

- Small: Limited to creating a simple interface for turn order selection and integrating basic logic for age-based, manual, or dice-based selection.
  - Testable: Verify that by default the first player is selected based on age, and that the manual selection and dice roll options correctly assign the first player when chosen.
7. As a **player**, I want to surrender or quit the game early at any time so that I could quickly start another game.
- Independent: The story focuses on player agency to stop a game before a winner is determined normally.
  - Negotiable: The surrender implementation and what happens after can be redefined later.
  - Valuable: Allows the player more control of whether they want to continue or surrender if they're not interested in the current game.
  - Estimable: The logic will alter the game state and break the loop so that the game stops.
  - Small: Limited to checking the game state.
  - Testable: Checks if the UI will return to the end game screen after surrendering.
8. As a **player**, I want to see visual highlights on all adjacent tiles that represent valid movement options (including lateral, upward, and downward moves) so that I can quickly assess where my worker can move—similar to how chess shows available moves.
- Independent: The story focuses on adding a UI overlay for displaying valid movement options without altering core movement logic.
  - Negotiable: The design of the tile highlight and special conditions can be refined during development.
  - Valuable: Provides immediate, clear feedback on all legal moves, enabling faster tactical decisions.
  - Estimable: Estimated by checking the validity of each adjacent tile for movement and updating the board UI accordingly.
  - Small: Limited to implementing a visual overlay during the move phase.
  - Testable: Confirm that only tiles representing legal moves (whether lateral, upward, or downward) are highlighted.
9. As a **player**, I want to see visual highlights on all adjacent tiles that represent valid building options (for blocks and domes) so that I can quickly assess where my worker can build—similar to how chess shows available moves.
- Independent: This story focuses on adding a UI overlay for displaying valid building options without altering core building logic.
  - Negotiable: The design of the highlight and any specific conditions can be refined during development.

- Valuable: Provides immediate, clear feedback on all legal build locations, enabling faster strategic decisions.
- Estimable: Estimated by checking the validity of each adjacent tile for building and updating the board UI accordingly.
- Small: Limited to implementing a visual overlay during the building phase.
- Testable: Confirm that only tiles representing legal building options (for blocks and domes) are highlighted.

10. As a **player**, I want to manually select where to build my blocks and then execute the build on my turn, so that I can strategically plan my moves

- Independent: This feature focuses on block placement and does not interact with movement and many other basic core features. While they do play together to form a bigger picture, the game at this point only needs to take into account god abilities and location of workers.
- Negotiable: The implementation details (e.g., UI method for selection, grid highlighting) can be adjusted.
- Valuable: This is a core mechanic, without it the game would not be the game.
- Estimable: The effort is straightforward. We need to handle user input, validating legal placements, consider gods abilities, and then update the game board.
- Small: Building can only be built during specific parts of the turn. While it requires game logic, it is not a major change.
- Testable: Can be tested by ensuring players can only build in legal spots and that the building updates correctly.

11. As a **player**, I want to select a god card and view their abilities before a game, so that I know what abilities each god has and incorporate their powers into my strategy.

- Independent: This feature only involves displaying information and does not affect game mechanics.
- Negotiable: The god power preview and selection UI can be adjusted.
- Valuable: Helps players make informed choices before starting the game. Without this, they might not know what their selected god does. Also gives the player more agency in performing strategies and increases replayability
- Estimable: The UI will display a list of available god cards and their powers before starting a game, then assign each player their chosen god cards
- Small: This is a self-contained feature that can be implemented with simple UI elements.
- Testable: Can be tested by checking if each god's ability is displayed correctly when previewed.

12. As a **player**, I want to clearly see the current turn phase (move or build) and know whose turn it is, so that I can plan my actions and time my god powers effectively.

- Independent: This feature is focused solely on displaying the current turn phase, independent of the core gameplay mechanics.
- Negotiable: The turn and phase display can be adjusted during development.
- Valuable: It enables players to quickly understand the game state and determine what actions are available to them at any moment.

- Estimable: UI display components, and assessing the integration of game flow logic
- Small: Limited to the turn and phase display UI.
- Testable: Verify that the UI correctly shows the current turn phase (move or build) and accurately displays whose turn it is throughout the game.

13. As a **player**, I want my turn to automatically end once I've completed a move followed by a build, so that the game maintains a smooth and uninterrupted pace.

- Independent: This feature operates independently of other mechanics, provided that the move and build actions are already implemented.
- Negotiable: The precise conditions and visual cues for ending the turn can be adjusted during development.
- Valuable: It enhances game flow by reducing downtime, ensuring that the game moves forward at a consistent pace.
- Estimable: The work required can be estimated by evaluating the integration of move/build actions with the turn-ending mechanism.
- Small: Only need to detect the completion of move and build actions and triggering the end-of-turn state
- Testable: It can be verified by simulating complete turn cycles to ensure the turn ends automatically only after both a move and a build are executed.

14. As a **beginner player**, I want to access game rules and keyword explanations during gameplay, so that I can quickly check them and make informed decisions,

- Independent: The story focuses on easing the learning process by allowing players to familiarize the rules while playing.
- Negotiable: The information can be displayed in different ways.
- Valuable: Allows the player to look up rules and keywords to avoid confusion during play.
- Estimable: Implement a submenu to display the rule book and glossary during a game.
- Small: Limited to UI displaying information.
- Testable: Check if the submenu opens and displays the correct information.

15. As a **player**, I want to check the action log that displays all my and my opponent's actions so that I can easily trace back actions done in current and previous turns.

- Independent: The story focuses on UI elements to display actions performed throughout the game.
- Negotiable: The action log display can be redefined.
- Valuable: Allows the player to trace back their or their opponent's actions.
- Estimable: The logic will record performed actions in order and display them.
- Small: Limited to action log UI.
- Testable: Check if every action performed is recorded, then check if those actions are displayed correctly in order.

16. As a **player**, I want to build domes on the third level so that I can block an opponent's worker from reaching the third level thereby preventing them from winning the game.

- Independent: This feature focuses solely on the dome construction mechanics and its effect on blocking worker movement.
- Negotiable: The specifics of dome placement and its visual representation can be refined during development.
- Valuable: It provides a strategic option to delay or prevent an opponent's victory, adding depth to gameplay.
- Estimable: Can be estimated by assessing the dome placement logic and its interaction with the winning conditions.
- Small: The scope is limited to dome building on fully built towers and ensuring it blocks opponent's workers to move up to the third level
- Testable: Verify that when a dome is built on a third-level tower, no worker can move onto or remain on that tower's top, thus preventing a win condition.

17. As a **player with Demeter's god card**, I want my worker to be able to build twice during my turn (on different spaces) so that I can speed up tower construction and block opponents strategies

- Independent: Focused solely on Demeter's double build mechanic
- Negotiable: The conditions under which the second build can happen (e.g., spacing, height limitations, and board restrictions) can be adjusted during development.
- Valuable: Provides a tactical advantage in controlling the board
- Estimable: Requires tracking build locations and establishing build restrictions
- Small: Limited to adding an optional second build action per turn.
- Testable: Verify the two builds occur on unique spaces and follow the height rules.

18. As a **player with Artemis's god card**, I want my worker to be able to move twice during my turn (without returning to its starting space) so that I can outmove opponents and secure advantageous positions.

- Independent: Focused solely on Artemis's double move mechanic
- Negotiable: The specific rules regarding valid movement options for the second move, including restrictions on returning to the starting space, can be adjusted during development.
- Valuable: Provides a tactical advantage by allowing greater mobility and secure more strategic positions
- Estimable: Defining the additional move logic, implementing movement restrictions, and integrating it with the existing movement logic
- Small: Limited to adding one extra move per turn
- Testable: Verify that the worker can perform another move in one turn, and that the second move does not return the worker to its starting position.

19. As a player, I want to configure the board size so that I can customize my playing experience **[Extension]**

- Independent: This feature can be developed independently provided the game supports variable grid dimensions
- Negotiable: Adjustable parameters include size ranges (e.g., 5x5 to 15x15 or 15x14), and UI presentation (e.g., dropdown menu vs slider)
- Valuable: Enhances the game's experience and replayability by letting players choose their board size
- Estimable: Depends on modifying the board creation logic and adding UI elements for size.
- Small: Only requires a setup menu option for changing board size and apply to the initialization of the board.
- Testable: This can be verified by starting games with different board sizes and ensuring that the basic mechanics (e.g., movement and building) work correctly.

20. As a player with friends, I want to play with 3 or 4 players so that I can enjoy the game with more friends **[Extension]**

- Independent: This feature is independent, if the basic mechanics (e.g., turn order, victory conditions) are not hardcoded for 2 players only.
- Negotiable: Things like limits of player count, workers' starting positions, can be refined during implementation
- Valuable: Allows larger groups to enjoy the game
- Estimable: Update player initialization, turn management (e.g., for 2v2, each team shares their workers), and the UI
- Small: Modify the setup to allow 3-4 players and update the turn management logic
- Testable: This can be verified by playing the game with 3/4 players, validating turn order, and ensure that all basic mechanics of the game is still correct

21. As a **player**, I want to view a history of past matches—including details such as how many players participated, which gods were chosen, how many rounds were played, who won, and by what condition—so that I can easily recall and analyze previous outcomes. **[Extension]**

- Independent: This feature does not depend on other in-game mechanics. It simply retrieves and displays past game data.
- Negotiable: The level of detail in the summary (e.g., filtering options, sorting) can be adjusted based on user preference.
- Valuable: Helps players reflect on past games.
- Estimable: Requires game to store, retrieve and display previous game data.
- Small: Very easy if it is not saved to persistent memory. Still easy if saved to persistent memory.
- Testable: Can be tested by verifying that game data is correctly saved and displayed.

22. As a **beginner player**, I want to receive a notification when a level 3 building has been built so that I know when the game may be nearing a win condition.

**[Extension]**

- Independent: This story focuses on tracking building status and triggering notifications.
- Negotiable: The notification visual and timing can be varied.
- Valuable: Provides timely strategic information, alerting players that a game is about to end if not responded to.
- Estimable: Estimated by monitoring block placements and integrating a notification system into the game flow.
- Small: Limited to checking block level immediately after each building action.
- Testable: Confirm that a notification is triggered precisely when a building reaches three blocks.

23. As a **casual player**, I want a beginner-friendly mode with no god cards so that I can learn the core mechanics of Santorini without added complexity. **[Extension]**

- Independent: Focuses on a standalone mode that excludes the god-card mechanics.
- Negotiable: What exactly qualifies as "beginner-friendly" can be discussed and modified in response to feedback.
- Valuable: Reduces obstacles for new/casual players by focusing on core movement, building, and winning rules.
- Estimable: Disabling the god cards only for that particular instance of game
- Small: Only need to disable the god-cards mechanic and add a mode toggle.
- Testable: Make sure/verify that the god mechanics are inactive and the gameplay complies with the basic Santorini Rules

24. As a **player with friends**, I want to see a summary of which players are on my team and which players are on the other, so that I can quickly understand the team composition for games with 3 or more players. (This is applicable for 3-4 players or more) **[Extension]**

- Independent: This feature is self-contained and does not rely on other features. A really simple UI addition.
- Negotiable: The specifics of how the summary is displayed (UI, colors, icons) can be adjusted, it just needs to exist and convey the information.
- Valuable: Helps players quickly remember the teams, especially in 3-4 player modes.
- Estimable: This is a UI element that draws upon existing game logic.
- Small: Can be implemented as a simple UI addition with minimal backend work. Should be a very easy edit.
- Testable: Can be tested easily by verifying visually that correct teams are displayed based on game setup



