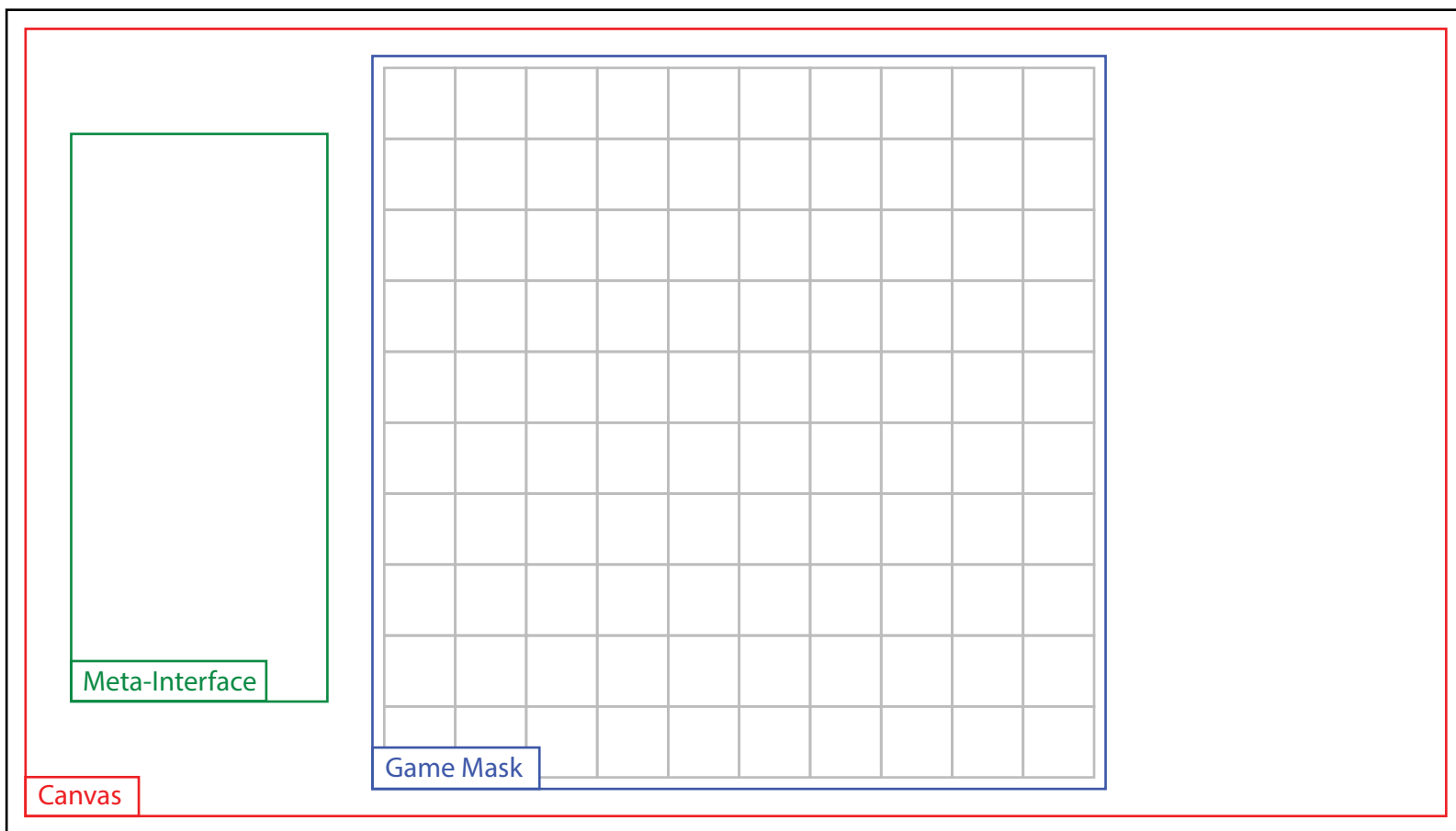


# Design Layout - Basic View



**Canvas:** An abstraction of the application window; may be dynamically resized in dimension and aspect ratio.

**Game Mask:** A rect which renders a subsection (or all) of the game space.

- This mask could change dimension based on screen type, but a scaled, square mask ensures the experience is the same on all types of monitors (and even other platforms if necessary).
- The game space could be of any size from 1 x 1 to 1,000,000 x 1,000,000 -- displaying only the area within the mask ensures render times are low and the player can effectively see and manage areas relevant to them.
- Depending on feedback or internal testing, the grid-scale can be appropriately adjusted to hit the Goldilocks-zone between viewing range and clarity of visuals (e.g., if players feel they are unable to react quickly enough to the appearance of enemies, the mask could be scaled from a 10x10 grid to a 20x20 one so that said enemies could be seen sooner).

**Meta-Interface:** A panel containing all information and controls not directly tied to the game space, including:

- Important information like how many rewards remain in the map, how many have been collected, and the duration of the play session.
- System-level controls like a pause button, exit button, etcetera.
- (Possibly) high score, leaderboards, similar social integration.

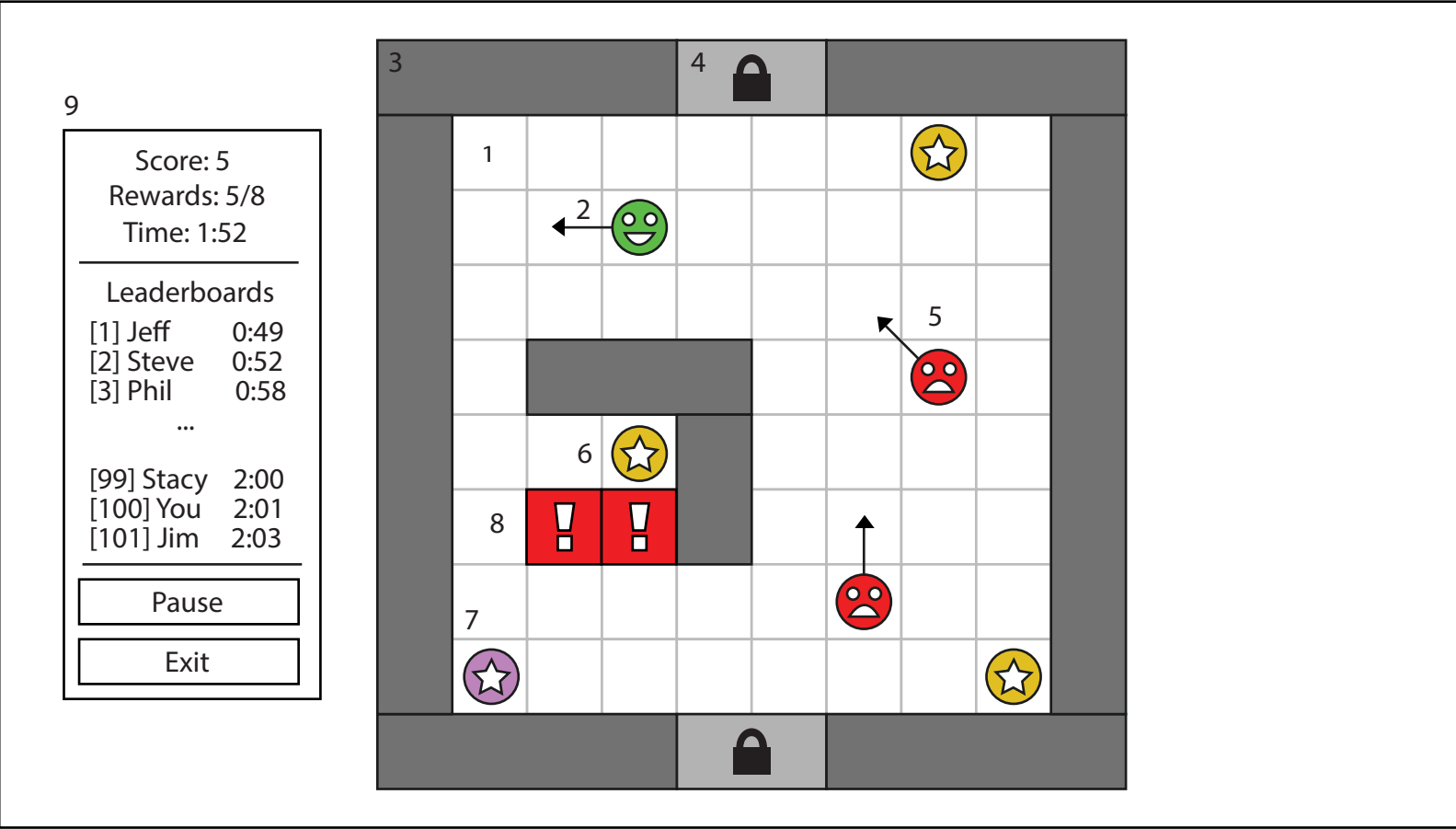
## Pros:

- Few, rectangular components make it easy to resize and rearrange.
- Device/platform independent (can adapt to different resolutions/aspect ratios).
- In-game and out-of-game data is cleanly partitioned into its own spaces.

## Cons:

- A lot of unused space in the right half (could be fixed by making the game mask stretch across the whole canvas).

# Design Layout - Example



1. Tile: Basis for Entity positions, may each contain one (1) Entity.
2. Player: May be moved by keyboard inputs once per tick (using WASD, Arrow Keys, etc.)
3. Walls: Obstacles the Player cannot pass through.
4. Exit: An Obstacle which dissapears once all Rewards are collected.
5. Enemy: Moves towards the Player every tick, and ends the game if it enters their Tile.
6. Reward: Increases Score when collected. If all are collected, the Exit opens.
7. Bonus: Occasionally appears, and increases Score by more than a normal reward. Not needed to win.
8. Punishment: Reduces Score if Player enters the same Tile; game ends if this reduces Score to zero (0).
9. Interface: Shows user data, leaderboards, and provides controls for handling game state.

## Notes:

This example uses a 10x10 grid which is fully represented by the Game Mask.

Icons are not representative of final version; simple assets used for ease of understanding.

Properties of note such as Entity starting position, Tile coordinate, etc. are omitted for clarity.