

# Gridlock Documentation

## User Stories

**ID:** 1

**Title:** Read and write to data files

**Description:** Reads in level data from file and constructs vehicle and level object. Also writes to stats file

**Responsibility:** Julian

**Acceptance Criteria:**

- Each line in file contains Vehicle information as follows:  
<orientation> <startpos> <endpos> <index in row/col array>
- Create a list of vehicle objects and store it in Level object
- Auto increment level id as you create new levels
- Each separate file corresponds to a separate level.
- A list of levels will be stored in main class
- Update row and column arrays once data is read in

**ID:** 2

**Title:** Generate Board from level

**Description:** takes in level id, creates 2D row and column arrays with vehicle objects. Load just the vehicles.

**Responsibility:** Vincent

**Acceptance Criteria:**

- When a level is clicked, the board should load that level
- Board is constructed from 2 things:
  - 2D Row array with List of vehicle objects. E.g. Row[1][Vehicle A, Vehicle B]
  - 2D Column array with List of vehicle objects e.g. Column[1][Vehicle A]
- Vehicle objects are constructed from the row and column array.
- Render board based on changes (add a listener)

**ID:** 3

**Title:** Collision management

**Description:** listens to mouse event, detect collision with other vehicles or edge of board

**Responsibility:** Mandar

**Acceptance Criteria:**

- Read mouse position
- Return true if mouse position conflicts with existing vehicle position or edge of board

**ID:** 4

**Title:** Update board (back end)

**Description:** listens to mouse event, modifies vehicle position in board array

**Responsibility:** Mandar / Aaron

**Acceptance Criteria:**

- On mouse click, find which vehicle is selected
- Read the x or y position of the mouse depending on the orientation of the selected vehicle
- On mouse release, update vehicle position in vehicle object if vehicle is not clashing with another vehicle or out of the board
- Otherwise, place the vehicle back in the last allowed position.

**ID: 5**

**Title:** View/display (front end)

**Description:** Display the level menu

**Responsibility:** Isaac & Aaron

**Acceptance Criteria:**

- When “Levels” is pressed, display a list of levels, with a scroll bar
- Completed levels have a tick on the right side
- Observer/observable class
- Display the following:
  - 6x6 board
  - Time
  - Moves
  - Restart button
  - **Exit button - goes back to main menu**

**ID: 7**

**Title:** Completed Level (front end)

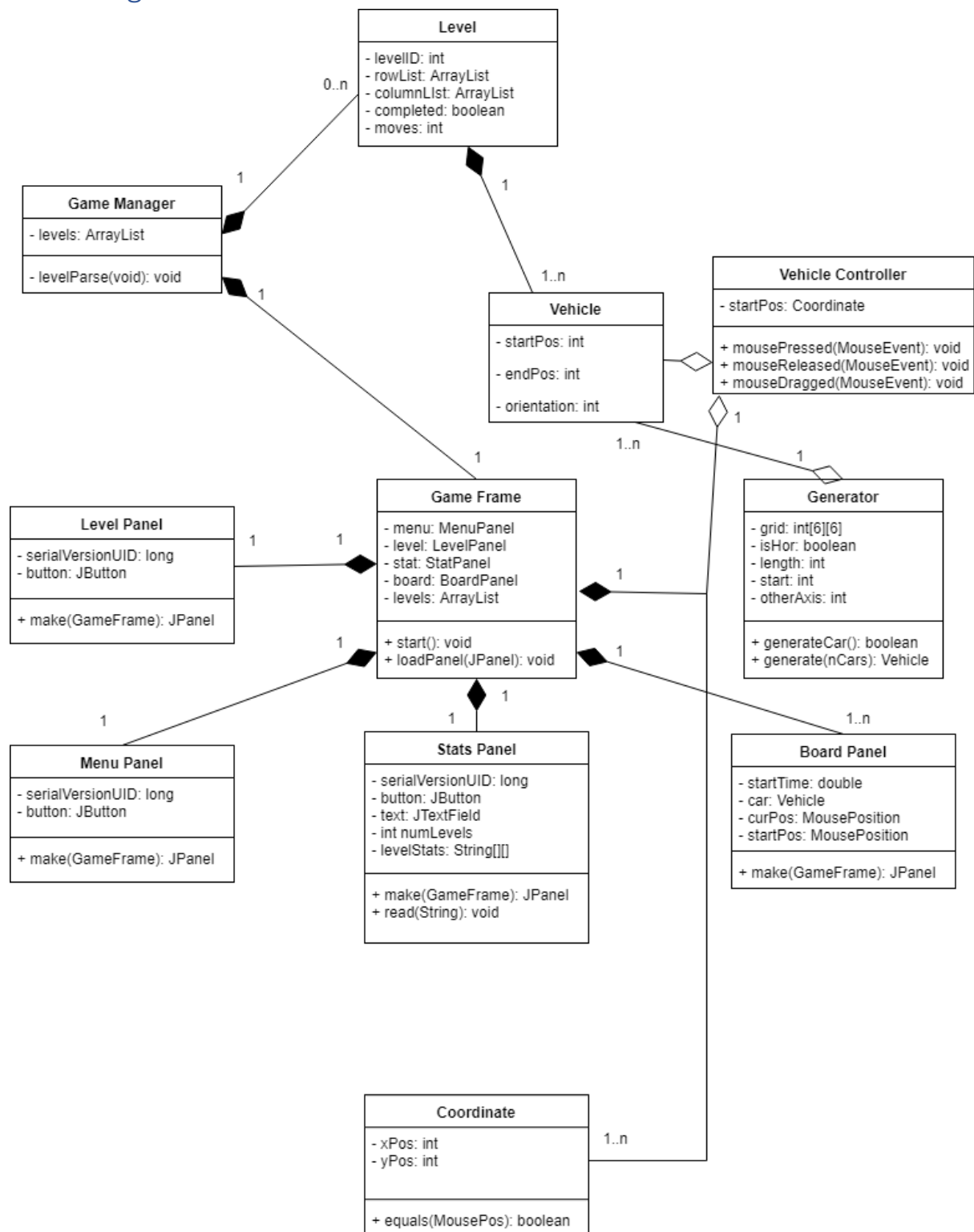
**Description:** Display the completed level screen

**Responsibility:** Isaac

**Acceptance Criteria:**

- Stop the timer
- Display moves, timer, “Complete” text
- Links to main menu, and next level
- Update stats if new record

## UML Diagram



## Sequence Diagram

