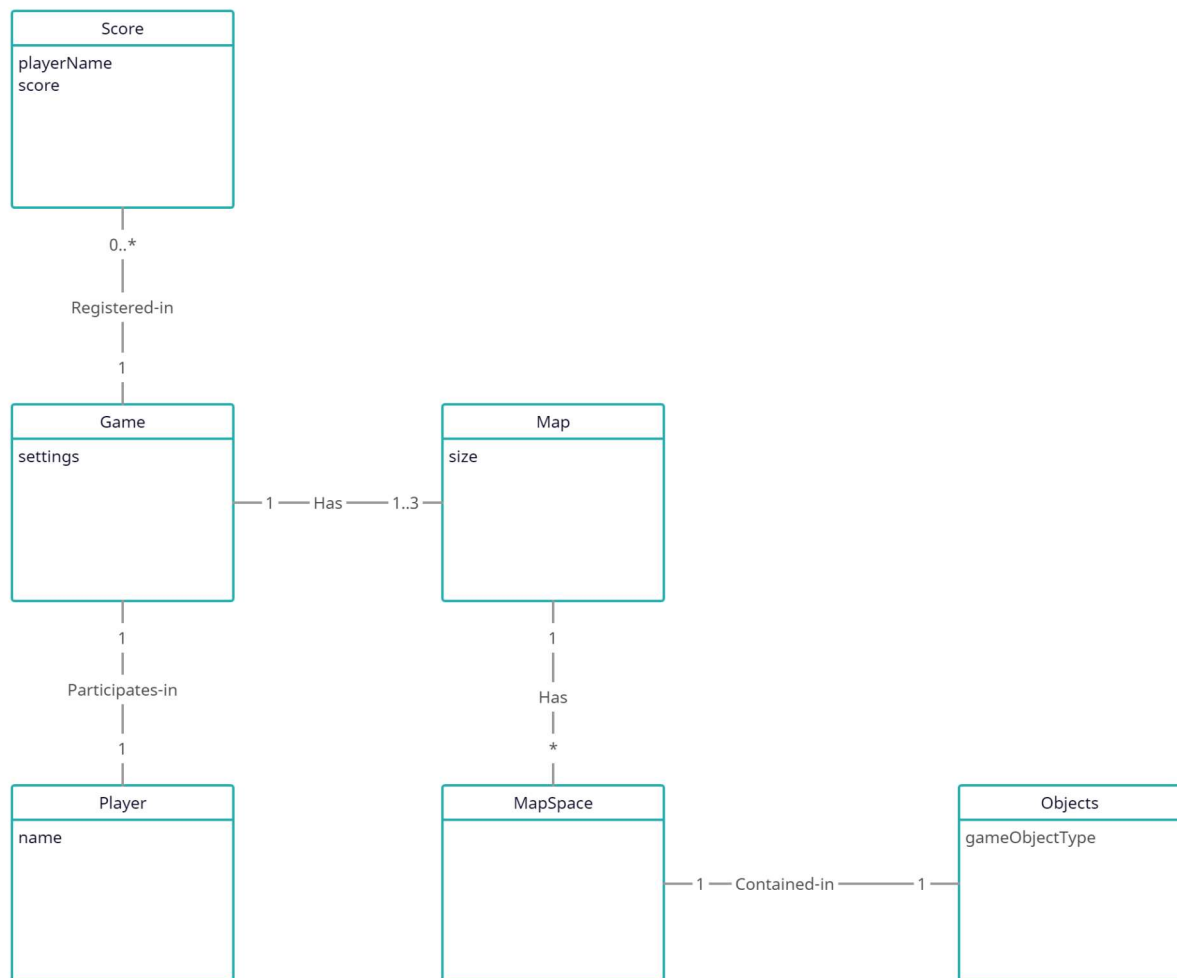# TAXI GAME

Introduction:

The goal of the project was to create a text-based taxi game in which a taxi driver can practice routes so that he/she may become a better taxi driver.

The game is run on an X by Y fields map and each field has an object that contains a text description. The game will automatically save any score (number of moves) after reaching the goal.

These concepts are drawn in the following domain model:

Controls and features:

The player has to type in their name before the game can start.

Before "starting" a game, the player has several options:

```
Welcome to the taxi game Jimmy: please choose an option from the menu
1 - Start game - Beginner
2 - Change map
3 - Preview maps
4 - High-scores / Statistics
5 - Settings
6 - Exit program
```

These options (as well as other important features) will be detailed in the following use cases:

Start Game	Default map will be beginner which is a 20 by 20 fields map. The map can be changed in menu option 2. The player and goal are placed on individual random positions on the map. The player then controls the taxi driver with text-inputs such as "N", "S", "E", "W" and combinations of those letters for diagonal movement. The player will continue typing inputs until they either reach the goal or crash into a tree/lake.
Change Map	Allows the player to change the played map.
Preview Map	Allows the player to preview an example of the map, with random positions for the player and goal.
See High Scores	Displays the top 3 scorers of each map, ranked by least number of moves to reach the goal.
Change Settings	Allows the player to toggle settings such as: Display player position Display goal position And the Safe Crash option, which gives a warning to the player if they are about to crash. All settings are on by default.

Move Player	In order to move the player, the game needs to check whether the desired position is available, in which it checks what object is in the destination.
Show Route	Display the route taken by the player during a game. It will show start position as well as the goal, unless the player crashes.

A full overview of the project can be seen on the following class diagram:
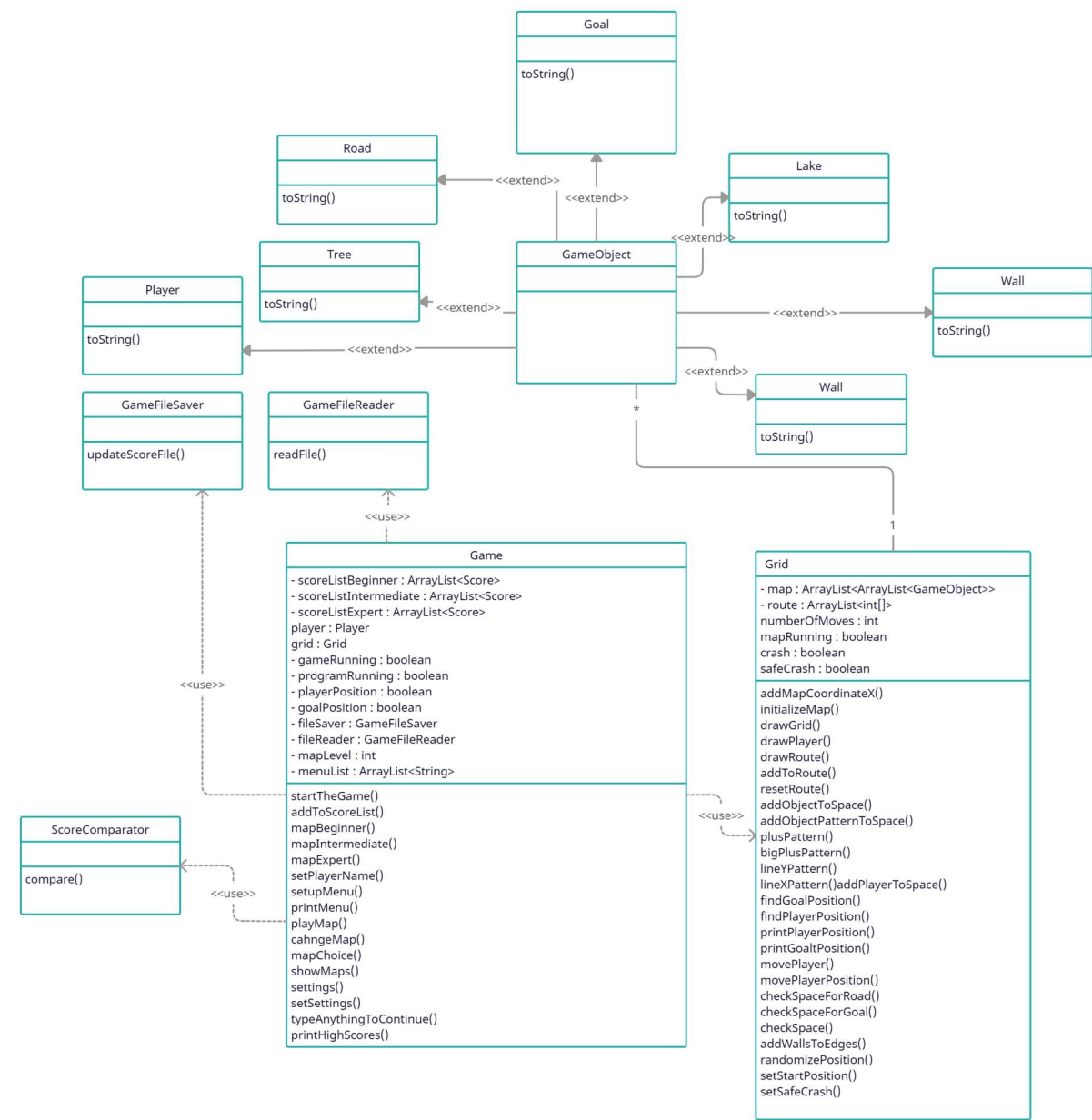
Reflection on future project features:

It was originally planned to have procedurally generated maps but given the time frame, the feature was dropped and three standard maps were designed by hand instead.

Another feature that was in the works was a Fire object that could spread on the map every X moves the player made. However due to time constraints this feature was also discarded.

The last and most useful (for the taxi driver) feature was a method to calculate the best route given the random start and goal positions, and using this method, the game would be able to calculate a "handicap" similar to golf wherein each hole (map) has an amount of strokes to reach "par". Again, this feature was regretfully not accomplished in time, but is a great contender for a future update.

Conclusion:

The core gameplay features have been added, such as destination-dependent movement (checks for obstacles) as well as the possibility to add new maps easily with a combination of the methods initializeMap() and addWallsToEdges() which when given a map size will automatically fill-out an ArrayList<ArrayList<GameObject>> with Road objects. In conclusion the game has completed the given requirements and has great potential to add new features in the future.