**Maze Design Document:**

1. There will be a new Match manager (**Match** class) that get 3 arguments (**mazes\_folder, players\_package, num\_threads**)

* Validation done for those args. The **num\_threads** is optional = default is 1
* According to **players\_package** argument, Match load the Player classes, and create instance for each game execution.
* According to **mazes\_folder** argument, all maze files loaded. only those passed validation, will be take in account of game execution.
* According to **num\_threads**, Match decide if execute the game tasks in threads pool, or open single thread, and send to it the game task when the thread available.

1. There will be a new Game manager (GameManagerTask) that inherit from the game manager implementation (**GameManagerImpl**).

The difference between the game managers are:

* **GameManageImpl** report to output the result for each move.
* **GameManageTask** implements runnable
* **GameManageTask** only collects the steps number for reporting.

1. There will be a new Output class ‘**MultipleGameOutputResult’** that will get a Map<MazeData,List<GameManager>>  the map will print to the console the game results.
2. Many issues were fixed form the previous assignment.
3. **Player** - player with bookmark algorithm improved and also the random player improved.