**Entity:** Cell

Attributes:

* abs\_x (int): the abscissa position of the box
* ord\_y (int) : the ordinate position of the case
* type\_case (int) : indicates if the case is a corridor or a wall
* est\_entree (bool): indicates if the cell is the entrance of the labyrinth or not

Methods:

* get\_neighbours () : List[Case()] -> returns the neighbouring cases accessible from this case
* get\_walls () : List[Case()] -> returns the neighbouring cases that are walls
* contain\_monster () : bool -> indicates if the cell contains a monster or not
* contain\_object () : bool -> indicates if the box contains an object or not
* contain\_aryan\_wire () : bool -> indicates if the cell is crossed by the Ariane's wire

**Entity:** Maze

Attributes:

* width (int) : the width of the maze
* height (int): the height of the labyrinth
* grid (List[List[Case()]]): the grid of cases forming the labyrinth

Methods:

* generate () : void -> generate the maze
* display () : void -> display the maze
* verify () : bool -> verify if the maze respect the conditions
* find\_enter () : Case() -> returns the case corresponding to the entry (which is also the exit) of the maze
* list\_objects () : List[Case()] -> returns the cases containing objects
* list\_monsters () : List[Case()] -> returns the cases containing monsters

**Entity:** Hero

Attributes:

* current\_position (Case): the current position of the hero in the maze
* current\_direction (int) : the current direction of the hero (north, south, east, west)
* list\_objects (List[Object]) : indicates if the hero has a sword or not
* speed (int): represents the hero's speed of travel.

Methods:

* move (int) : void -> move the hero in the given direction
* pick\_object () : Object() -> picks up the treasure on the current square
* attack () : Case() -> attack the monster on the current square with the sword
* use\_aryan\_thread () : void -> use the Ariadne's thread to go back to the entrance of the labyrinth
* change\_speed (int) : void -> change the speed of the hero according to the given value

**Entity:** Object

Attributes:

* current\_position (Case): the current position of the object in the maze
* type\_object (int) : represents the type of the object (bonus, treasure, sword).

**Entity:** Monster

Attributes:

* current\_position (Case): the current position of the monster in the maze

Methods:

* attack\_heros () : void -> the monster attacks the hero and the game ends