Test_Maze - MODEL

Item	ID	Comment
__init__	1001	Tests constructor of maze
__init__	1002	Tests constructor player if equal to “P”
maze	2001	Tests whether maze setter can change mazes
check_vacancy	3001	Test If location is not vacant on map, return False
check_vacancy	3002	If location is vacant on map, return True
find_random_spot	4001	If random spots are empty, return True
create_random_items	5001	Test to see if random items are generated
find_player_idx	6001	Tests to find player index is on map
find_player_idx	6002	Tests to see if player index can be found, raises Runtime Error when cannot.
check_item	7001	Test if a thing collected in the backpack is a gift item
is_exit	8001	Test if player reach exit door, return True
is_exit	8002	Test if player does not reach exit door, return False

Test_Player - MODEL

Item	ID	Comment
__init__	1001	Tests constructor name attribute
__init__	1002	Tests constructor backpack attribute
__init__	1003	Tests if constructor backpack is a list
pickup_item	2001	Tests if items picked up gets appended to backpack

Test Welcome View

Item	ID	Comment
display_welcome	1001	Tests if welcome display maze structure correctly

Test Game View

Item	ID	Comment
display_maze_game	1001	Tests if the game display maze structure correctly

Test Game View

Item	ID	Comment
__init__	1001	Test if the construct of the backpack is valid
display_maze_game	2001	Tests if the endgame display maze structure correctly

Test Maze View

Item	ID	Comment
display_maze	1001	Tests if the maze structure display correctly

Test Controller

Note that the welcome_controller and endgame_controller are just to call from welcome_view and endgame_view. Since we already test Views so, we believe we can skip them.

Test Game Controller

Item	ID	Comment
check_user_input	1001	Tests if user input is receive correctly

Note that run() method in game controller has no argument do we cannot test.