




由 Amanda Lu 最后更新于 3月 04, 2023 •  48 人已查看

Entity	JSON Prefix	Creatable from Dungeon Map?
Player	player	Yes
Wall	wall	Yes

Exit	exit	Yes
Boulder	boulder	Yes
Floor Switch	switch	Yes
Door	door	Yes
Portal	portal	Yes
Zombie Toast Spawner	zombie_toast_spawner	Yes
Spider	spider	Yes
Zombie Toast	zombie_toast	Yes
Mercenary	mercenary	Yes
Treasure	treasure	Yes
Key	key	Yes
Invincibility Potion	invincibility_potion	Yes
Invisibility Potion	invisibility_potion	Yes
Wood	wood	Yes
Arrows	arrow	Yes
Bomb	bomb	Yes
Sword	sword	Yes
Bow	bow	No, since this entity must be built by the player.
Shield	shield	No, since this entity must be built by the player.

1.1.1 Further Entities (Evolved)

Entity	JSON Prefix	Creatable from Dungeon Map?
Assassin	assassin	Yes
Hydra	hydra	Yes

Swamp Tile	swamp_tile	Yes
Sun Stone	sun_stone	Yes
Sceptre	sceptre	No, since this entity must be built by the player.
Midnight Armour	midnight_armour	No, since this entity must be built by the player.
Time Turner	time_turner	Yes
Time Travelling Portal	time_travelling_portal	Yes
Light Bulb (off)	light_bulb_off	Yes
Light Bulb (on)	light_bulb_on	No, since light bulbs will always be created off.
Wire	wire	Yes
Switch Door	switch_door	Yes
Older Player	older_player	No, since these will only appear when the player has time travelled.

1.1.2 Extra Fields (MVP)

Some entities will contain additional fields in their JSON entry, namely:

- All entities of type `portal` will have a field `colour`. Two portals which have the same `colour` are linked (travelling through one portal takes you to the other). We will never provide a dungeon which has more than two portals of the same colour, and all portals will have a counterpart of the same colour in the dungeon.
- All entities of type `door` and `key` will have a `key` field which, in the case of the `key` is the identifier of the key, and in the case of the `door` the id of the key which fits that door.

1.1.3 Extra Fields (Evolved)

- All logical entities will be created with the field `logic` which will be one of `and`, `or`, `xor`, or `co_and`. Note that light bulbs and switch doors will always be created with a logic field. Regular doors will never be created with a logic field, nor will floor switches or wires.
- Bombs may be created with the field `logic`. If they have this field, they are expected to be able to interact with other logical entities as described in Task 2g). Bombs created without this field function as they do in the MVP and do not need to interact with other logical entities.
- All swamp tiles will be created with a field `movement_factor` which will be the tile's movement factor, an integer ≥ 0 .

1.2 Input - Goals (MVP)

A basic goal is represented in the dungeon as:

```
1  "goal-condition": {
2    "goal": "goal"
```

```
2     goal : <goal>
3 }
```

Where <goal> is one of "boulders", "treasure" or "exit".

A complex goal is represented in the dungeon as:

```
1 "goal-condition": {
2   "goal": <supergoal>,
3   "subgoals": [
4     {"goal": <goal>},
5     {"goal": <goal>}
6   ]
7 }
```

Where <goal> is one of, "boulders", "treasure", "exit", or another nested goal conjunction/disjunction itself, and <supergoal> is one of "AND" or "OR".

1.3 Input - Goals (Evolved)

In Task 2 a), "enemies" is also included in this list of goals.

2. Configuration Files

In config_template.json we have specified the template for a configuration file. This file is important as it specifies internal game mechanics which will affect the external behaviour of your application. Rather than hard coding these constants into your classes, you must instead read in these values from the specified file when the game is created.

During automarking, we will be providing our own configuration files with each test dungeon - this allows our tests to set parameters that should ensure behaviours are produced without ambiguity. For this reason, if you do not read in the values correctly, you will likely fail a large number of our autotests.

2.1 Configuration Fields (MVP)

JSON Format	Description
ally_attack	Attack bonus each ally gives to the player.
ally_defence	Reduction in effect of enemy attack damage each ally gives to the player.
bribe_radius	Radius in which a mercenary can be bribed.
bribe_amount	Amount of gold required to bribe a mercenary.
bomb_radius	Blast radius of bomb.
bow_durability	The number of battles that the bow lasts for.

player_health	Health of the character.
player_attack	Attack damage of the character.
enemy_goal	At least x enemies must be killed to complete the enemy goal
invincibility_potion_duration	The effects of the potion only last for x ticks.
invisibility_potion_duration	The effects of the potion only last for x ticks.
mercenary_attack	Attack damage of the mercenary.
mercenary_health	Health of the mercenary.
spider_attack	Attack damage of the spider.
spider_health	Health of the spider.
spider_spawn_interval	Spiders spawn every x ticks, starting from the x'th tick. Spawn rate of 0 means that spiders will never spawn in the game.
shield_durability	The number of battles that the shield lasts for.
shield_defence	The reduction in the effect of the attack damage of the enemy as a result of the shield.
sword_attack	Amount of damage added to a players' attack damage when they use a sword in battle.
sword_durability	The number of battles that the sword lasts for.
treasure_goal	At least x treasure must be collected to complete the treasure goal
zombie_attack	Attack damage of the zombie toast.
zombie_health	Health of the zombie toast.
zombie_spawn_interval	Zombies spawn every x ticks from each spawner, starting from the x'th tick. Spawn rate of 0 means that zombies will never spawn in the game.

2.2 Configuration Fields (Evolved)

JSON Format	Description
assassin_attack	Attack damage of the assassin.

assassin_bribe_amount	The amount of gold required to perform an attempt to bribe an assassin.
assassin_bribe_fail_rate	The chance that the bribe on an assassin will fail. The value of this field should be always inclusively between 0 and 1.
assassin_health	Health of the assassin.
hydra_attack	Attack damage of the hydra.
hydra_health	Health of the hydra.
hydra_health_increase_rate	The chance that the health of a Hydra increases when it gets attacked each round. The value of this field should be always inclusively between 0 and 1.
hydra_health_increase_amount	The increment on the health of a Hydra increases when it gets attacked.
mind_control_duration	The amount of time mind controlling via a sceptre lasts for.
midnight_armour_attack	Attack bonus wearing midnight armour gives to the player.
midnight_armour_defence	Defence bonus wearing midnight armour gives to the player.

 A note about **backwards compatibility**:

- **All** the regression tests we have provided to you in the starter code should remain passing
- All of the MVP configuration files (in the provided config files) do not currently contain the fields listed in Section 2.1. Rather than retroactively adding these fields to the existing configuration files, you will need to design your implementation to accomodate for this and maintain backwards compatibility. **All Version 2 configuration files in our autotests** will contain all values from both Version 2 (Evolved) and Version 1 (MVP).

3. Interface

The layer of abstraction is at the level of the controller. In the starter code, we have provided a class `DungeonManiaController`.

The controller methods interact with a HTTP layer in the form of a web server, which we have written for you.

3.1 Interface Data Types

We have provided the following interface data types for you inside `response/models`. Similarly as for the assignment, you will need to create objects of these types for the controller to return and communicate information to the server layer.

In case you are interested, The server layer then wraps these objects inside a `GenericResponseWrapper`, a generic type we have made for you, and converts these objects to JSON using a library called `gson` to allow them to be communicated to the frontend via a HTTP response.

Constructor Prototype	Description
<pre>1 public DungeonResponse(String dungeonId, 2 String dungeonName, 3 List<EntityResponse> entities, 4 List<ItemResponse> inventory, 5 List<BattleResponse> battles, 6 List<String> buildables, 7 String goals)</pre>	<ul style="list-style-type: none">dungeonId is the unique identifier for the dungeondungeonName is the name of the dungeon map being used (i.e. maze , which corresponds to the file src/main/resources/dungeons/maze.jsonentities is a list of all entities currently in the dungeon (all entities in the Player's inventory aren't included); if a Player or enemy dies it is removed from this listinventory is the Player's current inventorybuildables is a list of buildable item types that the player can build, given their current inventory and game statebattles is the list of battles that has occured in total in the game so far (see BattleResponse), in the order that they occurredgoals is a string containing the goals yet to be completed. An empty string denotes the game has been won. Each goal in the string is preceded with a colon : and is one of the three basic goals listed in Section 2.7 or the fourth goal you will implement in task 2a). How you represent conjunctions (AND) and disjunctuions (OR) is up to you, as the frontend will simply render your string with the goals as images. We will only check for the goal strings in our tests (e.g. :exit). An example of the goals string is ":exit AND (:treasure OR :enemies)"
<pre>1 public BattleResponse(String enemy, 2 List<RoundResponse> rounds, 3 double initialPlayerHealth, 4 double initialEnemyHealth, 5 List<ItemResponse> weaponryUsed)</pre>	<ul style="list-style-type: none">enemy is the type of enemy (e.g. spider), rounds represent the rounds of the battle (see RoundResponse).initialPlayerHealth is the initial health of the player before the battleinitialEnemyHealth is the initial health of the enemy before the battle.weaponryUsed is a list of all attack and defence items used in the battle, including potions.
<pre>1 public RoundResponse(double deltaPlayerHealth, 2 double deltaEnemyHealth)</pre>	<ul style="list-style-type: none">deltaPlayerHealth is the change in health of the character in that round of the battle (e.g. -3 is a reduction of 3 in health)deltaEnemyHealth is the corresponding change of health of the enemy in that round of the battle. Note that each of these deltas can be positive and that the 'sign' of the health matters (e.g. positive deltas correlate to increase and negative deltas correlated to decrease in health).
<pre>1 public EntityResponse(String id, 2 String type, 3 Position position, 4 boolean isInteractable)</pre>	<ul style="list-style-type: none">id is the unique identifier for the respective entitytype is the type of the entity (a prefix corresponding to the table in Section 4.1)position is the x, y, z (layer) position of the entityisInteractable refers to if the entity can receive interaction updates from frontend, which only pertains to mercenaries and zombie toast spawners. When mercenaries become allies, they are no longer interactable.

1 public ItemResponse(String id, String type)	<ul style="list-style-type: none"><code>id</code> is the unique identifier for the item and <code>type</code> is the type of item (lowercase, see Section 3 for names).
1 public Position(int x, int y, int layer)	<ul style="list-style-type: none"><code>x</code>, <code>y</code> are the co-ordinates of the cell (the top-left cell is 0,0)<code>layer</code> is the Z-position of the entity on the screen (a higher layer is "in front" of a lower layer visually). The Z-position only matters for frontend rendering and is not something we will test.
1 public enum Direction { 2 UP(0, -1), 3 DOWN(0, 1), 4 LEFT(-1, 0), 5 RIGHT(1, 0); 6 }	Direction of movement for the player.

3.2 Interface Methods (MVP)

Method Prototype	Description	Exceptions
1 public DungeonResponse newGame(String dungeonName, 2 String configName) 3 throws IllegalArgumentException	Creates a new game, where <code>dungeonName</code> is the name of the dungeon map (corresponding to a JSON file stored in the model) and <code>configName</code> is the name of the configuration file.	IllegalArgumentException: <ul style="list-style-type: none">If <code>dungeonName</code> is not a dungeon that existsIf <code>configName</code> is not a configuration that exists
1 public DungeonResponse getDungeonResponseModel()	Return the dungeon response for the current state of the game without any side effects on the game.	N/A
1 public DungeonResponse tick(String itemUsedId) throws 2 InvalidActionException	Ticks the game state when the player uses/attempts to use an item. The player's action (attempts/using an item) must be carried out first, <i>then</i> enemy movement. As soon as the item is used, it is removed from the inventory.	IllegalArgumentException: <ul style="list-style-type: none">If <code>itemUsed</code> is not a bomb, invincibility_potion, or an invisibility_potion InvalidActionException: <ul style="list-style-type: none">If <code>itemUsed</code> is not in the player's inventory
1 public DungeonResponse tick(Direction movementDirection) <div><div></div></div>	Ticks the game state when the player moves in the specified direction one square. The player's movement must be carried out first, <i>then</i> enemy movement.	N/A
1 public DungeonResponse build(String buildable) 2 throws InvalidActionException	Builds the given entity, where <code>buildable</code> is one of bow, shield, (Evolved) sceptre, or midnight_armour.	IllegalArgumentException: <ul style="list-style-type: none">If <code>buildable</code> is not one of bow, shield, (Evolved) sceptre,

		<p>or <code>midnight_armour</code></p> <p><code>InvalidActionException</code>:</p> <ul style="list-style-type: none">If the player does not have sufficient items to craft the buildable, 🧟 or unbuildable for <code>midnight_armour</code> because there are zombies currently in the dungeon.
<pre>1 public DungeonResponse interact(String entityId) 2 throws IllegalArgumentException</pre>	<p>Interacts with a mercenary (where the Player bribes/mind controls [Evolved] the mercenary) or a zombie spawner, where the Player destroys the spawner.</p>	<p><code>IllegalArgumentException</code>:</p> <ul style="list-style-type: none">If <code>entityId</code> is not a valid entity ID <p><code>InvalidActionException</code></p> <ul style="list-style-type: none">If the player is not within specified bribing radius to the mercenary, when they are bribingIf the player is not cardinally adjacent to the spawner, if they are destroying a spawner(Evolved) If the player does not have enough gold and does not have a sceptre and attempts to bribe/mind-control a mercenaryIf the player does not have a weapon and attempts to destroy a spawner

3.3 Interface Methods (Evolved)

3.3.1 Time Travel

Method Prototype	Description	Exceptions
<pre>1 public DungeonResponse rewind(int ticks)</pre>	<p>Rewinds the game state a specified number of ticks.</p>	<ul style="list-style-type: none"><code>IllegalArgumentException</code>:If <code>ticks</code> is <code><= 0</code>;If the number of <code>ticks</code> have not occurred yet;

3.3.2 Dungeon Generation

Method Prototype	Description	Exceptions
<pre>1 public DungeonResponse generateDungeon(int xStart, int y 2 int xEnd, int yEnd, String configName)</pre>	<p>Generates a dungeon surrounded by walls in a rectangular grid from the start to the end position on the map, where the start position is the top left corner and the exit is the bottom right</p>	<p><code>IllegalArgumentException</code>: If <code>configName</code> is not a configuration that exists</p>



position is the top left corner and the exit is the bottom right corner. An exit will need to be at `(xEnd, yEnd)` and the player's starting position will be at `(xStart, yStart)`. A surrounding wall needs to encompass the dungeon which goes from `(xStart - 1, yStart - 1)` to `(xEnd + 1, yEnd + 1)`.

3.4 Server Layer

If you are completing Time Travel and Dungeon Generation, you will need to add code to `App.java` which sets up a HTTP endpoint that receives a web request from the frontend to call this API method. The request will be of the following format:

Route Name	HTTP Method	Data Types
1 <code>/api/game/rewind/</code>	POST	Parameters: { ticks: int } Return Type: { DungeonResponse }
1 <code>/api/game/new/generate/</code>	POST	Parameters: { xStart: int, yStart: int, xEnd: int, yEnd: int, configName: String } Return Type: { DungeonResponse }

We have handled potential concurrency issues by synchronising all endpoints - you will simply need to need to wrap your function call in the endpoint you create using `callUsingSessionAndArgument`. The existing endpoints are a good place to start when writing this code.

3.5 Interface Exceptions

The only two exceptions throwable by the Controller are:


- `IllegalArgumentException` (an builtin unchecked exception) on the specified conditions; and
- `InvalidActionException` (a custom-defined checked exception inside `src/main/java/dungeonmania/exceptions`).

You can throw them in any order you like, we will not test any input that would fit multiple exceptions at the same time.

3.6 Other Interface Files

File	Path	Description	Should you need to modify this?
<code>DungeonManiaController.java</code>	<code>src/main/java/dungeonmania/DungeonManiaController.java</code>	Contains one method for each command you need to implement.	Yes.
<code>App.java</code>	<code>src/main/java/App.java</code>	Runs a server for Dungeon Mania.	Only if completing Time Travel/Dungeon

			Generation.
Position.java, Direction.java, and FileLoader.java	src/main/java/dungeonmania/util/Position.java, src/main/java/dungeonmania/util/FileLoader.java, and src/main/java/dungeonmania/util/Direction.java	See Section 3.1.	No - do not modify these as we will rely on them being the same in automarking.
DungeonResponse.java, EntityResponse.java, GenericResponseWrapper.java, and ItemResponse.java	src/main/java/dungeonmania/response/models/	See Section 3.1.	No.
Scintilla.java and auxiliary files; Environment.java, PlatformUtils.java, and WebServer.java	src/main/java/scintilla	Contains a small custom built wrapper around Spark-Java for running a web server. When run it automatically opens a web browser.	No.
InvalidActionException.java	src/main/java/dungeonmania/exceptions	A checked exception to throw when an invalid action is attempted (See Section 3.5).	No - do not modify this class as we will be relying on it during automarking.

 成为第一个添加回复的用户