

# ASP8 TOWER DEFENCE DOCUMENTATION

ASP8 Team members:

- Alex Chu
- Dimitri Vlachos
- Freda Xiaoyun Yu
- Jeremy Matthews
- Sharif Khan

UNIVERSITY OF LONDON CM2020

## Table of Contents

audioManager .....	2
audioManager.mjs .....	2
Parameters.....	2
Functions.....	2
utilities .....	3
checkJSON.js .....	3
Functions.....	3
gameRecords.js .....	3
Parameters.....	3
Functions.....	3
resources.js .....	4

# audioManager

## audioManager.mjs

The audio manager is used to manage the audio of the game. The music, SFX and so on.

### Parameters

- `_soundManager` Phaser's default soundManager for the audioManager to manage.
- `_sfxEvents` A list of sfx 'events' to be called by the `playEffect()` function.
- `_musicEvents` A list of music 'events' to be called by the `playMusic()` function.
- `_musicSounds` A list of music 'sounds' to be called by the `playMusic()` function and looped.
- `_volume` (default: **1**) The audible volume of all sounds.
- `_musicVolume` (default: **1**) The audible volume of all music.
- `_sfxVolume` (default: **1**) The audible volume of all sfx.
- `_currentMusic` (default: **undefined**) Music to be playing currently.

### Functions

#### `pause()`

Pauses all sounds being played by the audioManager.

#### `resume()`

Resumes all sounds being played by the audioManager.

#### `stopAll()`

Stops all sounds being played by the audioManager.

#### `playMusic(musicEvent)`

Plays the `musicEvent`, adding it to the list `_musicEvents` if it is not already in it.

#### `stopMusic()`

Stops music from playing.

#### `playEffect(sfxEvent)`

Plays the `sfxEvent`, adding it to the list `_sfxEvents` if it is not already in it.

#### `getVolume()`

Returns the value of `_volume`.

#### `setVolume(volume)`

Sets the value of `_volume`.

#### `getMusicVolume()`

Returns the value of `_musicVolume`.

#### `setMusicVolume(musicVolume)`

Sets the value of `_musicVolume`.

#### `getSfxVolume()`

Returns the value of `_sfxVolume`.

#### `setSfxVolume(sfxVolume)`

Returns the value of `_sfxVolume`.

#### `_getInstance()`

Returns the `_instance`. Unless no instance exists, in which case it returns null.

## Map Objects

### enemy.js

## utilities

### checkJSON.js

This utility extends Phaser.io's default Phaser.Scene. It allows for the application to validate JSON files, ensuring their integrity and compatibility with the application.

#### Functions

##### checkDict(dict, keys)

This function checks that every relevant value of a given dictionary is the right data type.

It takes, as input:

- *dict*, the dictionary to check.
- *keys*, a second dictionary that contains the data types for all of the relevant values for any dictionary used in the game. Only those covered by the key dictionary are checked.

##### checkLevel(dict, data, key)

This function is used specifically for checking the inner arrays of the level dictionaries.

It takes, as input:

- *dict*, the dictionary to add the data to.
- *data*, the dictionary containing all of the data to check.
- *key*, the key of the dictionary to check.

By default, all dictionaries pass, therefore, the data may be added to the larger level dictionary. If any of the tests fail, then it will not be added to the larger level dictionary.

### gameRecords.js

This utility extends Phaser.io's default Phaser.Scene. It allows for levels to be selected

#### Parameters

- *levelSelect* (default: **-1**) This is how the levels are selected and the selection is preserved.
- *wavesSurvived* This records how many waves the player has survived.
- *lives* This records how many lives the player has remaining.
- *score* This records the highest score the player has attained during the current game session.
- *topScore* (default: **0**) This records the highest score the player has attained during the current game session.
- *instantWin* (default: **false**)

#### Functions

##### updateTopScore()

this function updates the top score by comparing it to the current score. If the current score is higher, then the top score is changed to the value of the current score.

`resources.js`

This utility pre-loads all of the necessary assets for the game.