Python Combat

Generated by Doxygen 1.8.17

1 Python-Combat **1**
1.1 Instructions . . . 1
1.2 Theme . . . 1
1.3 Levels . . . 1
1.3.1 Level 1 - Moving Across Time . . . 1
1.3.2 Level 2 - Key To Success . . . 2
1.3.3 Level 3 - A Soul, For A Soul . . . 2
1.3.4 Level 4 - Power Of Functions . . . 2
1.3.5 Level 5 - Traverse The Space . . . 2
1.4 Essential Features . . . 2
1.5 Extra Features . . . 3

2 File Index **5**
2.1 File List . . . 5

3 File Documentation **7**
3.1 levels/level1/level1.js File Reference . . . 7
3.1.1 Detailed Description . . . 8
3.1.2 Function Documentation . . . 8
3.1.2.1 check_success() . . . 8
3.1.2.2 create_arena() . . . 8
3.1.2.3 print_stat() . . . 8
3.1.2.4 speeddown() . . . 8
3.1.2.5 speedup() . . . 8
3.1.2.6 walk() . . . 9
3.1.3 Variable Documentation . . . 9
3.1.3.1 innerHTML . . . 9
3.1.3.2 log . . . 9
3.1.3.3 logger . . . 9
3.1.3.4 pos_x . . . 9
3.1.3.5 pos_y . . . 9
3.1.3.6 reloaded . . . 9
3.1.3.7 speed . . . 10
3.1.3.8 speed_dict . . . 10
3.1.3.9 speed_show . . . 10
3.1.3.10 total_gems . . . 10
3.2 levels/level2/level2.js File Reference . . . 10
3.2.1 Detailed Description . . . 11
3.2.2 Function Documentation . . . 11
3.2.2.1 check_success() . . . 11
3.2.2.2 create_arena() . . . 11
3.2.2.3 create_map() . . . 11
3.2.2.4 direction() . . . 12

3.2.2.5 print_stat() 12
3.2.2.6 speeddown() 12
3.2.2.7 speedup() 12
3.2.2.8 walk() 12
3.2.3 Variable Documentation 13
3.2.3.1 dict 13
3.2.3.2 innerHTML 13
3.2.3.3 log 13
3.2.3.4 logger 13
3.2.3.5 pos_x 13
3.2.3.6 pos_y 13
3.2.3.7 reloaded 13
3.2.3.8 speed 13
3.2.3.9 speed_dict 14
3.2.3.10 speed_show 14
3.2.3.11 total_gems 14
3.3 levels/level3/level3.js File Reference 14
3.3.1 Detailed Description 15
3.3.2 Function Documentation 15
3.3.2.1 check_success() 15
3.3.2.2 create_map() 15
3.3.2.3 print_stat() 15
3.3.2.4 snowman_kill() 16
3.3.2.5 speeddown() 16
3.3.2.6 speedup() 16
3.3.2.7 walk() 16
3.3.3 Variable Documentation 16
3.3.3.1 arena_type 16
3.3.3.2 innerHTML 16
3.3.3.3 log 16
3.3.3.4 logger 17
3.3.3.5 pos_x 17
3.3.3.6 pos_y 17
3.3.3.7 reloaded 17
3.3.3.8 snowman_killed 17
3.3.3.9 speed 17
3.3.3.10 speed_dict 17
3.3.3.11 speed_show 17
3.4 levels/level4/level4.js File Reference 17
3.4.1 Detailed Description 18
3.4.2 Function Documentation 18
3.4.2.1 check_success() 19

3.4.2.2 create_arena() 19
3.4.2.3 print_stat() 19
3.4.2.4 speeddown() 19
3.4.2.5 speedup() 19
3.4.2.6 walk() 19
3.4.3 Variable Documentation 20
3.4.3.1 arena_type 20
3.4.3.2 innerHTML 20
3.4.3.3 log 20
3.4.3.4 logger 20
3.4.3.5 pos_x 20
3.4.3.6 pos_y 20
3.4.3.7 reloaded 20
3.4.3.8 speed 20
3.4.3.9 speed_dict 21
3.4.3.10 speed_show 21
3.4.3.11 total_stones 21
3.5 levels/level5/level5.js File Reference 21
3.5.1 Detailed Description 22
3.5.2 Function Documentation 22
3.5.2.1 add_weapon() 22
3.5.2.2 check_success() 22
3.5.2.3 create_arena() 23
3.5.2.4 print_stat() 23
3.5.2.5 speeddown() 23
3.5.2.6 speedup() 23
3.5.2.7 walk() 23
3.5.2.8 weapon() 23
3.5.3 Variable Documentation 23
3.5.3.1 gem_found 23
3.5.3.2 id_order 24
3.5.3.3 index 24
3.5.3.4 innerHTML 24
3.5.3.5 insequence 24
3.5.3.6 log 24
3.5.3.7 logger 24
3.5.3.8 pos_x 24
3.5.3.9 pos_y 24
3.5.3.10 reloaded 24
3.5.3.11 speed 25
3.5.3.12 speed_dict 25
3.5.3.13 speed_show 25

3.6 README.md File Reference 25
3.7 welcome.js File Reference 25
3.7.1 Detailed Description 25
3.7.2 Function Documentation 25
3.7.2.1 typeWriter() 25
3.7.3 Variable Documentation 26
3.7.3.1 i 26
3.7.3.2 speed 26
3.7.3.3 title 26

Index **27**

Chapter 1

Python-Combat

A Web Platform that allows individuals to learn python programming in an enjoyable way, made as Course Project for CS251 - Software Systems Laboratory. The Project is inspired by `Code Combat`. More detailed description of Project can be found `here`.

The Combat is hosted at `https://prolific-pythonists.netlify.app/`

1.1 Instructions

- Click on **Start** button on the home page to start the **Combat**
- Choose the level which you want to play. There are 5 levels in the game
- You will see a tutorial about the programming concepts of **python** which are being used in the level
- Click on **Play Now** -> **Start Combat** to play that Level
- Instructions and Logger will be displayed along with the arena
- Follow the instructions and write your code in code editor provided
- You have to help our *Warrior* using your Coding Skills to achieve his goal!

1.2 Theme

The warrior has decided to repeat the history of his ancestor **Thanos**. He is on his way to collect all the Six `Infinity Stones` and become the Supreme Power in Universe. But, He is intelligent and calm. He is not going to wipe out half the Universe, instead use the stones powers for well being of it. You are his friend from Earth to help him in his Journey. Each level awaits a infinity stone on success. You along with the warrior already have the mind stone with you. So, use it to collect the rest of the 5 stones and help the warrior for achieveing his aim.

1.3 Levels

1.3.1 Level 1 - Moving Across Time

Level 1	Moving Across Time
Programming Concept	Python Variables
Infinity Stone	Time Stone
Arena	Grassy Arena

1.3.2 Level 2 - Key To Success

Level 2	Key To Success
Programming Concept	Conditional Statements
Infinity Stone	Reality Stone
Arena	Fire Arena :fire:

1.3.3 Level 3 - A Soul, For A Soul

Level 3	A Soul, For A Soul
Programming Concept	Loops in Python
Infinity Stone	Soul Stone
Arena	Icy Arena ↩ :snowflake:

1.3.4 Level 4 - Power Of Functions

Level 4	Power Of Functions
Programming Concept	Functions in Python
Infinity Stone	Power Stone
Arena	Rock Arena

1.3.5 Level 5 - Traverse The Space

Level 5	Traverse The Space
Programming Concept	List Data Structure in Python
Infinity Stone	Space Stone
Arena	Universe Arena

1.4 Essential Features

- 5 Levels with different programming concepts to battle
- **Logger** with success/failure messages and `print` statements for debugging
- **Code Editor**
  - Used `Code Mirror` Library for creating a Code editor
  - Syntax highlighting using Code Mirror Themes, modes and several Addons for different Programming languages

  - Ability to get code from uploaded file using JavaScript Function
  - Suggestions/Auto-completions using CodeMirror Hints
- **Arena**
  - Dynamic Configuration of Arena at each level
  - Each battle sets up a different placements of objects on reloading, that is it is Dynamically sets itself (**Bonus part**)
  - Each Battle fought on different arena settings with a unique Theme and underlying Programming concept
- Used **Doxygen** for Report and Documentation. For Documentation of the Project, refer to the index.html file in html folder of this Project. For viewing the documentation, please download the html.zip and latex.zip files, unzip them and open index.html in html folder.

1.5 Extra Features

- **Hint** Button at every level to assist the user with some help
- A **Tutorial** at start of every level to teach some basic part of Programming concept that is being used in the level
- A Modal Box at start of each level that explains a storyfied theme of the level to user. This is made to fascinate user through an underlying story theme so as to keep the user's interest along with Coding
- An **Animation Speed** Button in each level to control speed of the warrior's moves. Using it, the Speed can be made
  - Slow
  - Medium
  - Fast
- Used `doxygen-awesome-css` theme for more elegant and nice Documentation of the Project.

Created with :heart: by `Atishay`, `Megh` & `Avadhoot`
`Team Prolific Pythonsists`

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

welcome.js
JAVASCRIPT FOR Welcome Page animation 25
levels/level1/level1.js
JAVASCRIPT FOR LEVEL 1 7
levels/level2/level2.js
JAVASCRIPT FOR LEVEL 2 10
levels/level3/level3.js
JAVASCRIPT FOR LEVEL 3 14
levels/level4/level4.js
JAVASCRIPT FOR LEVEL 4 17
levels/level5/level5.js
JAVASCRIPT FOR LEVEL 5 21

Chapter 3

File Documentation

3.1 levels/level1/level1.js File Reference

JAVASCRIPT FOR LEVEL 1.

Functions

- function speedup ()

  This function is bind to the increase button it decreases the value of var speed so that animation time is less and hence speed increases finally.
- function speeddown ()

  This function is bind to the decrease button it increases the value of var speed so that animation time is more and hence speed decreases finally.
- function create_arena ()

  This function creates the whole arena dynamically, and restores all values of the global variable when the site is reloaded.
- function walk (r)

  This function controls the movement of warrior based on the input given.
- function check_success (gems_collected)

  The function checks if the players code is correct or not.
- function print_stat ()

  This function is used to show print statement in the logger.

Variables

- var total_gems = 0

  stores the total number of gems in the path
- var pos_x = 0

  stores the X-coordinate of warrior
- var pos_y = 0

  stores the Y-coordinate of warrior
- var reloaded = true

  check if reloading was done before or not
- var speed = 500

  For the animation speed of the player.
- speed_dict = {250 : "fast", 500 : "medium", "750" : "slow"}

  maps speeds in milliseconds to the name corresponding to them.
- const speed_show = document.getElementById("speed_")

  The element in which we have to show speed.
- speed_show innerHTML = speed_dict[speed]
- var logger = document.getElementById('log')
- console log

3.1.1 Detailed Description

JAVASCRIPT FOR LEVEL 1.

Author

Prolific Pythonists

Date

24-11-2022

3.1.2 Function Documentation

3.1.2.1 check_success()

```
function check_success (
              gems_collected )
```

The function checks if the players code is correct or not.

If it is correct then it prints the message and if it is not, it will reload page after 5 second.

Parameters

{int}	gems_collected number of gems collected and assigned by player.

3.1.2.2 create_arena()

```
function create_arena ( )
```

This function creates the whole arena dynamically, and restores all values of the global variable when the site is reloaded.

3.1.2.3 print_stat()

```
function print_stat ( )
```

This function is used to show print statement in the logger.

This function is implemented by overloading console.log function to show print statements on logger as well.

3.1.2.4 speeddown()

```
function speeddown ( )
```

This function is bind to the decrease button it increases the value of var speed so that animation time is more and hence speed decreases finally.

3.1.2.5 speedup()

```
function speedup ( )
```

This function is bind to the increase button it decreases the value of var speed so that animation time is less and hence speed increases finally.

3.1.2.6 walk()

```
function walk (
               r )
```

This function controls the movement of warrior based on the input given.

Exception handling is also done in this function as while moving if warrior steps out of arena alert box is shown

Parameters

{int}	r integer determining direction of traversal. 1 => move_up, 2=>move_down() 3 => move_right() 4=> move_left

3.1.3 Variable Documentation

3.1.3.1 innerHTML

```
speed_show innerHTML = speed_dict[speed]
```

3.1.3.2 log

```
console log
```

Initial value:

```
= function (message) {
        if(String(message).includes("level1/level1.html#__main__")){
            logger.innerHTML += "error" + '<br />';
        }
        else if (typeof message == 'object') {
            logger.innerHTML += (JSON && JSON.stringify ?  JSON.stringify(message) :  message) + '<br />';
        }
        else{
            logger.innerHTML += message + '<br />'
        }
    }
})()
```

3.1.3.3 logger

```
var logger = document.getElementById('log')
```

3.1.3.4 pos_x

```
var pos_x = 0
```

stores the X-coordinate of warrior

3.1.3.5 pos_y

```
var pos_y = 0
```

stores the Y-coordinate of warrior

3.1.3.6 reloaded

```
var reloaded = true
```

check if reloading was done before or not

3.1.3.7 speed

```
var speed = 500
```
For the animation speed of the player.

3.1.3.8 speed_dict

```
speed_dict = {250 :  "fast", 500 :  "medium", "750" :  "slow"}
```
maps speeds in milliseconds to the name corresponding to them.

3.1.3.9 speed_show

```
const speed_show = document.getElementById("speed_")
```
The element in which we have to show speed.

3.1.3.10 total_gems

```
var total_gems = 0
```
stores the total number of gems in the path

3.2 levels/level2/level2.js File Reference

JAVASCRIPT FOR LEVEL 2.

Functions

- function speedup ()

  This function is bind to the increase button it decreases the value of var speed so that animation time is less and hence speed increases finally.
- function speeddown ()

  This function is bind to the decrease button it increases the value of var speed so that animation time is more and hence speed decreases finally.
- function create_arena ()

  The function creates the arena using recursion of create_map.
- function create_map (x, y, n, val)

  This function first creates a horizontal path and two vertical paths one above and one below.Assigns the values to the stones and then recalls itself.
- function walk (r)

  This function controls the movement of warrior based on the input given.
- function direction ()

  At each junction the value of each stone will be returned.
- function check_success ()

  The function checks if the players code is correct or not.
- function print_stat ()

  This function is used to show print statement in the logger.

Variables

- var total_gems = 0

  stores the total number of gems in the path
- var pos_x = 0

  stores the X-coordinate of warrior
- var pos_y = 14

stores the Y-coordinate of warrior

- var reloaded = true

  check if reloading was done before or not
- var speed = 500

  For the animation speed of the player.
- speed_dict = {250 : "fast", 500 : "medium", "750" : "slow"}

  maps speeds in milliseconds to the name corresponding to them.
- const speed_show = document.getElementById("speed_")

  The element in which we have to show speed.
- speed_show innerHTML = speed_dict[speed]
- window dict = {}

  stores relation between paths i.e. which path to be followed after a certain path.
- var logger = document.getElementById('log')
- console log

3.2.1 Detailed Description

JAVASCRIPT FOR LEVEL 2.

Author

Prolific Pythonists

Date

24-11-2022

3.2.2 Function Documentation

3.2.2.1 check_success()

```
function check_success ( )
```

The function checks if the players code is correct or not.

If it is correct then it prints the message and if it is not, it will reload page after 3 second.

3.2.2.2 create_arena()

```
function create_arena ( )
```

The function creates the arena using recursion of create_map.

It also restores the values of global variables.

3.2.2.3 create_map()

```
function create_map (
              x,
              y,
              n,
              val )
```

This function first creates a horizontal path and two vertical paths one above and one below.Assigns the values to the stones and then recalls itself.

Parameters

{int}	x x coordinate at which recursion should start

Parameters

{int}	y y coordinate at which recursion should start
{int}	n total paths to be added to make the path symmetric.
{int}	val the value to be assigned to each stone for direction.

Returns

3.2.2.4 direction()

```
function direction ( )
```

At each junction the value of each stone will be returned.

If value is 1 then go above. If value is 2 then go below If value is 3 then you are on wrong path. If value is -1 then there are no gems at that position.

Returns

the direction values and data type is int.

3.2.2.5 print_stat()

```
function print_stat ( )
```

This function is used to show print statement in the logger.

This function is implemented by overloading console.log function to show print statements on logger as well.

3.2.2.6 speeddown()

```
function speeddown ( )
```

This function is bind to the decrease button it increases the value of var speed so that animation time is more and hence speed decreases finally.

3.2.2.7 speedup()

```
function speedup ( )
```

This function is bind to the increase button it decreases the value of var speed so that animation time is less and hence speed increases finally.

3.2.2.8 walk()

```
function walk (
              r )
```

This function controls the movement of warrior based on the input given.

Exception handling is also done in this function as while moving if warrior steps out of arena alert box is shown

Parameters

{int}	r integer determining direction of traversal. 1 => move_up, 2=>move_down() 3 => move_right() 4=> move_left

3.2.3 Variable Documentation

3.2.3.1 dict

```
window dict = {}
```

stores relation between paths i.e. which path to be followed after a certain path.

3.2.3.2 innerHTML

```
speed_show innerHTML = speed_dict[speed]
```

3.2.3.3 log

```
console log
```

Initial value:

```
= function (message) {
        if(String(message).includes("level1/level1.html#__main__")){
            logger.innerHTML += "error" + '<br />';
        }
        else if (typeof message == 'object') {
            logger.innerHTML += (JSON && JSON.stringify ?  JSON.stringify(message) :  message) + '<br />';
        }
        else{
            logger.innerHTML += message + '<br />'
        }
    }
})()
```

3.2.3.4 logger

```
var logger = document.getElementById('log')
```

3.2.3.5 pos_x

```
var pos_x = 0
```

stores the X-coordinate of warrior

3.2.3.6 pos_y

```
var pos_y = 14
```

stores the Y-coordinate of warrior

3.2.3.7 reloaded

```
var reloaded = true
```

check if reloading was done before or not

3.2.3.8 speed

```
var speed = 500
```

For the animation speed of the player.

3.2.3.9 speed_dict

```
speed_dict = {250 :  "fast", 500 :  "medium", "750" :  "slow"}
```
maps speeds in milliseconds to the name corresponding to them.

3.2.3.10 speed_show

```
const speed_show = document.getElementById("speed_")
```
The element in which we have to show speed.

3.2.3.11 total_gems

```
var total_gems = 0
```
stores the total number of gems in the path

3.3 levels/level3/level3.js File Reference

JAVASCRIPT FOR LEVEL 3.

Functions

- function speedup ()

  This function is bind to the increase button it decreases the value of var speed so that animation time is less and hence speed increases finally.
- function speeddown ()

  This function is bind to the decrease button it increases the value of var speed so that animation time is more and hence speed decreases finally.
- function create_map ()

  This function creates the arena.
- function snowman_kill ()

  Kills the snowman and changes the value of snowman_killed to 1.
- function walk (r)

  This function controls the movement of warrior based on the input given.
- function check_success (code)

  This function is used for logger message of success or error.
- function print_stat ()

  This function is used to show print statement in the logger.

Variables

- var pos_x = 0

  stores the X-coordinate of warrior
- var pos_y = 0

  stores the Y-coordinate of warrior
- var reloaded = true

  check if reloading was done before or not
- var arena_type = 0

  check if arena was of type 1 or 2
- var speed = 500

  For the animation speed of the player.
- speed_dict = {250 : "fast", 500 : "medium", "750" : "slow"}

  maps speeds in milliseconds to the name corresponding to them.

- const speed_show = document.getElementById("speed_")

  The element in which we have to show speed.
- speed_show innerHTML = speed_dict[speed]
- var snowman_killed = 0

  The following var stores if info about status of snowman.
  0 menas alive
  1 means dead.
- var logger = document.getElementById('log')
- console log

3.3.1 Detailed Description

JAVASCRIPT FOR LEVEL 3.

Author

Prolific Pythonists

Date

24-11-2022

3.3.2 Function Documentation

3.3.2.1 check_success()

```
function check_success (
            code )
```

This function is used for logger message of success or error.

final position of warrior and number of lines is checked in this function to check if user have successfully completed task or not.

Parameters

{string}	code code written by user as a string. This is used to calculate number of lines

3.3.2.2 create_map()

```
function create_map ( )
```

This function creates the arena.
There is code for two different arena we choose a random numberbetweeen 0 and 1 if number is less than 0.5 we choose arena_type 1 else arena_type 2.
For sprite of each path we again choose a randon number and if this number is greater than 0.6 we add a diamond at that point.

3.3.2.3 print_stat()

```
function print_stat ( )
```

This function is used to show print statement in the logger.

This function is implemented by overloading console.log function to show print statements on logger as well.

3.3.2.4 snowman_kill()

```
function snowman_kill ( )
```

Kills the snowman and changes the value of snowman_killed to 1.

3.3.2.5 speeddown()

```
function speeddown ( )
```

This function is bind to the decrease button it increases the value of var speed so that animation time is more and hence speed decreases finally.

3.3.2.6 speedup()

```
function speedup ( )
```

This function is bind to the increase button it decreases the value of var speed so that animation time is less and hence speed increases finally.

3.3.2.7 walk()

```
function walk (
              r )
```

This function controls the movement of warrior based on the input given.

Exception handling is also done in this function as while moving if warrior steps out of arena alert box is shown

Parameters

{int}	r integer determining direction of traversal. 1 => move_up, 2=>move_down() 3 => move_right() 4=> move_left

3.3.3 Variable Documentation

3.3.3.1 arena_type

```
var arena_type = 0
```

check if arena was of type 1 or 2

3.3.3.2 innerHTML

```
speed_show innerHTML = speed_dict[speed]
```

3.3.3.3 log

```
console log
```

Initial value:

```
= function (message) {
        if (String(message).includes("level1/level1.html#__main__")) {
            logger.innerHTML += "error" + '<br />';
        }
        else if (typeof message == 'object') {
            logger.innerHTML += (JSON && JSON.stringify ?  JSON.stringify(message) :  message) + '<br />';
        }
        else {
            logger.innerHTML += message + '<br />'
        }
    }
})()
```

3.3.3.4 logger

```
var logger = document.getElementById('log')
```

3.3.3.5 pos_x

```
var pos_x = 0
```
stores the X-coordinate of warrior

3.3.3.6 pos_y

```
var pos_y = 0
```
stores the Y-coordinate of warrior

3.3.3.7 reloaded

```
var reloaded = true
```
check if reloading was done before or not

3.3.3.8 snowman_killed

```
var snowman_killed = 0
```
The following var stores if info about status of snowman.
0 menas alive
1 means dead.

3.3.3.9 speed

```
var speed = 500
```
For the animation speed of the player.

3.3.3.10 speed_dict

```
speed_dict = {250 :  "fast", 500 :  "medium", "750" :  "slow"}
```
maps speeds in milliseconds to the name corresponding to them.

3.3.3.11 speed_show

```
const speed_show = document.getElementById("speed_")
```
The element in which we have to show speed.

3.4 levels/level4/level4.js File Reference

JAVASCRIPT FOR LEVEL 4.

Functions

- function speedup ()

  This function is bind to the increase button it decreases the value of var speed so that animation time is less and hence speed increases finally.
- function speeddown ()

  This function is bind to the decrease button it increases the value of var speed so that animation time is more and hence speed decreases finally.
- function create_arena ()

  This function creates the arena.
- function walk (r)

  This function controls the movement of warrior based on the input given.
- function check_success (code)

  This function is used for logger message of success or error.
- function print_stat ()

  This function is used to show print statement in the logger.

Variables

- var total_stones = 0

  stores the total number of stones in the path
- var pos_x = 0

  stores the X-coordinate of warrior
- var pos_y = 0

  stores the Y-coordinate of warrior
- var reloaded = true

  check if reloading was done before or not
- var arena_type = 0

  check if arena was of type 1 or 2
- var speed = 500

  For the animation speed of the player.
- speed_dict = {250 : "fast", 500 : "medium", "750" : "slow"}

  maps speeds in milliseconds to the name corresponding to them.
- const speed_show = document.getElementById("speed_")

  The element in which we have to show speed.
- speed_show innerHTML = speed_dict[speed]
- var logger = document.getElementById('log')
- console log

3.4.1 Detailed Description

JAVASCRIPT FOR LEVEL 4.

Author

Prolific Pythonists

Date

24-11-2022

3.4.2 Function Documentation

3.4.2.1 check_success()

```
function check_success (
              code )
```

This function is used for logger message of success or error.

final position of warrior and number of lines is checked in this function to check if user have successfully completed task or not.

Parameters

{string}	code code written by user as a string. This is used to calculate number of lines

3.4.2.2 create_arena()

```
function create_arena ( )
```

This function creates the arena.
There is code for two different arena we choose a random numberbetweeen 0 and 1 if number is less than 0.5 we choose arena_type 1 else arena_type 2.
For sprite of each path we again choose a randon number and if this number is greater than 0.6 we add a diamond at that point.

3.4.2.3 print_stat()

```
function print_stat ( )
```

This function is used to show print statement in the logger.

This function is implemented by overloading console.log function to show print statements on logger as well.

3.4.2.4 speeddown()

```
function speeddown ( )
```

This function is bind to the decrease button it increases the value of var speed so that animation time is more and hence speed decreases finally.

3.4.2.5 speedup()

```
function speedup ( )
```

This function is bind to the increase button it decreases the value of var speed so that animation time is less and hence speed increases finally.

3.4.2.6 walk()

```
function walk (
              r )
```

This function controls the movement of warrior based on the input given.

Exception handling is also done in this function as while moving if warrior steps out of arena alert box is shown

Parameters

{int}	r integer determining direction of traversal. 1 => move_up, 2=>move_down() 3 => move_right() 4=> move_left

3.4.3 Variable Documentation

3.4.3.1 arena_type

```
var arena_type = 0
```

check if arena was of type 1 or 2

3.4.3.2 innerHTML

```
speed_show innerHTML = speed_dict[speed]
```

3.4.3.3 log

```
console log
```

Initial value:

```
= function (message) {
        if (String(message).includes("level1/level1.html#__main__")) {
            logger.innerHTML += "error" + '<br />';
        }
        else if (typeof message == 'object') {
            logger.innerHTML += (JSON && JSON.stringify ?  JSON.stringify(message) :  message) + '<br />';
        }
        else {
            logger.innerHTML += message + '<br />'
        }
    }
})()
```

3.4.3.4 logger

```
var logger = document.getElementById('log')
```

3.4.3.5 pos_x

```
var pos_x = 0
```

stores the X-coordinate of warrior

3.4.3.6 pos_y

```
var pos_y = 0
```

stores the Y-coordinate of warrior

3.4.3.7 reloaded

```
var reloaded = true
```

check if reloading was done before or not

3.4.3.8 speed

```
var speed = 500
```

For the animation speed of the player.

3.4.3.9 speed_dict

```
speed_dict = {250 :  "fast", 500 :  "medium", "750" :  "slow"}
```
maps speeds in milliseconds to the name corresponding to them.

3.4.3.10 speed_show

```
const speed_show = document.getElementById("speed_")
```
The element in which we have to show speed.

3.4.3.11 total_stones

```
var total_stones = 0
```
stores the total number of stones in the path

3.5 levels/level5/level5.js File Reference

JAVASCRIPT FOR LEVEL 5.

Functions

- function create_arena ()

  This function is used to restore the values of the global variables after reload.
- function speedup ()

  This function is bind to the increase button it decreases the value of var speed so that animation time is less and hence speed increases finally.
- function speeddown ()

  This function is bind to the decrease button it increases the value of var speed so that animation time is more and hence speed decreases finally.
- function weapon ()

  This function checks if the gem is collected or not in order.
- function check_success ()

  The function checks if the players code is correct or not.
- function add_weapon (px, py, n)

  This function is used to add gems on the arena using the location parameters like px, py provided from brython.
- function walk (r)

  This function controls the movement of warrior based on the input given.
- function print_stat ()

  This function is used to show print statement in the logger.

Variables

- var gem_found = [false, false, false, false, false]

  The list stores info about which gems are collected.
- var pos_x = 0

  stores the X-coordinate of warrior
- var pos_y = 0

  stores the Y-coordinate of warrior
- var reloaded = true

  check if reloading was done before or not
- var insequence = true

  it stores true if the gems are collected in sequence else false.

- var id_order = [ ]

  it stores the values as the gems are added to the arena.
- var index = 0

  stores the index upto which the gems found are checked while checking in animate part of jquery. This is required since jquery implements animation through queue.
- var speed = 500

  For the animation speed of the player.
- speed_dict = {250 : "fast", 500 : "medium", "750" : "slow"}

  maps speeds in milliseconds to the name corresponding to them.
- const speed_show = document.getElementById("speed_")

  The element in which we have to show speed.
- speed_show innerHTML = speed_dict[speed]
- var logger = document.getElementById('log')
- console log

3.5.1 Detailed Description

JAVASCRIPT FOR LEVEL 5.

Author

Prolific Pythonists

Date

24-11-2022

3.5.2 Function Documentation

3.5.2.1 add_weapon()

```
function add_weapon (
                px,
                py,
                n )
```

This function is used to add gems on the arena using the location parameters like px, py provided from brython.

n is used to indicate the nth gem added.

Parameters

{∗}	px
{∗}	py
{∗}	n

3.5.2.2 check_success()

```
function check_success ( )
```

The function checks if the players code is correct or not.

If it is correct then it prints the message and if it is not, it will reload page after 5 second.

3.5.2.3 create_arena()

```
function create_arena ( )
```

This function is used to restore the values of the global variables after reload.

3.5.2.4 print_stat()

```
function print_stat ( )
```

This function is used to show print statement in the logger.

This function is implemented by overloading console.log function to show print statements on logger as well.

3.5.2.5 speeddown()

```
function speeddown ( )
```

This function is bind to the decrease button it increases the value of var speed so that animation time is more and hence speed decreases finally.

3.5.2.6 speedup()

```
function speedup ( )
```

This function is bind to the increase button it decreases the value of var speed so that animation time is less and hence speed increases finally.

3.5.2.7 walk()

```
function walk (
              r )
```

This function controls the movement of warrior based on the input given.

Exception handling is also done in this function as while moving if warrior steps out of arena alert box is shown

Parameters

{int}	r integer determining direction of traversal. 1 => move_up, 2=>move_down() 3 => move_right() 4=> move_left

3.5.2.8 weapon()

```
function weapon ( )
```

This function checks if the gem is collected or not in order.

It sets value of insequence to be true if gems are collected in correct order.

3.5.3 Variable Documentation

3.5.3.1 gem_found

```
var gem_found = [false, false, false, false, false]
```

The list stores info about which gems are collected.

3.5.3.2 id_order

```
var id_order = [ ]
```

it stores the values as the gems are added to the arena.

3.5.3.3 index

```
var index = 0
```

stores the index upto which the gems found are checked while checking in animate part of jquery.
This is required since jquery implements animation through queue.

3.5.3.4 innerHTML

```
speed_show innerHTML = speed_dict[speed]
```

3.5.3.5 insequence

```
var insequence = true
```

it stores true if the gems are collected in sequence else false.

3.5.3.6 log

```
console log
```

Initial value:

```
= function (message) {
        if (typeof message == 'object') {
            logger.innerHTML += (JSON && JSON.stringify ?  JSON.stringify(message) :  message) + '<br />';
        }
        else{
            logger.innerHTML += message + '<br />'
        }
    }
})()
```

3.5.3.7 logger

```
var logger = document.getElementById('log')
```

3.5.3.8 pos_x

```
var pos_x = 0
```

stores the X-coordinate of warrior

3.5.3.9 pos_y

```
var pos_y = 0
```

stores the Y-coordinate of warrior

3.5.3.10 reloaded

```
var reloaded = true
```

check if reloading was done before or not

3.5.3.11 speed

```
var speed = 500
```
For the animation speed of the player.

3.5.3.12 speed_dict

```
speed_dict = {250 :  "fast", 500 :  "medium", "750" :  "slow"}
```
maps speeds in milliseconds to the name corresponding to them.

3.5.3.13 speed_show

```
const speed_show = document.getElementById("speed_")
```
The element in which we have to show speed.

3.6 README.md File Reference

3.7 welcome.js File Reference

JAVASCRIPT FOR Welcome Page animation.

Functions

- function typeWriter ()

  Return the ratio of the inline text length of the links in an element to the inline text length of the entire element.

Variables

- var title = "Python Combat"

  Title of the webpage.
- var i = 0

  Iterator for all letters in the word "Python Combat".
- var speed = 150

  Speed of the animation.

3.7.1 Detailed Description

JAVASCRIPT FOR Welcome Page animation.

Author

Prolific Pythonists

Date

24-11-2022

3.7.2 Function Documentation

3.7.2.1 typeWriter()

```
function typeWriter ( )
```
Return the ratio of the inline text length of the links in an element to the inline text length of the entire element.

3.7.3 Variable Documentation

3.7.3.1 i

```
var i = 0
```

Iterator for all letters in the word "Python Combat".

3.7.3.2 speed

```
var speed = 150
```

Speed of the animation.

3.7.3.3 title

```
var title = "Python Combat"
```

Title of the webpage.

Index

add_weapon
- level5.js, 22

arena_type
- level3.js, 16
- level4.js, 20

check_success
- level1.js, 8
- level2.js, 11
- level3.js, 15
- level4.js, 18
- level5.js, 22

create_arena
- level1.js, 8
- level2.js, 11
- level4.js, 19
- level5.js, 22

create_map
- level2.js, 11
- level3.js, 15

dict
- level2.js, 13

direction
- level2.js, 12

gem_found
- level5.js, 23

i
- welcome.js, 26

id_order
- level5.js, 23

index
- level5.js, 24

innerHTML
- level1.js, 9
- level2.js, 13
- level3.js, 16
- level4.js, 20
- level5.js, 24

insequence
- level5.js, 24

level1.js
- check_success, 8
- create_arena, 8
- innerHTML, 9
- log, 9
- logger, 9
- pos_x, 9
- pos_y, 9
- print_stat, 8
- reloaded, 9
- speed, 9
- speed_dict, 10
- speed_show, 10
- speeddown, 8
- speedup, 8
- total_gems, 10
- walk, 8

level2.js
- check_success, 11
- create_arena, 11
- create_map, 11
- dict, 13
- direction, 12
- innerHTML, 13
- log, 13
- logger, 13
- pos_x, 13
- pos_y, 13
- print_stat, 12
- reloaded, 13
- speed, 13
- speed_dict, 13
- speed_show, 14
- speeddown, 12
- speedup, 12
- total_gems, 14
- walk, 12

level3.js
- arena_type, 16
- check_success, 15
- create_map, 15
- innerHTML, 16
- log, 16
- logger, 17
- pos_x, 17
- pos_y, 17
- print_stat, 15
- reloaded, 17
- snowman_kill, 15
- snowman_killed, 17
- speed, 17
- speed_dict, 17
- speed_show, 17
- speeddown, 16
- speedup, 16
- walk, 16

level4.js
- arena_type, 20
- check_success, 18
- create_arena, 19
- innerHTML, 20
- log, 20
- logger, 20
- pos_x, 20
- pos_y, 20
- print_stat, 19
- reloaded, 20
- speed, 20
- speed_dict, 20
- speed_show, 21
- speeddown, 19
- speedup, 19
- total_stones, 21
- walk, 19

level5.js
- add_weapon, 22
- check_success, 22
- create_arena, 22
- gem_found, 23
- id_order, 23
- index, 24
- innerHTML, 24
- insequence, 24
- log, 24
- logger, 24
- pos_x, 24
- pos_y, 24
- print_stat, 23
- reloaded, 24
- speed, 24
- speed_dict, 25
- speed_show, 25
- speeddown, 23
- speedup, 23
- walk, 23
- weapon, 23

levels/level1/level1.js, 7
levels/level2/level2.js, 10
levels/level3/level3.js, 14
levels/level4/level4.js, 17
levels/level5/level5.js, 21

log
- level1.js, 9
- level2.js, 13
- level3.js, 16
- level4.js, 20
- level5.js, 24

logger
- level1.js, 9
- level2.js, 13
- level3.js, 17
- level4.js, 20
- level5.js, 24

pos_x
- level1.js, 9
- level2.js, 13
- level3.js, 17
- level4.js, 20
- level5.js, 24

pos_y
- level1.js, 9
- level2.js, 13
- level3.js, 17
- level4.js, 20
- level5.js, 24

print_stat
- level1.js, 8
- level2.js, 12
- level3.js, 15
- level4.js, 19
- level5.js, 23

README.md, 25

reloaded
- level1.js, 9
- level2.js, 13
- level3.js, 17
- level4.js, 20
- level5.js, 24

snowman_kill
- level3.js, 15

snowman_killed
- level3.js, 17

speed
- level1.js, 9
- level2.js, 13
- level3.js, 17
- level4.js, 20
- level5.js, 24
- welcome.js, 26

speed_dict
- level1.js, 10
- level2.js, 13
- level3.js, 17
- level4.js, 20
- level5.js, 25

speed_show
- level1.js, 10
- level2.js, 14
- level3.js, 17
- level4.js, 21
- level5.js, 25

speeddown
- level1.js, 8
- level2.js, 12
- level3.js, 16
- level4.js, 19
- level5.js, 23

speedup
- level1.js, 8
- level2.js, 12
- level3.js, 16

level4.js, 19
level5.js, 23

title
welcome.js, 26
total_gems
level1.js, 10
level2.js, 14
total_stones
level4.js, 21
typeWriter
welcome.js, 25

walk
level1.js, 8
level2.js, 12
level3.js, 16
level4.js, 19
level5.js, 23
weapon
level5.js, 23
welcome.js, 25
i, 26
speed, 26
title, 26
typeWriter, 25