

PSU_ZAPPY_2017

DOCUMENTATION

Summary

1. Project introduction
2. Run the project
3. Communication protocol IA
4. Communication protocol graphical interface
5. Create your own graphical interface
6. Create your own IA

1. Project introduction

The goal of this project is to create a network game.

Several teams confront on a tiles map containing resources.

The winning team is the one with 6 players who reached maximum elevation.

2. Run the project

First, please launch the “make” command to compile the project

Server:

To run the server binary, you have to set some rules, like:

- Map size
- Different team names
- Number of IA in each team
- The frequency time also described as the time used by each action
- Port of the server

Here is an example of arguments:

```
./zappy_server -p 8080 -x 10 -y 10 -c 3 -n AER Koalas DRP Students -f 2
```

Graphical interface and IAs:

The graphical interface and the IAs are contained in the same binary. By passing arguments, you can so launch IA with graphical interfaces or not, or just run the interface.

Some examples:

```
./zappy_ai -p 8080 -n AER -h localhost --no-graph --nb-ia 2
```

```
./zappy_ai -p 8080 -n Students -h localhost --nb-ia 1
```

3. Communication protocol IA

action	command	time limit	response
move up one tile	Forward	7/f	ok
turn 90° right	Right	7/f	ok
turn 90° left	Left	7/f	ok
look around	Look	7/f	[tile1, tile2,...]
inventory	Inventory	1/f	[linemate n , sibur n ,...]
broadcast text	Broadcast text	7/f	ok
number of team unused slots	Connect_nbr	-	value
fork a player	Fork	42/f	ok
eject players from this tile	Eject	7/f	ok/ko
death of a player	-	-	dead
take object	Take object	7/f	ok/ko
set object down	Set object	7/f	ok/ko
start incantation	Incantation	300/f	Elevation underway Current level: k /ko

4. Communication protocol graphical interface

SYMBOL	MEANING	SYMBOL	MEANING
X	width or horizontal position	n	player number
Y	height or vertical position	O	orientation: 1(N), 2(E), 3(S), 4(W)
q0	resource 0 (food) quantity	L	player or incantation level
q1	resource 1 (linemate) quantity	e	egg number
q2	resource 2 (deramere) quantity	T	time unit
q3	resource 3 (sibur) quantity	N	name of the team
q4	resource 4 (mendiane) quantity	R	incantation result
q5	resource 5 (phiras) quantity	M	message
q6	resource 6 (thystame) quantity	i	resource number

SERVER	CLIENT	DETAILS
msz X Y\n	msz\n	map size
bct X Y q0 q1 q2 q3 q4 q5 q6\n	bct X Y\n	content of a tile
bct X Y q0 q1 q2 q3 q4 q5 q6\n * nbr_tiles	mct\n	content of the map (all the tiles)
tna N\n * nbr_teams	tna\n	name of all the teams
pnw #n X Y O L N\n		connection of a new player
ppo n X Y O\n	ppo #n\n	player's position
plv n L\n	plv #n\n	player's level
pin n X Y q0 q1 q2 q3 q4 q5 q6\n	pin #n\n	player's inventory
pex n\n		expulsion
pbc n M\n		broadcast
pic X Y L n n ... \n		start of an incantation (by the first player)
pie X Y R\n		end of an incantation
pfk n\n		egg laying by the player
pdr n i\n		resource dropping
pgt n i\n		resource collecting
pdi n\n		death of a player
enw e n X Y\n		an egg was laid by a player
eht e\n		egg hatching
ebo e\n		player connection for an egg
edi e\n		death of an hatched egg
sgt T\n	sgt\n	time unit request
sst T\n	sst T\n	time unit modification
seg N\n		end of game
smg M\n		message from the server
suc\n		unknown command
sbp\n		command parameter

In addition to these default commands, we added:

SERVER	CLIENT	DETAILS
gnp n\n	gnp\n	Get Number of Player
gpt TEAM\n	gpt\n	Get Player Team
Plv\n	plv\n	Get Player Level

5. Create your own graphical interface

This part can be considered as a tutorial which will allow to create your own graphical interface in python. You have several ways to create a new interface; you can either create the server connection and thus start from the beginning or inherit from our class or overload the IA class. Let me introduce you the inherit solution below because it's the easiest way for you.

- Create a new python file
- Let's write a similar class:

```
from Server import Server
from graphicalInterface import GraphicalInterface
import threading

class NewGraphicalInterface(Server, GraphicalInterface, threading.Thread):
    def __init__(self, port=4242, ip="localhost"):
        super().__init__(port, ip)
        threading.Thread.__init__(self)
        self.readTh.start()
        self.manageConnection()
        self.daemon = True
```

This is the requirement for a basic communication.

- Implement the run method called by the main thread

```
def run(self):
    print(["Hey! I'm in a thread"])
```

When each thread will start, it'll call this function so please make a game-loop there

- You're lucky to inherit from an incredible class, called ThreadRead. It allows, when you are connected, to communicate to the server using an independent thread, let me explain. Each IA or graphical interface has a thread that will loop on reading the server. So, you can receive some news like the connection of a new player. Let's implement a simple method to get data about clients:

```
def getPlayerPosition(self, id):  
    self.write("ppo " + id) #Use the write method, from server inherit, to write data  
    pos = self.readTh.get_command()[1:] #Get response from the server, using blocking get_command method. We don't care about the first world, it'll be "ppo"  
    print("X: ", pos[0], "Y: ", pos[1])
```

- Now, you have enough knowledge to build your interface, good luck!

6. Create your own IA

Like the graphical interface, we'll inherit from the IA class, it'll be quick:

- Let's create a new python file
- Write a new class like this:

```
class NewIA(Server, IAServer, threading.Thread):  
    def __init__(self, team, port, ip):  
        threading.Thread.__init__(self)  
        self.server = IAServer(team, port, ip)  
        self.daemon = True
```

Is there something disturbing you? Again, you are lucky you have a beautiful class call IAServer to communicate with the server. Please check the class code to get more use information. The rest is easy to understand.

- Implement the run method called by the main thread

```
def run(self):  
    print("Hey! I'm in a thread")
```

- Let's do some magic! Good luck