



DOCUMENTATION ZAPPY

Projet réalisé par :

Charles Delachapelle

Lysandra Manguerra

Yoann Godard

Gabriel Rouchon

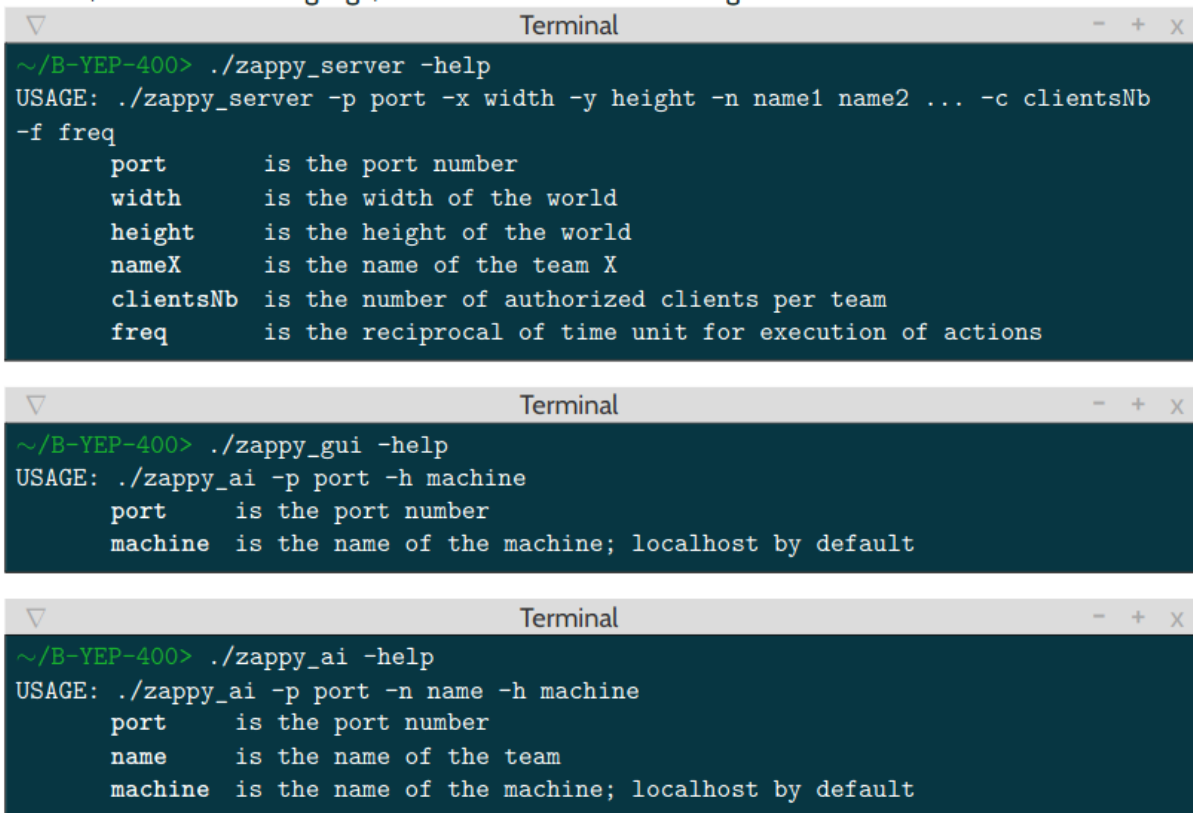
TABLES DES MATIÈRES

Introduction.....	1
Serveur.....	2
IA.....	3
GUI.....	4

Introduction

Ce projet consiste à créer une IA sur un serveur qui a pour but d'évoluer en ramassant de la nourriture et en effectuant des actions

Pour exécuter le programme veuillez taper la commande suivante dans votre terminal :



```
Terminal
~/B-YEP-400> ./zappy_server -help
USAGE: ./zappy_server -p port -x width -y height -n name1 name2 ... -c clientsNb
-f freq
    port      is the port number
    width     is the width of the world
    height    is the height of the world
    nameX     is the name of the team X
    clientsNb is the number of authorized clients per team
    freq      is the reciprocal of time unit for execution of actions

Terminal
~/B-YEP-400> ./zappy_gui -help
USAGE: ./zappy_ai -p port -h machine
    port      is the port number
    machine    is the name of the machine; localhost by default

Terminal
~/B-YEP-400> ./zappy_ai -help
USAGE: ./zappy_ai -p port -n name -h machine
    port      is the port number
    name      is the name of the team
    machine    is the name of the machine; localhost by default
```

LE SERVEUR

add_player.c → permet d'ajouter un joueur

attribute_values.c → attributs les valeurs en début de programme

check_look.c → permet de vérifier les cases.

check_values.c → permet de vérifier

create_map.c → créer la map et la redimensionner

elevation_system.c/system2.c → gère le système d'élévation

error_managment.c → c'est toute la gestion d'erreur

inventory_management.c → gère l'inventaire

manage_commands.c/commands2 → gère les commandes server/ia

manage_gui_commands.c → gère les commandes de l'interface graphique

move_function.c → gère les mouvements

normes_function.c → met à la norme le serveur

print_help.c → affiche le help

print_map_test.c → affiche le test de la map

replace_map.c → ajoute des ressources au fil de la game

server.c → le serveur

L'IA

environment.cpp →

handle_client.cpp → envoie le message et reçois le message

ia.cpp → setup l'ia et la connecte au serveur

incantation.cpp → permet de réaliser l'élévation

inventory.cpp → met à jour et gère l'inventaire des bots

main.cpp → gère la gestion d'erreur, manage l'ia et l'exécute

Le GUI

le gui est composé de deux dossiers, le Front et le Back

Le back:

- `getActionCommand.cpp` → Parser les commandes qui ne concerne pas le joueur.
- `getOtherCommand.cpp` → Parser les commandes qui concerne le joueur
- `guiToServer.cpp` → init les sockets recevoir et envoi des messages au serveurs
- `responseToServer.cpp` → les différents retour du GUI vers le serveur

Le Front:

- `createObject.cpp` → créer tous les objets(cristaux, food, joueurs)
- `dataFromServer.cpp` → récupération de données pour la map et les tags
- `gameScene.cpp` → l'affichage de la scène du jeu et sa gestion (inventaire, map etc..)

- InitRayLib.cpp → Initialisation des caméras pour le jeu de la window et des différents modèle (draw le jeu de manière générale)
- inputPlayer.cpp → gestion des input
- menuScene.cpp → gestion de toutes les scène et du menu.

Nous contacter

Si vous souhaitez des informations supplémentaires vous pouvez nous contacter par mail :

- charles.delachapelle@epitech.eu
- gabriel.rouchon@epitech.eu
- lysandra.manuguerra@epitech.eu
- yoann.godard@epitech.eu