

Documentation

Zappy: IA

Introduction de la partie IA:

Le but de l'IA était de pouvoir monter 6 personnes d'une même équipe au niveau 8 grâce à un algorithme. Cet algorithme va communiquer avec le serveur en lui envoyant des commandes à effectuer pour pouvoir survivre et passer niveau 8.

Tout d'abord nous nous connectons au serveur grâce à la librairie socket et avec la fonction login nous connectons l'IA avec le serveur.

```
def login(ip, port, name, timeout, ia):
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.settimeout(timeout)
    try:
        s.connect((ip, port))
        reponse = s.recv(1024)
        print("Réponse : ", reponse.decode())
        ia.teamname = name
        s.sendall(name.encode() + b'\n')
        reponse = s.recv(1024)
        rep = reponse.decode().split("\n")
        ia.client_num = rep[0]
        print("CLIENT: ", ia.client_num)
        print("client num : ", reponse.decode())
        return s
    except socket.timeout:
        raise ConnectionError(f"Erreur : le serveur {ip} sur le port {port} n'a pas répondu dans le délai imparti.")
    except Exception as e:
        raise ConnectionError(f"Erreur lors de la connexion au serveur : {e}")
```

Si la connexion réussit, le programme lance la fonction main_game qui permet de vérifier toutes les réponses reçues par le serveur et d'effectuer les actions voulues selon le mode choisit.

Au tout début du programme l'IA va ramasser le maximum de Food qu'il faut et rentrer dans le state "FOOD_MOD". Il quitte ce mode quand il atteint 70 de flood. La première IA qui atteint 70 de Food devient le chef pour la suite des évènements. Et ensuite va chercher toutes les pierres jusqu'à que l'inventaire global ait toute les pierres nécessaires à la montée level 8.

```

if ia.nb_food > 70:
    if ia.chief_is_designed == False:
        print("\n\n\n\n\n\n\n\n\n\nJE SUUUIS LE KIIIIIIIIIIIIIIIIIIIIING\n\n\n\n\n")
        ia.chief_is_designed = True
        ia.is_chief = True
        ia.steps = []
        chief = f"Broadcast -{ia.teamname}-chief"
        ia.steps.append(chief)
        handle_movement(s, ia)
    if ia.inventory_is_send == False:
        ia.inventory_is_send = True
        inv = f"Broadcast -{ia.teamname}-{ia.nb_linemate}-{ia.nb_deraumere}-{ia.nb_sibur}-{ia.nb_mendiane}-{ia.nb_phiras}-{ia.nb_thystame}"
        ia.steps.append(inv)
    if ia.state == "FOOD_MOD":
        ia.state = "OUAIS"

```

Nous avons choisi de gérer dans des queues séparer les commandes envoyées vers le serveur et les réponses reçu de la part du serveur pour faciliter la gestion des actions à effectuer.

- La queue de commandes envoyées :

Nous avons décidé de faire un système de queue pour anticiper le nombre de commande max que le serveur peut avoir en même temps : 10 commandes. Puis nous stoppons l'envoi de commandes au serveur quand le nombre de commandes envoyer atteint 10 et le nombre diminue à chaque réponse reçue.

```

def handle_movement(s, ia):
    if (ia.nb_command_send < 10 and ia.steps):
        command = ia.steps[0].encode() + b'\n'
        s.sendall(command)
        ia.nb_command_send += 1
        print("commande exec : ", command)
        ia.steps.pop(0)
        handle_movement(s, ia)

```

- La queue de réponses reçu :

Nous avons créé une queue qui traite chaque réponse et qui lance des actions selon le message reçu. S'il reçoit un message avec écrit Food c'est que c'est l'inventaire s'il reçoit ok, c'est que l'action effectuer a bien marché.

S'il reçoit message, c'est qu'un Broadcast a été effectué.

```
if (ia.reponse[0].find("food", 0, 6) != -1):
    ia.nb_command_send -= 1
    get_and_update_inventory(s, ia)
    passeed = False
elif (ia.reponse[0].find("player", 0, 8) != -1) and ia.state != "MEET_UP_CHIEF":
    ia.nb_command_send -= 1
    double_tableau = look_and_parse(s, ia)
    choose_mov(s, ia, double_tableau)
    passeed = True
elif (ia.reponse[0].find("message", 0, 9) != -1):
    rep = ia.reponse[0].split(",")
    if rep[1].find("readymeetup") != -1 and ia.state != "MEET_UP_CHIEF":
        current_team = rep[1].split("_")
        if current_team[1] == ia.teamname:
            gooo = f"Broadcast_{ia.teamname}_goo_{ia.random_num}"
            ia.steps.append(gooo)
            ia.reponse.popleft()
            handle_movement(s, ia)
        else:
            ia.meetup = True
            ia.reponse.popleft()
```

Quand le programme atteint 70 de Food, l'IA va commencer à chercher les pierres manquantes pour passer niveau 8 quand il a assez, l'IA va envoyer des messages pour se rejoindre. Grâce à la fonction meet-up et la fonction increment_global_inventory, nous allons pouvoir se rejoindre et effectuer toutes les actions nécessaires au passage jusqu'au niveau 8.

```

def increment_global_inventory(s, ia):
    rep = ia.reponse[0].split(",")
    message = rep[1].split("-")
    if message[2] == "chief" and message[1] == ia.teamname:
        ia.not_chief = True
        ia.chief_is_designed = True
        ia.reponse.popleft()
        return
    if message[2] == "jesuisla" and ia.state == "MEET_UP_CHIEF" and ia.teamname == message[1]:
        ia.nb_player += 1
        print("\n\n\n\n\nIL YYYYYYYYYYYY AAAAAAAAAAAAAAAAAA: ", ia.nb_player)
        print("\n\n\n\n\n")
        ia.reponse.popleft()
        return
    elif message[2] == "jesuisla" and ia.teamname == message[1] and ia.state != "MEET_UP_CHIEF":
        ia.nb_player += 1
        ia.reponse.popleft()
        return
    if message[2] == "je_suis_pret" and message[1] == ia.teamname and ia.state == "MEET_UP_CHIEF":
        ia.player_ready += 1
        ia.reponse.popleft()
        print("\n\n\n\n\n JOUEUR PRET A INCANTER: ", ia.player_ready)
        print("\n\n\n\n\n")
        return
    if message[2] == "je_suis_pret" and message[1] == ia.teamname and ia.state == "MEETUP":
        ia.player_ready += 1
        ia.reponse.popleft()
        return
    if message[1] == ia.teamname:
        ia.nb_global_linemate += int(message[2])
        ia.nb_global_deraumere += int(message[3])
        ia.nb_global_sibur += int(message[4])
        ia.nb_global_mendiane += int(message[5])
        ia.nb_global_phiras += int(message[6])
        ia.nb_global_thystame += int(message[7])
        print("\n global linemate : ", ia.nb_global_linemate)
        print("\n global deraumere : ", ia.nb_global_deraumere)
        print("\n global sibur : ", ia.nb_global_sibur)
        print("\n global mendiane : ", ia.nb_global_mendiane)
        print("\n global phiras : ", ia.nb_global_phiras)
        print("\n global thystame : ", ia.nb_global_thystame)
    ia.reponse.popleft()

```

```

def meetup(s, ia):
    tab = ia.reponse[0].split(",")
    tab2 = tab[0].split(" ")
    dir = tab2[1]
    print("je bouge : -", dir, "-")
    tab3 = tab[1].split("_")
    nb = int(tab3[3])
    if nb != ia.random_num:
        return
    if dir.find("0") != -1:
        ia.steps = []
        if ia.joined == False:
            meetup = f"Broadcast -{ia.teamname}-jesuisla-"
            ia.steps.append(meetup)
            ia.joined = True
            ia.nb_player += 1
        print("\n\n\n ON EST ENSEMBLE \n\n\n")
        handle_movement(s, ia)
        return
    elif dir.find("3") != -1:
        ia.steps.append("Left")
    elif dir.find("4") != -1:
        ia.steps.append("Left")
    elif dir.find("5") != -1:
        ia.steps.append("Left")
        ia.steps.append("Left")
    elif dir.find("6") != -1:
        ia.steps.append("Right")
    elif dir.find("7") != -1:
        ia.steps.append("Right")
    ia.steps.append("Forward")
    gooo = f"Broadcast _{ia.teamname}_goo_{ia.random_num}"
    ia.steps.append(gooo)
    handle_movement(s, ia)

```

```

def go_level_8(s, ia):
    while ia.nb_linemate > 0:
        ia.steps.append('Set linemate')
        ia.nb_linemate -= 1
    while ia.nb_deraumere > 0:
        ia.steps.append('Set deraumere')
        ia.nb_deraumere -= 1
    while ia.nb_sibur > 0:
        ia.steps.append('Set sibur')
        ia.nb_sibur -= 1
    while ia.nb_mendiane > 0:
        ia.steps.append('Set mendiane')
        ia.nb_mendiane -= 1
    while ia.nb_phiras > 0:
        ia.steps.append('Set phiras')
        ia.nb_phiras -= 1
    while ia.nb_thystame > 0:
        ia.steps.append('Set thystame')
        ia.nb_thystame -= 1
    if ia.state != "MEET_UP_CHIEF" and ia.broadcast_is_sended == False:
        ia.broadcast_is_sended = True
        broadcast = f"Broadcast -{ia.teamname}-je_suis_pret-"
        ia.steps.append(broadcast)
    if ia.state == "MEET_UP_CHIEF" and ia.incantation == False and ia.player_ready >= 6:
        ia.incantation = True
        ia.steps.append('Incantation')
        ia.steps.append('Incantation')
        ia.steps.append('Incantation')
        ia.steps.append('Incantation')
        ia.steps.append('Incantation')
        ia.steps.append('Incantation')
        handle_movement(s, ia)

```

Quand tous les personnages d'une même équipe se sont rejoints, ils posent tous les objets sur leurs cases pour passer level 8 et ainsi gagner la partie. En ajoutant à notre queue de commande les commandes pour poser les pierres (Set Object) et les commandes pour passer de niveau (Incantation).