

# **Super Zappy**

(Ce nom est temporaire, rassurez-vous)

# Table des matières

I - Présentation générale.....	3
II – Description.....	4
1) Le serveur.....	4
a) Le monde.....	4
b) Les équipes.....	4
c) Le déroulement du jeu.....	5
d) Le panneau de contrôle.....	5
2) Le joueur.....	6
a) Le client.....	6
b) L'équipe.....	6
c) Les classes.....	7
d) Le joueur.....	7
e) Les caractéristiques.....	7
f) Les compétences.....	8
g) Les actions.....	9
h) Les interruptions.....	9
3) Le rendu graphique.....	10
III – Intérêt et ambition du projet.....	11
1) A l'aboutissement.....	11
2) Possibilités d'extentions.....	11
a) Plateforme web.....	11
b) Arènes et championnats.....	11

# **I - Présentation générale**

Ce projet est un mélange entre un Zappy et un Corewar.

Le but est de connecter plusieurs IA, groupés en équipe, sur un serveur simulant un monde de cases (Zappy), chacune étant d'un type de terrain différent, pouvant contenir des items à ramasser, et ayant parfois un objet bloquant le passage sur elles.

Les IA seront exécutés à partir d'un script (Corewar) en javascript et/ou en python, et devront coopérer pour atteindre leur objectif, qui peut varier selon le type de partie.

Un autre type de client, graphique, pourra récupérer la carte et les actions des IA en temps réel pour afficher un rendu graphique compréhensible par le commun des mortels sur ce qui se passe sur le serveur.

## **II – Description**

### **1) Le serveur**

Réalisé en C++ pour des questions d'optimisation, il assure la génération du monde, les réponses aux clients IA, et la liaison entre la maintenance des informations générales aux clients graphiques.

#### **a) Le monde**

Le monde est une île de taille variable, composée de cases. Chacune d'entre elle est composée :

- Un type de terrain, représentant le sol qui la compose. Celui-ci varie selon son altitude et son humidité (proximité des rivières et des lacs) et ne peut être altéré.
- Un éventuel item ou groupe d'items trainant dessus. Celui-ci peut être ramassé par des IA, et ne bloque pas le passage.
- Un éventuel objet installé dessus. Celui-ci peut être naturel (arbre, buisson...) ou artificiel (Mur, Tour de guet...) et empêche les joueurs de passer par cette case.
- De l'éventuel joueur qui est actuellement dessus.

L'île est générée avant que les joueurs se connectent, certains paramètres pouvant être changés dans un fichier de configuration.

#### **b) Les équipes**

Les IA sont réparties en équipes, d'un nombre minimum de deux, dont les noms sont décidés au lancement du serveur. Les joueurs peuvent ensuite se connecter et se définir avant le lancement de la partie. (Voir partie IA)

### **c) Le déroulement du jeu**

Le jeu se déroule selon un système de cycle : chaque action demandée par un client prendra un certain nombre de cycles à exécuter, un cycle étant représenté par une unité de temps par le serveur, allant de 1 à 1000 millisecondes.

Lorsque le serveur reçoit une action à effectuer de la part d'un joueur il effectue les actions suivantes :

- Définir si celle-ci est possible d'essai, auquel cas contraire il renvoie un message d'erreur au client (dépendant de l'action).
- Evaluer le temps d'exécution que va prendre l'action. Celui-ci peut varier en fonction de nombreux paramètres : niveau de certaines compétences, position du joueur, nature de la cible si cible il y a...
- Attendre un nombre de cycles correspondant.
- Tenter d'exécuter l'action, et renvoyer un message correspondant. L'action peut échouer, auquel cas un message d'erreur est renvoyé.

Un joueur peut également recevoir pendant l'exécution d'une action un message l'informant de quelque chose pouvant l'intéresser : apparition d'un ennemi dans son champ de vision, message d'un allié... Auquel cas il a la possibilité d'interrompre son action en cours pour y réagir.

### **d) Le panneau de contrôle**

Le serveur a la possibilité de se lancer en mode "panneau de contrôle", auquel cas se lance un panneau en Ncurses depuis lequel il est possible de lancer certaines commandes au serveur, tels que arrêt de jeu, régénération de l'île, changement de l'unité de temps du cycle, etc.

## **2) Le joueur**

Répartis en équipe, chaque joueur sera composé d'un script javascript ou python qui s'exécutera, attendant les réponses du serveur entre chaque action et qui répondra en conséquences.

### **a) Le client**

Afin de mieux coordonner les joueurs d'une équipe et les possibilités de celles-ci (voir plus bas), une équipe sera représentée par un et un seul client, dont un thread s'occupera de la connection avec le serveur, et un autre pour chaque joueur à exécuter.

### **b) L'équipe**

Une équipe dispose d'un certain nombre de points sur sa définition, pouvant être répartis jusqu'à un certain maximum dans les définitions suivantes :

- Le nombre de joueurs.
- Le capital d'achat de départ, permettant aux joueurs de commencer avec des items sur eux.
- Le nombre de classes disponibles.
- Le nombre de points maximum à dépenser dans les caractéristiques pour chaque classe.
- Le nombre de points maximum à dépenser dans les compétences pour chaque classe.

Cette définition est la première chose que le client communique au serveur. Celui-ci disqualifiera l'équipe si il estime que celle-ci a dépassé le maximum total autorisé.

Le serveur vérifiera aussi si l'équipe se plie à ses propres règles : si elle tente de créer plus de joueurs que le nombre annoncé, par exemple, elle se fera disqualifier de même.

### c) Les classes

Chaque joueur se verra associé à une classe. Celle-ci à des limites définies par la définition de l'équipe.

Elle contient ses caractéristiques et ses compétences, ainsi que son script à exécuter lors du lancement de la partie.

### d) Le joueur

Chaque joueur possède un nom unique, et est associé à l'une des classes de son équipe. Deux joueurs avec la même classe auront exactement les mêmes caractéristiques, compétences et équipement de départ. Leur script d'exécution sera également le même. Seul leur position de départ différera légèrement.

Les joueurs ont également un capital de vie qu'ils doivent surveiller comme si leur vie en dépendait, puisque... Leur vie en dépend. Si un joueur voit son indice de vie atteindre 0, il meurt et disparaît de la partie, emportant son équipement avec lui dans l'outre-monde, et laissant sur la case où il est mort un cadavre à dépecer.

### e) Les caractéristiques

Définie dans la classe des joueurs, elles se dénombrent à 5 :

- La **force**, augmentant la portée et les dommages de certaines armes, ainsi que la contenance de l'inventaire.
- La **perception**, augmentant le champ de vision.
- La **dextérité**, n'ayant aucun effet en soi.
- L'**endurance**, augmentant l'indice de vie de départ, ainsi que la contenance de l'inventaire.
- L'**intuition**, n'ayant aucun effet en soi.

Elles sont définies dans la classe par un nombre pour chacun, proportionnelles à la capacité du joueur pour ce domaine. La somme de ses nombres doit être inférieure ou égal à 10 x l'indice de caractéristique maximum de l'équipe.

Toutes ces caractéristiques, en plus d'avoir pour certaines un effet particulier sur le joueur, débloquent des compétences.

## f) Les compétences

Contrairement aux caractéristiques, qui représentent des traits passifs, les compétences sont des domaines de talents bien spécifiques, liés généralement à des actions qu'elles débloquent.

Chaque classe dispose d'un niveau entre 0 et 4 pour chaque compétence. Chaque niveau doit être acheté avec des points de compétences, dont le maximum par classe est spécifié dans la définition de l'équipe. Pour qu'un niveau soit acheté, tous les niveaux précédents doivent être achetés. Certaines compétences exigent pour certains de leurs niveaux de maîtrises un niveau minimal dans certaines caractéristiques, voir certaines autres compétences.

Le prix des compétences varie, mais a toujours le même multiplicateur en fonction du niveau souhaité :

- (0) **Ignorant** : Niveau de départ pour toutes les compétences. N'attribue aucune capacité.
- (1) **Apprenti** : En cours de formation sur le domaine. Le joueur connaît les bases de la compétence. Il a accès aux commandes de base, avec quelques éventuels malus. Multiplicateur : x1.
- (2) **Adepté** : Le joueur est un représentant classique de pratiquant de cette compétence. Il maîtrise les compétences classiques, sans bonus ni malus. Multiplicateur : x2.
- (3) **Expert** : Le joueur est un spécialiste de cette compétence, dépassant les pratiquants habituels par son expérience et son savoir-faire. Il sait pratiquement tout faire dans ce domaine, avec quelques éventuels bonus. Multiplicateur : x3.
- (4) **Maître** : Le joueur maîtrise parfaitement cette compétence, en connaissant les moindres secrets par coeur. Il a accès à toutes les actions liées à cette compétence, en bénéficiant des plus grands bonus. Multiplicateur : x4.

La liste des compétences disponibles est non-exhaustive et peut varier avec les versions ou les modes de jeu, il est donc impossible d'en effectuer une liste complète ici. En voici néanmoins quelques exemples :

- **Orientation** : Trouve le plus rapide chemin vers un point.
- **Crafting** : Permet l'assemblage de certains items.
- **Construction** : Permet la construction de certains objets.



## **g) Les actions**

Le script d'un joueur est composé d'une succession d'action à tenter d'exécuter selon un certain algorithme. Chaque action est représentée sous la forme d'une fonction, prenant ou non un certain nombre de paramètres, et ayant une valeur de retour dépendant de l'action : certaines renvoient un booléen pour indiquer leur succès. D'autres renvoie un objet représentant le résultat de l'action (comme l'item ramassé pour l'action ramasser) ou null en cas d'erreur.

Le client est là pour simplifier au maximum la communication. Un simple script de localisation d'un buisson et du déplacement jusqu'à celui-ci ressemble à ça :

```
var bush = search(g_objects.bush);  
if (bush != null)  
{  
    displace(bush.position.x, bush.position.y);  
}
```

Ici, la fonction 'displace' renverra true en cas de déplacement réussi, et faux en cas d'impossibilité à s'y rendre. (Chemin obstrué par exemple)

## **h) Les interruptions**

Un joueur peut à tout moment décider d'un script à exécuter en cas de certains événements pouvant surgir pendant l'exécution d'une action : message d'un ami, apparition d'un ennemi, impact d'un projectile... Si il décide d'effectuer une autre action durant ce script, celle précédemment en cours s'annulera.

### **3) Le rendu graphique**

Un client graphique peut demander la composition de chaque chunk de l'île un par un. Il est aussi informé sans le demander de chaque changement qu'il y a lieu durant la partie : les actions des joueurs, le changement des cases...

Celui-ci peut être en 2D ou en 3D, tant qu'on peut comprendre le déroulement de la partie.

### **III – Intérêt et ambition du projet**

#### **1) A l'aboutissement**

Les jeux de programmation fonctionnant par scripting peuvent intéresser un bon nombre de développeurs. L'exportation du scripting en python peut également augmenter le nombre potentiel de joueurs : avec la nouvelle option de programmation du lycée, certains lycéen se lancent dans les bases du python.

#### **2) Possibilités d'extensions**

##### **a) Plateforme web**

Le problème de ce projet réside dans son manque d'accessibilité : il faut se fournir le client IA spécifique et se connecter à un serveur au moment où celui-ci est disponible pour faire une partie. Ceci est pratiquement impensable sans une interface web.

L'extension rendant ce projet commercialisable/source de startup serait donc un site permettant à la fois une présentation du jeu, munie d'un tutoriel pour le scripting, avec la possibilité de coder en ligne ; ainsi que la possibilité de lancer des serveurs à distance pour tester/jouer ses scripts.

Cela impliquera néanmoins de devoir faire un client graphique web, afin de pouvoir jouer sans rien télécharger.

##### **b) Arènes et championnats**

Si avec un système comme celui proposé ci-haut, un nombre conséquent de joueurs afflue, il paraît évident que l'un de leurs objectifs sera de revendiquer leur script comme le meilleur de tous. Pour les départager, un système de tournoi sera donc nécessaire.

Celui-ci pourra être inclus sur la plate-forme web, permettant des championnats réguliers ou non, donnant des récompenses ou des titres à la clé.