



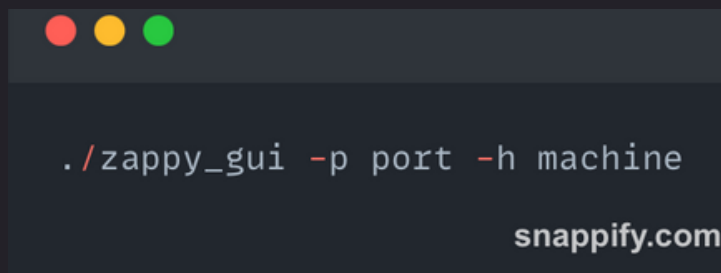
ZAPPY

Introduction :

Le client graphique pour Zappy, codé en C++ avec la bibliothèque Raylib, permet de visualiser le monde de Trantor en temps réel. Il offre une représentation graphique des joueurs, des ressources et des événements sur la carte.

Utilisation du GUI

Pour lancer le client graphique, utilisez la commande suivante avec les options appropriées :

A terminal window with a dark background and three colored window control buttons (red, yellow, green) at the top left. The command `./zappy_gui -p port -h machine` is entered in the terminal. The website `snappify.com` is visible in the bottom right corner of the terminal window.

```
./zappy_gui -p port -h machine  
snappify.com
```

Options:

- `-p port` : Numéro du port.
- `-h machine` : Nom de l'hôte du serveur (localhost par défaut).

Protocole de Connexion Client/Serveur

La communication entre le client graphique et le serveur se fait via des sockets TCP. Voici le protocole de connexion et les messages échangés :

Connexion Initiale :

- Le client ouvre un socket sur le port du serveur.
- Le serveur envoie : WELCOME\n
- Le client répond avec : GRAPHIC\n
- Le serveur envoie les dimensions de la carte

Commandes des Clients

Les clients répondent aux actions suivantes avec le format et les temps limites spécifiés :

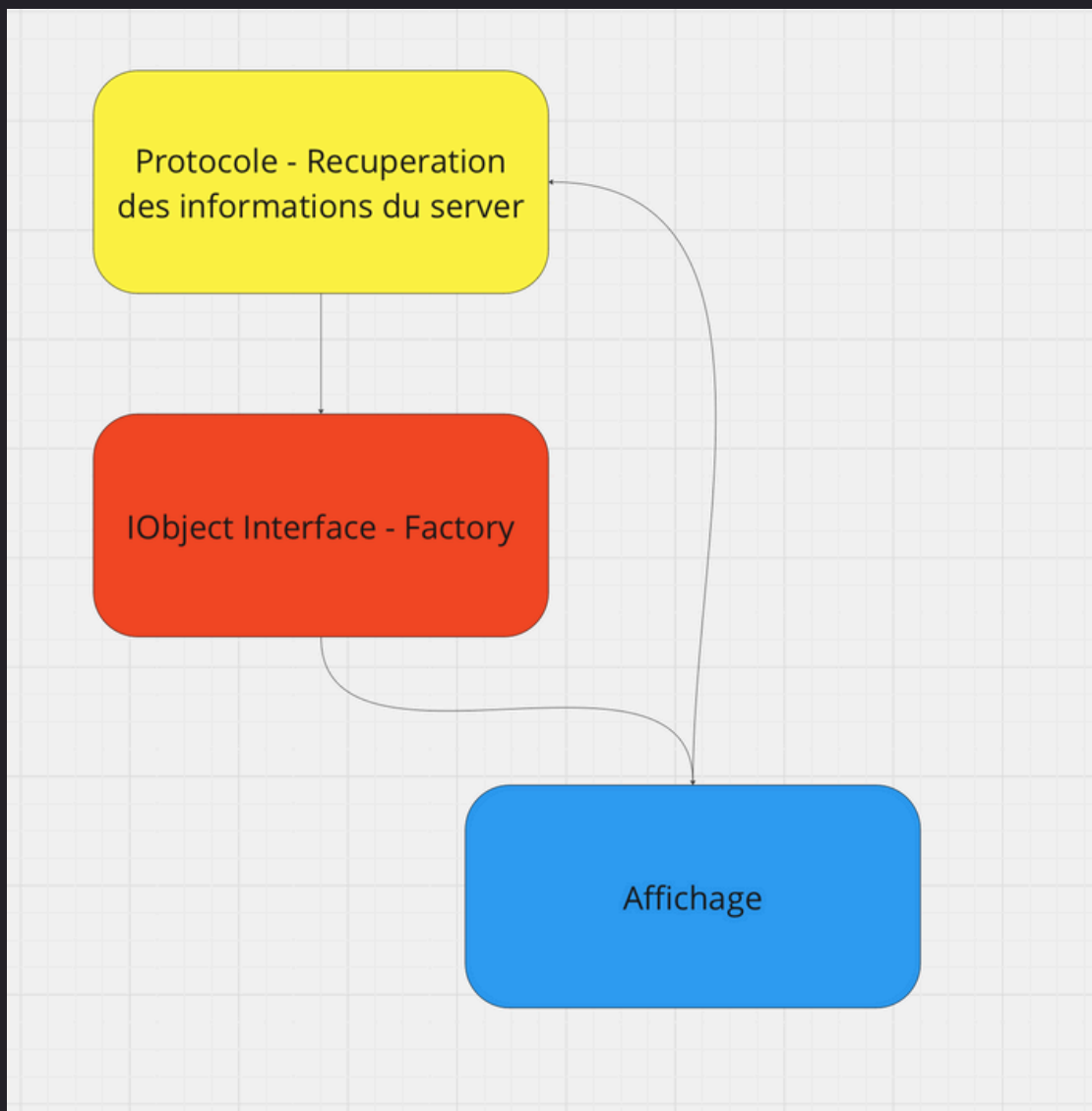
Commande envoyée par le GUI	Description	Réponse du serveur
`msz\n`	Demande de la taille de la carte	`msz X Y\n`
`bct X Y\n`	Contenu d'une case	`bct X Y q0 q1 q2 q3 q4 q5 q6\n`
`mct\n`	Contenu de toutes les cases	`bct X Y q0 q1 q2 q3 q4 q5 q6\n` pour chaque case
`tna\n`	Noms de toutes les équipes	`tna N\n` pour chaque équipe
`ppo #n\n`	Position d'un joueur	`ppo #n X Y 0\n`
`plv #n\n`	Niveau d'un joueur	`plv #n L\n`
`pin #n\n`	Inventaire d'un joueur	`pin #n X Y q0 q1 q2 q3 q4 q5 q6\n`
`sgt\n`	Demande de l'unité de temps	`sgt T\n`
`sst T\n`	Modification de l'unité de temps	`sst T\n`

Architecture

Notre architecture pour le client graphique est très simple :
Une récupération des données sur un processus séparé de la
génération et de l'affichage de notre jeu

Le protocole envoie les changements à opéré sur le jeu et si
nécessaire, crée de nouveaux objets qui seront ensuite rendus
dans le moteur

L'affichage utilise une méthode d'encapsulation de la raylib grâce
à une class "RaylibWrapper" et permet de stocker une instance de
la raylib dans notre class "Game"



Conclusion

Cette documentation couvre les aspects essentiels du client graphique pour Zappy, y compris la configuration initiale, le protocole de connexion, les commandes et les notifications d'événements. En suivant ces instructions, vous serez en mesure de visualiser et d'interagir efficacement avec le monde de Trantor. Pour toute question ou problème technique, n'hésitez pas à consulter la documentation détaillée fournie avec le projet ou à contacter un expert en développement graphique.