

ZAPPY Documentation



Realized by :

Romain Pillot, Antonin Rapini, Wilka Sombrun
Clement Gomis, Hugo Meyer, Killian Mayans

Summary

- I. Zappy – The Game
- II. How to launch
- III. Architecture
 - a. Server
 - b. Client - The Artificial Intelligence
 - c. Client – The Graphical User Interface (GUI)
 - d. Client – The network library
 - e. Global Structure
- IV. Bonus

I. Zappy – The Game

Multiple teams of AIs race to have six players reach the highest level on a tiled map.

Each tiles of this map will contain resources that the AI's will try to collect :

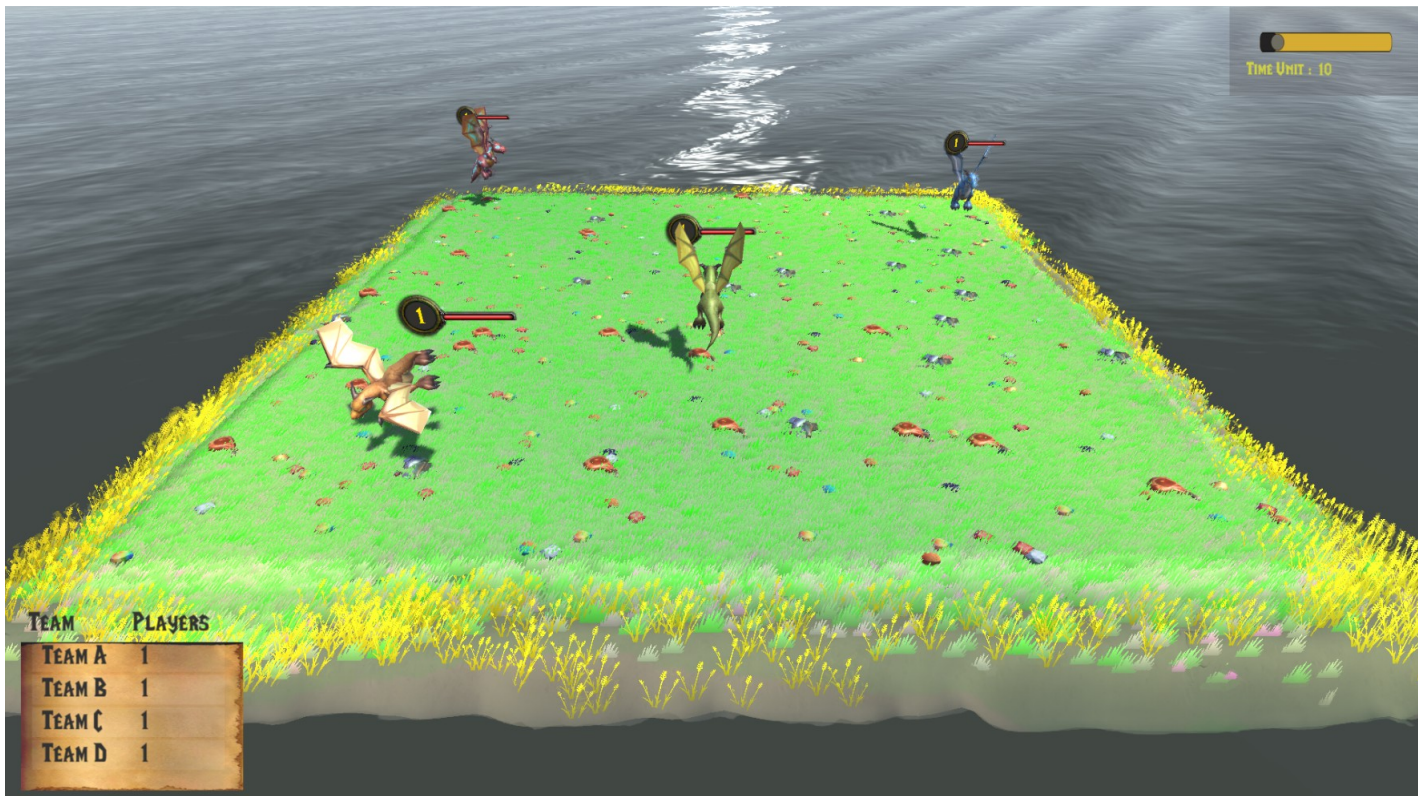
- Stones, which will be used to complete the rituals to level up
- Food, to keep the AI's alive

Mutual assistance will be the key to victory.

II. How to launch

For launch the game :

- 1) Use the make command to compile the project
- 2) Launch the server (./zappy_server -help for usage)
- 3) Launch the ai's (./zappy_ai -help for usage)
- 4) Launch the GUI (./zappy_gui)
 - a) Enter the port and localhost then click play and watch



III. Architecture

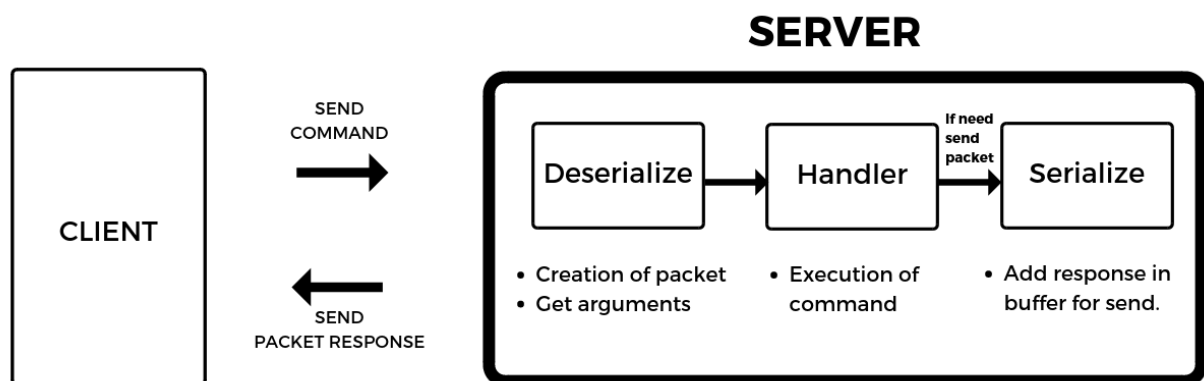
The zappy project is composed of multiples parts : a server, clients, and a graphical interface.

A) The server

The server (coded in C) centralizes the games logic, it receives commands from the AI's, treats them, then dispatch answers to the relevant parties.

The server's logic is the following :

- Create the network to allow connections from the clients
- Initialize the game (map, teams, etc)
- Poll messages from the clients

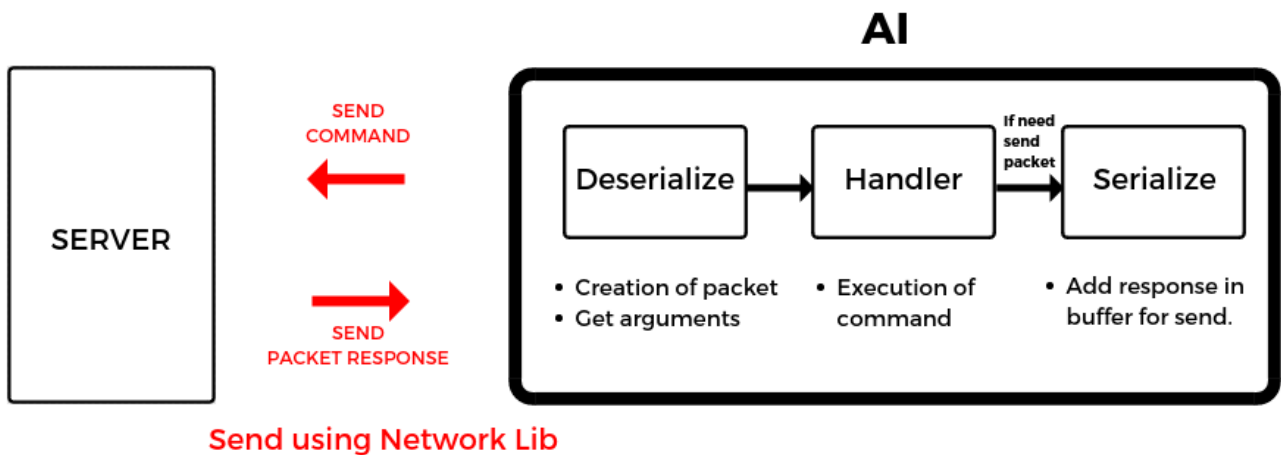


the server - client relationship

B) Client - The Artificial Intelligence (AI)

The AI (coded in C++), which corresponds to a player, will establish the best actions possible to ensure the victory for his team.

It sends commands to the server and reacts to its answers strategically.



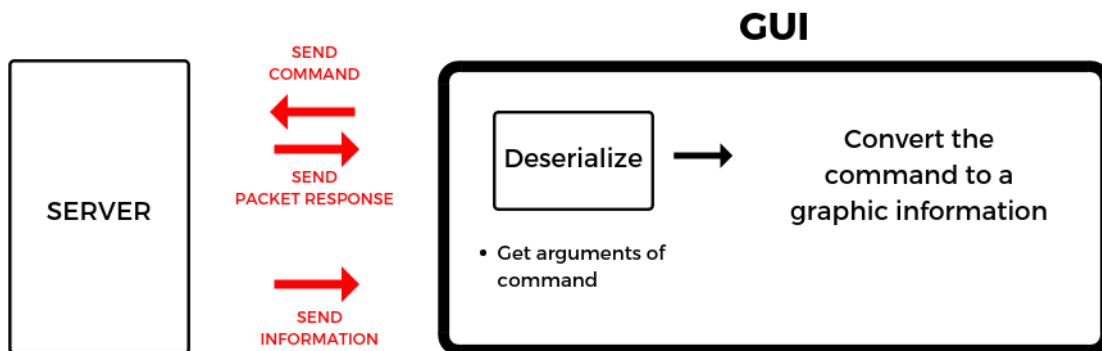
the ai - server relationship

The AI connects six players for each time. Only the first one enters the game the others are put in a waiting queue. They will enter the game one by one each time an egg laid by a player of their team hatches.

C) Client – The Graphical User Interface (GUI)

The GUI (coded in C# with Unity) gives a visual representation of each element and actions of the game.

The GUI continuously receives information from the server, interprets them and represents them visually. It can also send commands to the server to enhance the user's visual experience (ex : time unit changes)



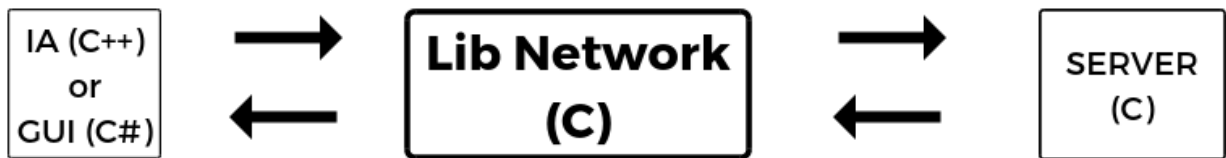
Send using Network Lib

the server - gui relationship

D) Client – The network library

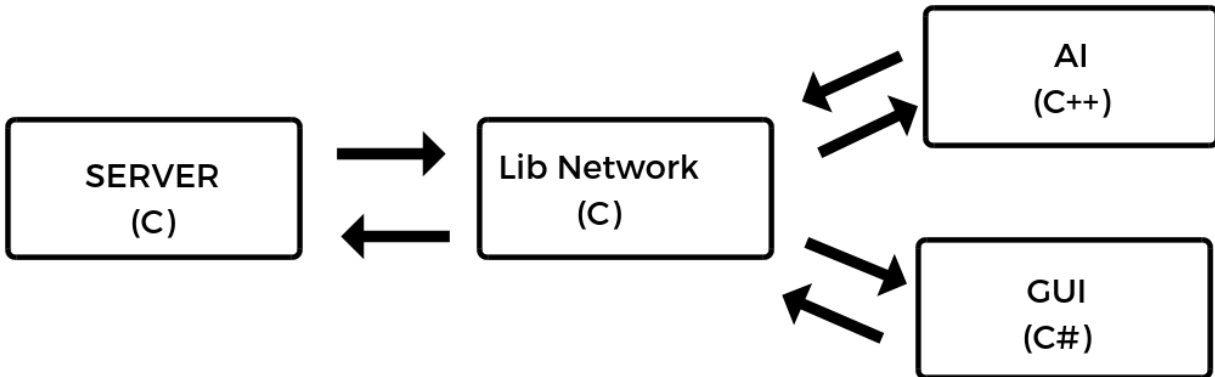
The network library (coded in C), allows the IA and the GUI to communicate.

The IA and the GUI defines callbacks (methods that will be called by the library) and communicates them in the form of a configuration structure to the library.



the place of the network library in the architecture

E) Global structure



recap of the the relations between the differents parts of the Zappy

First the server is launched, creating the network, and waits for each team before launching the game.

Once the game is launched, the server receives commands from the AI's, treats them and execute the corresponding actions. The AI's interprets the answers from the server and determines the best course of action to ensure its team's victory.

When the game's state changes (ex : a player moving), the server will communicate the information to the GUI which will then update the visual representation of the game state.

The network communication between the clients (AI and GUI) and the server, will be handled by the network library.

IV. Bonus

- Network Full C
- Native Lib
- Change Unit Time directly in gui
- Multiple Graphics clients possible