Ares - Game Demo

1. Description

The project consists in develop two parts of a shooting game. The first part is the AresUnityDemo and as the name suggests was build using Unity3D ans it's responsible for all the graphic part of the game: car movement and physics, target movement and so on. The second part is is the AresGameInput and it's responsible for all the interface between user and game: command input, communication controller, log and etc.

4. AresUnityDemo

- Client.cs
    - Class responsible for manage game(client) connection with the input server.
    - To prioritize speed, was choose UDP as the protocol for the communication.
    - Consists in initialize the connection between client-server and controls the exchange of data(keyboard/mouse inputs and signal to connect/disconnect).
- EnemyTarget.cs
    - Class responsible for target controller.
    - Initialize randomly the movement pattern between circular, horizontally and sinus wave
    - Manages all the collisions with the scene
    - Responsible for kill the target if hit by a projectile
- GenerateScene.cs
    - Initializes the targets on random position on the scene
- Player.cs
    - Class responsible for all the player mechanics including car, weapon and camera movements and actions.
    - It creates a connection with the server and triggers a thread update the class with the user inputs from the server.
    - Receives a array consisting in flags indicating if the movement keys(a,w,s,d) or shooting button(mouse left button) are pressed down or up, and also receives the mouse cursor position relative to its last position, delta x and delta y.
    - When comes time to end the game, sends the signal to server and closes the game.

5. AresGameInput

- Server.h
    - Library for server controller
    - Create a UDP server on 127.0.0.1:8081 and waits for game client. After connection is set sends input data for client in a constant way.
    - Controls all the data flow between client and server
- PlayerController.h
    - Library for input controller
    - Uses xlib for grab mouse and keyboard, and capture all the inputs from them
- Server.cpp
    - Responsible for initialize the server and input capture
    - Triggers a thread the will listen on 127.0.0.1:8089 for the end game signal and signalize to the server that it is time to shut down

6. Execution

1. Build AresGameDemo
2. run **bash run_server.sh** on terminal
3. start AresGameDemo!