player.hpp

```
#pragma once

#include "headers.hpp"
#include "map.hpp"
#include "object.hpp"
#include "client.hpp"
#include "toaster.hpp"
#include <SFML/Audio.hpp>


class Player
{
private:
    // game logic
    sf::RenderWindow& window;
    Toaster& toaster;
    bool has_quit;
    bool dead = false;

    sf::Texture* wall_texs;
    sf::Sprite wall_sprite;

    sf::Texture enemy_tex;
    vector<Object> objects;
    vector<Object*> sorted_objects;



    // gun animation
    int gun_animation_frame;
    float gun_animation_timer, gun_movement_stopwatch;
    float* gun_animation_duration;
    sf::Sprite gun_sprite;
    sf::Texture* gun_texs;
    v2f gun_position;
    float gun_offset_y = 0;
    v2f gun_offset;
    float max_hand_range = 40;
    float hand_move_range;

    string verbs[9] = { "zoinked", "zooked", "styled on",
        "slaughtered", "kassifed on", "nuked",
        "eradicated", "decimated", "pulverized" };

    // hit direction and reticle indicator
    sf::Texture indicator_texture, reticle_texture;
    sf::Sprite hit_indicator_sprite, reticle_sprite;
    vector<float> hit_direction_timers;
    vector<float> hit_direction_angles;
    float reticle_timer = -1;


    //gun shot
```

```
    bool gun_shot;
    sf::Texture damage_overlay_tex;
    sf::Sprite damage_overlay_sprite;
    float current_damage_opacity;
    float max_damage_opacity = 130;
    string killer_name;

    // movement
    v2f position;
    float speed = 2.0f;
    bool running = false, crouching = false;
    bool moving = false, moving_forward;
    float run_multiplier = 1.75f, crouch_multiplier = 0.5f;

    // orientation
    float rotation_x = -3.169f;
    float rotation_y = -0.56f;
    float mouse_sensitivity = 0.05f;
    float fov_y = 0.7f;
    float fov_x = 1.22173f; // 70 degrees

    // map
    float body_radius = 0.4f;

    //sound
    sf::SoundBuffer gunshot_buffer, gunclick_buffer;
    sf::Sound gun_sound, click_sound;

    //font
    sf::Font nametag_font, bold_font, deathscreen_font;

    //debug
    float debug_float = 0;

    int received_events_size = 0;
    char received_events[128];

    int score = 0;

public:
    Client client;
    std::mutex mtx;
    Map map;
    bool window_focused;

    // leaderboard
    vector<Toaster::LeaderboardEntry> leaderboard;

    bool debug_mode = false;

    struct HitInfo {
        float distance;
        bool on_x_axis;
```

```
    float texture_x;

    float perceived_distance;
};



Player(int x, int y, sf::RenderWindow& window, Toaster& toaster);

void setFocus(bool focus);
void handleKeys(float dt);
void rotateHead(int delta_x, int delta_y, float dt);
void move(float angle_offset, float dt);

void shootRays(HitInfo*& hits);
void drawWorld(HitInfo*& hits, float dt);
void drawColumn(int x, const Player::HitInfo& hit_info);
Player::HitInfo shootRay(float angle_offset);

void drawObject(Object& object, float dt);
void drawGun(float dt);
void drawCrosshair(float dt);
void drawDeathScreen(float dt);

void shootGun(bool left_click);
void getShot(int shooter_id);

void loadSFX();
void loadTextures();

void quitGame();

//server
void updateServer();
void listenToServer();
void handleEvents(char* events, int event_count);
Client::PlayerInfo getPlayerInfo();

Object* getObject(int id);
Object* getAnyObject();
void handle_shooting_victim(int victim_id, int shooter_id);
void handle_killing(int killer_id, int victim_id);
void getKilled(const string& killer_name);
void respawn();

void addToLeaderboard(int player_id, int score, const string&
username);
void updateLeaderboard(int player_id);


string getUsername(int id);

//debug
```

```
    void debug();

};
```