

object.cpp

```
#include "headers.hpp"
#include "object.hpp"
#include "client.hpp" // for struct PlayerInfo

Object::Object(float x, float y, const sf::Texture& tex)
    : player_id(-1), position(x, y), direction_index(0)
{
    tex_size = (v2f)tex.getSize();
    sprite.setTexture(tex, true);
    sprite.setTextureRect(getTextureRect(direction_index, 0));
    shrink_by = 1.9f;
    scale_by = 0.0039f;
}

float Object::distFrom(const v2f& pos)
{
    return sqrtf(pow(pos.x - position.x, 2) + pow(pos.y - position.y,
2));
}

void Object::animate(float dt)
{
    if (dying_timer >= 0)
    {
        dying_timer += dt;
        if (dying_timer > 0.5f)
        {
            dead = true;
            sprite.setTextureRect(getTextureRect(4, 8));
            dying_timer = -1;
        }
        else
        {
            int animation_index = dying_timer * 10;
            sprite.setTextureRect(getTextureRect(animation_index, 8));
        }
        return;
    }

    if (dead)
        return;

    if (gun_timer >= 0)
    {
        gun_timer += dt;

        if (gun_timer > 0.2f)
            gun_timer = -1;
    }
}
```

object.cpp

```
else if (gun_timer < 0.03f)
    sprite.setTextureRect(getTextureRect(direction_index, 5));
else if (gun_timer < 0.12f)
    sprite.setTextureRect(getTextureRect(direction_index, 6));
else
    sprite.setTextureRect(getTextureRect(direction_index, 5));

return;
}

if (getting_shot_timer >= 0)
{
    getting_shot_timer += dt;
    if (getting_shot_timer > 0.22f)
        getting_shot_timer = -1;

    sprite.setTextureRect(getTextureRect(direction_index, 7));
    return;
}

if (!moving)
{
    sprite.setTextureRect(getTextureRect(direction_index, 4)); // 4
    - standing frame
    return;
}

float interval = 0.2f;
animation_timer += dt;
if (started_moving || animation_timer > interval)
{
    animation_timer -= interval;

    // there are 4 animation frames
    if (forward)
        animation_index = (animation_index + 1) % 4;
    else
        animation_index = (animation_index - 1 + 4) % 4;

    sprite.setTextureRect(getTextureRect(direction_index,
        animation_index));

    started_moving = false;
}
}

void Object::shootGun()
{
    gun_timer = 0;
}
```

object.cpp

```
void Object::gotShot()
{
    getting_shot_timer = 0;
}

void Object::gotKilled()
{
    dying_timer = 0;
}

void Object::loadPlayerInfo(Client::PlayerInfo player_info)
{
    player_id = player_info.player_id;

    username = player_info.username;

    position.x = player_info.pos_x;
    position.y = player_info.pos_y;

    rotation_x = player_info.rot_x;

    bool moving_flag = player_info.flags &
Client::PlayerInfo::Flag::moving;
    bool forward_flag = player_info.flags &
Client::PlayerInfo::Flag::forward;
    bool shot_gun_flag = player_info.flags &
Client::PlayerInfo::Flag::gun_shot;
    bool got_shot_flag = player_info.flags &
Client::PlayerInfo::Flag::got_shot;
    bool dead_flag = player_info.flags &
Client::PlayerInfo::Flag::dead;

    if (moving_flag != moving)
    {
        //player started or stopped moving
        started_moving = true;
    }

    dead = dead_flag;

    if (dead)
        sprite.setTextureRect(getTextureRect(4, 8));

    moving = moving_flag;
    forward = forward_flag;

    if (shot_gun_flag)
        shootGun();
```

object.cpp

```
}

sf::IntRect Object::getTextureRect(float rotation, float frame)
{
    return sf::IntRect(280 * rotation, 280 * frame, 280, 280);
}

Object::Object()
{
    player_id = -1;
}
```