

Tavernlight

Install Visual Studio 2022

Download from <https://visualstudio.microsoft.com/vs/community/>

Install compiler and english language pack

Install vcpkg

```
cd \  
  
git clone https://github.com/Microsoft/vcpkg  
  
cd vcpkg  
  
.\bootstrap-vcpkg.bat  
.\vcpkg integrate install
```

Add the `C:\vcpkg` to the system's `PATH`

Add a new ambient variable called `VCPKG_ROOT` with the value `C:\vcpkg`

Install MySQL

Download from <https://dev.mysql.com/downloads/installer/>

Install and, on configure, set root password to `mysql`

When asked the password mode, do not use the most advanced/new encrypted one, use the old one (Legacy Authentication)

Add the `C:\Program Files\MySQL\MySQL Server 8.0\bin` to the system's `PATH`

Edit `C:\ProgramData\MySQL\MySQL Server 8.0\my.ini` and add to the end: `sql_mode=''`

The Forgotten Server

```
git clone --recursive https://github.com/otland/forgottenserver.git server  
  
cd server  
  
git checkout origin/1.4
```

Add `vcpkg.json` to `<server root>\vc17` folder with content:

```
{  
  "dependencies": [  
    "boost-iostreams",  
    "boost-asio",  
    "boost-system",  
    "boost-variant",  
    "boost-lockfree",  
    "luajit",  
    "libmariadb",  
    "pugixml",  
  ]  
}
```

```

        "cryptopp",
        "fmt",
        "boost-filesystem"
    ],
    "builtin-baseline": "8518284c18b9f47b1ccae804cb68694181b780a8",
    "overrides": [
        {
            "name": "fmt",
            "version": "8.1.1#0"
        }
    ]
}

```

Open `<server root>\vc17\theforgottenserver.sln` and, in Visual Studio, go to **Project > Properties**, then **Configuration Properties > vcpkg** and set **Use Vcpkg Manifest** to **Yes**

In the drop downs, select **Release** and **x64**

Then **Build > Build Solution**

Copy the `<server root>\data` directory to `<server root>\vc17\x64\Release\data`

Copy the `<server root>\config.lua.dist` to `<server root>\vc17\x64\Release\config.lua`

Change the `-- MySQL` "section" to look like the following in `<server root>\vc17\x64\Release\config.lua`:

```

-- MySQL
mysqlHost = "127.0.0.1"
mysqlUser = "root"
mysqlPass = "mysql"
mysqlDatabase = "forgottenserver"
mysqlPort = 3306
passwordType = "sha1"

```

Copy the `<server root>\key.pem` to `<server root>\vc17\x64\Release\key.pem`

From command line, inside the root directory, execute: `mysql -u root -p`

Type in your password (`mysql`)

Execute:

```

CREATE DATABASE forgottenserver;

exit

```

From command line, inside the `<server root>` directory, execute: `mysql -u root -p forgottenserver < schema.sql`

Type in your password (`mysql`)

From the `<server root>\vc17\x64\Release` run `theforgottenserver-x64.exe`

HTTP Server

Download from <https://sourceforge.net/projects/miniserver/files/>

Get the `15_0_2_ZeroXV.exe`

Install it and run `UniController.exe`

Click `Start Apache`

Delete all contents from `<UniServerZ root>\www` directory

Get <https://github.com/gesior/Gesior2012> and extract all the files inside `<UniServerZ root>\www` directory

On browser, go to `http://localhost/install.php`

Change your IP in `<UniServerZ root>\www\install.txt` if needed

Go to 1. Set server path on the left panel

Type in the path to the `<server root>\vc17\x64\Release` directory

Go through each of other steps and on 5. Set Admin Account set the password for the user 1 to 1

Go to `http://localhost`

Click `Logout`

Click `Create Account`

Fill in your data

Click on `I agree`

Click `Login`

Type in your credentials

Click `Create Character`

Fill in your data

Click `Submit`

Mehah's OTClient

```
git clone --recursive https://github.com/mehah/otclient.git client
```

Open `<client root>\vc17\otclient.sln`

In the drop downs, select `Release` and `x64`

Then `Build > Build Solution`

Inside `<client root>\data\things`, create a directory called `1098`

Get `Tibia.spr` and `Tibia.dat` files from `Tibia 10.98` and put inside the `<client root>\data\things\1098` directory

(Got from <https://otserverlist.me/tibia-clients-download.php>)

Run `<client root>\otclient_x64.exe`

Put your `username` and `password` and select `10.98` in `Client Version` drop down

Question 1

Fix or improve the implementation of the below methods

```
local function releaseStorage(player)
    player:setStorageValue(1000, -1)
end
```

```
function onLogout(player)
    if player:getStorageValue(1000) == 1 then
        addEvent(releaseStorage, 1000, player)
    end

    return true
end
```

Solution

It is not needed to call `addEvent` here, so I stripped it off. But, if we want to use it, it accepts only one argument, that is the callback function. So it can be an anonymous function or create a separate function to deal specifically with the storageId `1000` (in later case, `player` is out of scope and would need to get it with `g_game.getLocalPlayer()`, which can then be not found if the `logout` process already occurred when the event is dispatched)

If we do not want to reuse the `releaseStorage` function, then:

```
function onLogout(player)
    -- If storage 1000 is set to 1, release it
    if player:getStorageValue(1000) == 1 then
        player:setStorageValue(1000, -1)
    end

    return true
end
```

If we want to reuse the `releaseStorage` function, then:

```
-- Release storageId for the given player
-- If storageId is negative here, it will be wrapped in C++ because it is handled as uint32_t. So,
let's avoid it
function releaseStorage(player, storageId)
    -- Validate arguments
    if type(player) ~= 'userdata' or type(storageId) ~= 'number' then
        return
    end

    if storageId < 0 then
        return
    end

    -- Release storage's slot
    player:setStorageValue(storageId, -1)
end

function onLogout(player)
    -- If storage 1000 is set to 1, release it
    if player:getStorageValue(1000) == 1 then
        releaseStorage(player, 1000)
    end

    return true
end
```

Question 2

Fix or improve the implementation of the below method

```
function printSmallGuildNames(memberCount)
    -- this method is supposed to print names of all guilds that have less than memberCount max
    members
    local selectGuildQuery = "SELECT name FROM guilds WHERE max_members < %d;"
    local resultId = db.storeQuery(string.format(selectGuildQuery, memberCount))
    local guildName = result.getString("name")

    print(guildName)
end
```

Solution

Modify `<server root>\vc17\x64\Release\data\talkactions\talkactions.xml` and add this child to the `<talkactions>` node:

```
<talkaction words="!q2" separator=" " script="q2.lua" />
```

Then create a file called `q2.lua` inside `<server root>\vc17\x64\Release\data\talkactions\scripts` with the content:

```
-- Suppose the following tables:
--
-- rd_players
--     id    AUTO INCREMENT
--     name  VARCHAR(45)
--
-- rd_guilds
--     id    AUTO INCREMENT
--     name  VARCHAR(45)
--
-- rd_players_guilds
--     player_id  INT
--     guild_id   INT
--     fk_player_id ON  player_id = rd_players.id
--     fk_guild_id  ON  guild_id  = rd_guilds.id

-- Configuration
-- If true, it will also returns guild with 0 players in it
local ACCEPT_EMPTY_GUILDS = true

-- Accepts a number as maxMembersCount argument
-- Returns a table with all guilds' name and member count that have less or
-- equal maxMembersCount
function getSmallGuildNames(maxMembersCount)
    -- If we do not want guilds with 0 players (which is probably impossible to
    -- have), it will use INNER JOIN, otherwise it will use LEFT JOIN
    local selectGuildsQuery = [[
        SELECT
            `g`.`name`                AS `guild_name`,
            COUNT(`pg`.`player_id`) AS `members_count`
    ]]
```

```

FROM
    `rd_guilds` AS `g`
    %s JOIN
        `rd_players_guilds` AS `pg`
    ON
        `pg`.`guild_id` = `g`.`id`
GROUP BY
    `g`.`id`
HAVING
    `members_count` <= %d
ORDER BY
    `members_count` ASC,
    `guild_name` ASC;
]]

```

```

-- Format the query and execute
local selectGuildsPreparedQuery = string.format(selectGuildsQuery, ACCEPT_EMPTY_GUILDS and
'LEFT' or 'INNER', maxMembersCount)
local selectGuildsResultId      = db.storeQuery(selectGuildsPreparedQuery)

-- If empty, return empty
if selectGuildsResultId == false then
    return {}
end

local guildsEntries = {}

-- Iterate over results and simply push them into returning array
repeat
    guildName      = result.getString(selectGuildsResultId, 'guild_name')
    membersCount   = result.getNumber(selectGuildsResultId, 'members_count')

    local guildEntry = string.format(
        '%s (%d member%s)',
        guildName,
        membersCount,
        membersCount ~= 1 and 's' or ''
    )

    table.insert(guildsEntries, guildEntry)
until not result.next(selectGuildsResultId)

-- Free the query results
result.free(selectGuildsResultId)

-- Return found guilds
return guildsEntries
end

-- Process !q2 <max members count> command
-- Always returns false to not echo the command back to the player
function onSay(player, words, param)
    -- Validate param
    if #param == 0 then
        local message = 'Usage: !q2 <max members count>'

```

```

        player:sendTextMessage(MESSAGE_STATUS_CONSOLE_RED, message)

        return false
    end

    local maxMembersCount = tonumber(param)

    if maxMembersCount == nil then
        local message = 'The max members count should be a number'

        player:sendTextMessage(MESSAGE_STATUS_CONSOLE_RED, message)

        return false
    end

    local minAcceptedMembersCount = ACCEPT_EMPTY_GUILDS and 0 or 1

    if maxMembersCount < minAcceptedMembersCount then
        local message = string.format(
            'The max members count should be greater or equal than %d',
            minAcceptedMembersCount
        )

        player:sendTextMessage(MESSAGE_STATUS_CONSOLE_RED, message)

        return false
    end

    -- Retrieve all guilds with <= maxMembersCount
    local guildsEntries = getSmallGuildNames(maxMembersCount)

    -- If no results, inform player
    if #guildsEntries == 0 then
        local message = 'No guilds were found'

        player:sendTextMessage(MESSAGE_STATUS_CONSOLE_BLUE, message)

        return false
    end

    -- Format the response to the player and send it
    local messageTitle = string.format(
        'There %s %d guild%s with %d member%s or less:\n',
        #guildsEntries ~= 1 and 'are' or 'is',
        #guildsEntries,
        #guildsEntries ~= 1 and 's' or '',
        maxMembersCount,
        maxMembersCount ~= 1 and 's' or ''
    )

    local messageBody = table.concat(guildsEntries, '\n')
    local message = messageTitle .. messageBody

    player:sendTextMessage(MESSAGE_STATUS_CONSOLE_BLUE, message)

```

```
    return false
end
```

Configure the `ACCEPT_EMPTY_GUILDS`. To accept empty guilds in result, set it to `true`. Otherwise, set it to `false`

Restart the server and, in game, use as follow:

```
!q2 <max members count>
```

Question 3

Fix or improve the name and the implementation of the below method

```
function do_sth_with_PlayerParty(playerId, membername)
    player = Player(playerId)
    local party = player:getParty()

    for k,v in pairs(party:getMembers()) do
        if v == Player(membername) then
            party:removeMember(Player(membername))
        end
    end
end
```

Solution

Modify `<server root>\vc17\x64\Release\data\talkactions\talkactions.xml` and add this child to the `<talkactions>` node:

```
<talkaction words="!q3" separator=" " script="q3.lua" />
```

Then create a file called `q3.lua` inside `<server root>\vc17\x64\Release\data\talkactions\scripts` with the content:

```
-- Configuration
local ASSERT_PARTY_LEADERSHIP = true

-- Remove member with memberName name from player with id playerId party
function removeMemberFromPlayerParty(playerId, memberName)
    -- Get player by ID
    local player = Player(playerId)

    -- If player not found, nothing to do
    if player == nil then
        return
    end

    -- Get the player's party
    local party = player:getParty()

    -- If not in party, nothing to do
    if party == nil then
        local message = 'You are not in a party'
```



```

        player:sendTextMessage(MESSAGE_STATUS_CONSOLE_RED, message)

    return
end

-- Get party's leader
local partyLeader = party:getLeader()

-- If should assert party's leadership player is not the party's leader,
-- nothing to do
if ASSERT_PARTY_LEADERSHIP and partyLeader ~= player then
    local message = 'Only the party leader can use this command'

    player:sendTextMessage(MESSAGE_STATUS_CONSOLE_RED, message)

    return
end

-- Get target member by name
local targetMember = Player(memberName)

-- If target member not found, nothing to do
if targetMember == nil then
    local message = 'Target player not found'

    player:sendTextMessage(MESSAGE_STATUS_CONSOLE_RED, message)

    return
end

-- A player can't remove itself (design decision)
if targetMember == player then
    local message = 'You can\'t remove yourself from party with this command'

    player:sendTextMessage(MESSAGE_STATUS_CONSOLE_RED, message)

    return
end

-- Can't remove the party's leader (will only have effect if the ASSERT_PARTY_LEADERSHIP
-- is false, otherwise it would have already failed in the previous check, since
-- player == partyLeader at this point when ASSERT_PARTY_LEADERSHIP is true)
if targetMember == partyLeader then
    local message = 'You can\'t remove the party leader'

    player:sendTextMessage(MESSAGE_STATUS_CONSOLE_RED, message)

    return
end

-- Check if target member is in party and, if it is, remove it
for _, partyMember in pairs(party:getMembers()) do
    if partyMember == targetMember then
        party:removeMember(targetMember)
    end
end

```

```

        return
    end
end

-- This will only execute if the target member is a valid player in the server
-- but is not in the party of the command issuer
local message = 'Target player is not in your party'

player:sendTextMessage(MESSAGE_STATUS_CONSOLE_RED, message)
end

-- Process !q3 <member name> command
-- Always returns false to not echo the command back to the player
function onSay(player, words, param)
    -- Check if the command has the needed argument
    if #param == 0 then
        local message = 'Usage: !q3 <member name>'

        player:sendTextMessage(MESSAGE_STATUS_CONSOLE_RED, message)

        return false
    end

    -- Get the ID of the issuing player
    local playerId = player:getId()

    -- Get the player object for the player that has name equals to the
    -- command's argument
    local targetMember = Player(param)

    -- Check if player exists
    if targetMember == nil then
        local message = 'Target player not found'

        player:sendTextMessage(MESSAGE_STATUS_CONSOLE_RED, message)

        return false
    end

    -- Get the member name from the player's object
    local memberName = targetMember:getName()

    -- Try to remove the member from party
    removeMemberFromPlayerParty(playerId, memberName)

    return false
end
end

```

Configure the `ASSERT_PARTY_LEADERSHIP`. To allow only party leaders to kick party members, set it to `true`. Otherwise, set it to `false`

Restart the server and, in game, while in party, use as follow:

```
!q3 <member name>
```

Question 4

Assume all method calls work fine. Fix the memory leak issue in below method

```
void Game::addItemToPlayer(const std::string& recipient, uint16_t itemId)
{
    Player* player = g_game.getPlayerByName(recipient);

    if (!player) {
        player = new Player(nullptr);

        if (!IOLoginData::loadPlayerByName(player, recipient)) {
            return;
        }
    }

    Item* item = Item::CreateItem(itemId);
    if (!item) {
        return;
    }

    g_game.internalAddItem(player->getInbox(), item, INDEX_WHEREEVER, FLAG_NOLIMIT);

    if (player->isOffline()) {
        IOLoginData::savePlayer(player);
    }
}
```

Solution

There are 3 solutions, as follow

The bad:

```
// Add item by ID to player by name
void Game::addItemToPlayer(const std::string& recipient, uint16_t itemId)
{
    // Get player by name
    Player* player = g_game.getPlayerByName(recipient);

    // If can't find the player, try to load it from database
    if (!player) {
        player = new Player(nullptr);

        if (!IOLoginData::loadPlayerByName(player, recipient)) {
            delete player;

            return;
        }
    }

    // Create item by ID
    Item* item = Item::CreateItem(itemId);

    if (!item) {
        delete player;
    }
}
```

```

        return;
    }

    // Send item to player's inbox
    g_game.internalAddItem(player->getInbox(), item, INDEX_WHEREEVER, FLAG_NOLIMIT);

    // If player is offline, save it
    if (player->isOffline()) {
        IOLoginData::savePlayer(player);
    }

    // Clean up
    delete item;
    delete player;
}

```

The ugly:

```

// Add item by ID to player by name
void Game::addItemToPlayer(const std::string& recipient, uint16_t itemId)
{
    // Get player by name
    Player* player = g_game.getPlayerByName(recipient);

    // If can't find the player, try to load it from database
    if (!player) {
        player = new Player(nullptr);

        if (!IOLoginData::loadPlayerByName(player, recipient)) {
            goto defer;
        }
    }

    // Create item by ID
    Item* item = Item::CreateItem(itemId);

    if (!item) {
        goto defer;
    }

    // Send item to player's inbox
    g_game.internalAddItem(player->getInbox(), item, INDEX_WHEREEVER, FLAG_NOLIMIT);

    // If player is offline, save it
    if (player->isOffline()) {
        IOLoginData::savePlayer(player);
    }

    // Clean up
defer:
    delete item;
    delete player;
}

```

The beauty:

```
// Add item by ID to player by name
void Game::addItemToPlayer(const std::string& recipient, uint16_t itemId)
{
    // Get player by name
    std::unique_ptr<Player> pPlayer = std::unique_ptr<Player>(g_game.getPlayerByName(recipient));

    // If can't find the player, try to load it from database
    if (!pPlayer) {
        pPlayer = std::make_unique<Player>(nullptr);

        if (!IOLoginData::loadPlayerByName(pPlayer.get(), recipient)) {
            return;
        }
    }

    // Create item by ID
    std::unique_ptr<Item> pItem = std::unique_ptr<Item>(Item::CreateItem(itemId));

    if (!pItem) {
        return;
    }

    // Send item to player's inbox
    g_game.internalAddItem(pPlayer->getInbox(), pItem.get(), INDEX_WHEREEVER, FLAG_NOLIMIT);

    // If player is offline, save it
    if (pPlayer->isOffline()) {
        IOLoginData::savePlayer(pPlayer.get());
    }
}
```

Question 5

Vortex shader effect

Solution

Result video: https://youtu.be/W_D37NWqfZI

First, let's add a shader entry to the `<client root>\modules\game_shaders\shaders.lua` inside the `MAP_SHADERS` table:

```
{
    name = 'Map - Question 5',
    frag = 'shaders/fragment/q5.frag',
    tex1 = 'images/q5'
}
```

This makes use of a vortex texture, which will be a `q5.png` file put inside `<client root>\modules\game_shaders\images\q5.png`

Also we will need to create a `q5.frag` file inside `<client root>\modules\game_shaders\shaders\fragment\q5.frag` with the content:

```
// DISCLAIMER
//
```

```

// I am no expert at shaders and it is my first attempt to create such effect. So,
// as long as I replicated the desired effect, it is probably not the right way of
// achieving it. I hope my efforts are taken into account (laughing with a tear)
//
// The vortex's texture has its width and height as multiples of the tileSize. It
// is 96x64 pixels, meaning a 3x2 tileset with 32x32 pixels each tile. This info is
// kinda hardcoded in this code for the sake of simplicity, but all these values
// can be passed to the shader through uniforms to make the shader more versatile
// and value independent

uniform sampler2D u_Tex0;           // Game texture
uniform sampler2D u_Tex1;           // Vortex textures
varying vec2      v_TexCoord;       // Texture coordinates
uniform float      u_Time;           // Ellapsed time

const int          tileSize         = 32;      // Tile size, used to subdivide the game screen
const vec2         tileOffset      = vec2(8); // Tile offset to get the player correct position on
screen
const ivec2        spritesCount    = ivec2(3); // Vortex's sprites count
const int          radius          = 3;        // Radius of the vortex effect, from player position

vec2               middleTile;         // Will hold the middle tile, aka player position

// Check if tile is the bottom part of the vortex
bool isVortexBottom(ivec2 tile)
{
    // Do not consider the player position to have a vortex
    if (tile == middleTile)
    {
        return false;
    }

    // Check if the tile is within the effect radius
    return (distance(tile, middleTile) <= radius);
}

// Check if tile is the upper part of the vortex
bool isVortexTop(ivec2 tile)
{
    // It is, if the bottom tile is the vortex bottom
    return isVortexBottom(tile + ivec2(0, 1));
}

// Check if the vortex is visible based on time, just to not show all at once or
// at the same time
bool isVortexVisible(ivec2 tile)
{
    float c = cos(tile.x * tile.y * (u_Time + 1337) / 100);

    return (c > -0.35 && c < 0.25);
}

```

```

// Get a "random" vortex row to draw, to animate it
int getVortexRow(ivec2 tile)
{
    return ((tile.x * 1117 + tile.y * 73 * (int(u_Time) % 5)) % spritesCount.y);
}

// Get a "random" vortex sprite to draw, to animate it
int getVortexIndex(ivec2 tile)
{
    return (int(tile.x * 73 + tile.y * 1117 * u_Time / 1000) % spritesCount.x);
}

void main(void)
{
    // Store game and vortex textures' size
    ivec2 gameSize    = textureSize(u_Tex0, 0);
    ivec2 vortexSize  = textureSize(u_Tex1, 0);

    // Calculate how many tiles fit in the game screen
    ivec2 tiles       = gameSize / tileSize;

    // Get the middle screen tile, indexing from 0
    middleTile        = tiles / 2 - 1;

    // Convert from bottom left (0, 0) with (0..1, 0..1) space to
    // top left with (0..gameSize.x, 0..gameSize.y) space
    ivec2 coord        = (vec2(0.0, 1.0) - v_TexCoord * vec2(-1.0, 1.0)) * gameSize + tileOffset;

    // Get the current working tile and its bottom tile
    ivec2 tile          = coord / tileSize;
    ivec2 tileBottom    = tile + ivec2(0, 1);

    // Get the remaining pixels inside the tile to sample from the vortex texture
    ivec2 offset        = coord % tileSize;

    // Get the current color to start blending
    vec4 gameColor      = texture2D(u_Tex0, v_TexCoord);

    // Check if the current tile is part of the effect
    bool isTileTop      = isVortexTop(tile);
    bool isTileBottom   = isVortexBottom(tile);

    // Already set the output color as the current read color
    gl_FragColor        = vec4(gameColor.rgb, 1.0);

    // If the tile is not part of the effect, skip it
    if (!isTileTop && !isTileBottom)
    {
        return;
    }

    // If the tile is part of the effect, is vortex's bottom part, sample from
    // vortex's texture and, if the sampled color is not transparent, blend it into
    // the output color

```

```

if (isTileBottom && isVortexVisible(tile))
{
    // Get a "random" sprite index for the vortex's texture
    int row = getVortexRow(tile);
    int index = getVortexIndex(tile);

    // Sample from the vortex's texture
    vec4 color = texture2D(u_Tex1, (vec2(index * tileSize, row * tileSize * 2 + tileSize) +
offset) / vortexSize);

    // If the color is not transparent, blend it to the output color
    if (color.a > 0)
    {
        gl_FragColor += color;
    }
}

// If the tile is part of the effect, is vortex's upper part, sample from
// vortex's texture and, if the sampled color is not transparent, blend it into
// the output color
if (isTileTop && isVortexVisible(tileBottom))
{
    // Get a "random" sprite index for the vortex's texture
    int row = getVortexRow(tileBottom);
    int index = getVortexIndex(tileBottom);

    // Sample from the vortex's texture
    vec4 color = texture2D(u_Tex1, (vec2(index * tileSize, row * tileSize * 2) + offset) /
vortexSize);

    // If the color is not transparent, blend it to the output color
    if (color.a > 0)
    {
        gl_FragColor += color;
    }
}

// Ensure the alpha is opaque
gl_FragColor.a = 1.0;
}

```

To test/spawn this effect, let's add a GUI button

Create a folder named `game_q5` inside `<client root>\mods`

Add 2 files to it:

`<client root>\mods\game_q5\q5.otmod:`

```

Module
  name: game_q5
  description: Interview Question 5
  author: Wagner Barongello
  website: https://github.com/barongello
  sandboxed: true
  scripts: [ q5 ]

```



```
@onLoad: init()
@onUnload: terminate()
```

<client root>\mods\game_q5\q5.lua:

```
-- Configurations
local EFFECT_DURATION = 5000

-- References
local q5Button = nil

-- Events
local toggleEvent = nil

-- Run on game start
function online()
    -- If can find the toggle button, show it
    if q5Button ~= nil then
        q5Button:show()
    end
end

-- Run on game end
function offline()
    resetEffect()
end

-- Module initialization on load
function init()
    -- Connect the signals/callbacks
    connect(g_game, {
        onGameStart = online,
        onGameEnd = offline
    })

    -- Create the trigger button, get its reference and set it as not toggled
    q5Button = modules.client_topmenu.addRightGameToggleButton('q5Button', tr('Q5'),
'/images/topbuttons/q5', toggle)

    -- If can find the toggle button, set it to off
    if q5Button ~= nil then
        q5Button:setOn(false)
    end

    -- If initialized after game started, call the callback manually
    if g_game.isOnline() then
        online()
    end
end
```

```

-- Module termination on unload
function terminate()
    -- Remove the signals/callbacks
    disconnect(g_game, {
        onGameStart = online,
        onGameEnd = offline
    })

    -- Reset the effect
    resetEffect()

    -- If can find the toggle button, destroy it
    if q5Button ~= nil then
        q5Button:destroy()

        q5Button = nil
    end
end

-- Toggle function of the trigger button
function toggle()
    -- If q5Button is not found, this should not have been called
    if q5Button == nil then
        return
    end

    -- If the toggle state is on, reset the effect. Otherwise, start it
    if q5Button:isOn() then
        resetEffect()
    else
        startEffect()
    end
end

-- Start effect and toggle button on
function startEffect()
    -- If can't find the map, nothing to do
    local map = modules.game_interface.getMapPanel()

    if map == nil then
        return
    end

    -- Set map shader
    map:setShader('Map - Question 5')

    -- Schedule an event to stop the effect
    toggleEvent = scheduleEvent(resetEffect, EFFECT_DURATION)

    -- If can find the toggle button, toggle it on
    if q5Button ~= nil then
        q5Button:setOn(true)
    end
end

```

```

-- Stop effect and toggle button off
function resetEffect()
    -- If there is a scheduled event, remove it
    if toggleEvent ~= nil then
        removeEvent(toggleEvent)

        toggleEvent = nil
    end

    -- If can find map, reset map shader
    local map = modules.game_interface.getMapPanel()

    if map ~= nil then
        map:setShader('Map - Default')
    end

    -- If can find toggle button, set it off
    if q5Button ~= nil then
        q5Button:setOn(false)
    end
end
end

```

Add a 16x16 PNG file named q5.png to `data\images\topbuttons\q5.png`

Edit the <client root>\modules\game_interface\interface.otmod and add - game_q5 under the load-later section

Reload the client and click on the Q5 button on the top right corner

Question 6

Dash shader effect

Solution

Result video: <https://youtu.be/ch75SvQyw1U>

First, let's add a shader entry to the <client root>\modules\game_shaders\shaders.lua inside the MAP_SHADERS table:

```

{
    name = 'Map - Question 6',
    frag = 'shaders/fragment/q6_map.frag',
    tex1 = 'images/q6'
}

```

This makes use of a player texture, which will be a q6.png file put inside <client root>\modules\game_shaders\images\q6.png

Let's add another shader entry to the <client root>\modules\game_shaders\shaders.lua inside the OUTFIT_SHADERS table:

```

{
    name = 'Outfit - Question 6',
    frag = 'shaders/fragment/q6_outfit.frag',
}

```

```
useFramebuffer = true
}
```

Also we will need to create a `q6_map.frag` file inside `<client root>\modules\game_shaders\shaders\fragment\q6_map.frag` with the content:

```
// DISCLAIMER
//
// I am no expert at shaders and it is my first attempt to create such effect. So,
// as long as I replicated the desired effect, it is probably not the right way of
// achieving it. I hope my efforts are taken into account (laughing with a tear)
//
// Spent a lot of time trying to pass down to the shader the current player's
// texture, but failed. All my attempts resulted in empty textures (all black).
// So I just copied the player's silhouette and made a PNG texture for it. Also
// I decided to space it more from the player's position

uniform sampler2D u_Tex0;           // Game texture
uniform sampler2D u_Tex1;           // Dash texture
varying vec2      v_TexCoord;       // Texture coordinates
uniform float      u_Time;           // Ellapsed time
uniform float      u_MapShaderTime; // Shader ellapsed time
uniform int         u_PlayerDirection; // Player direction
uniform int         u_PlayerStepDuration; // Player direction step duration

const int          tileSize = 32;    // Tile size, used to subdivide the game
screen
const vec2          tileOffset = vec2(8); // Tile offset to get the player correct
position on screen
const int           trailSize = 4;    // How many trails will have
const float         fade      = 1.0 / (trailSize + 1); // Fade step for trails
vec2                middleTile;       // Will hold the middle tile, aka player
position

void main(void)
{
    // Store game and dash textures' size
    ivec2 gameSize      = textureSize(u_Tex0, 0);
    ivec2 dashSize      = textureSize(u_Tex1, 0);

    // Calculate how many tiles fit in the game screen
    ivec2 tiles          = gameSize / tileSize;

    // Get the middle screen tile, indexing from 0
    middleTile           = tiles / 2 - 1;

    // Convert from bottom left (0, 0) with (0..1, 0..1) space to
    // top left with (0..gameSize.x, 0..gameSize.y) space
    ivec2 coord          = (vec2(0.0, 1.0) - v_TexCoord * vec2(-1.0, 1.0)) * gameSize + tileOffset;

    // Get the current working tile and its bottom tile
    ivec2 tile           = coord / tileSize;
```

```

// Get the remaining pixels inside the tile to sample from the vortex texture
ivec2 offset          = coord % tileSize;

// Start with the current color
gl_FragColor          = texture2D(u_Tex0, v_TexCoord);

// Initialize the trail direction vector
ivec2 trailDirection = ivec2(0);

// Update the trail direction vector based on player's direction
// 0 - North
// 1 - East
// 2 - South
// 3 - West
// The trail direction will be opposite to the player's direction
if (u_PlayerDirection == 0)
{
    trailDirection.y = 1;
}
else if (u_PlayerDirection == 1)
{
    trailDirection.x = -1;
}
else if (u_PlayerDirection == 2)
{
    trailDirection.y = -1;
}
else if (u_PlayerDirection == 3)
{
    trailDirection.x = 1;
}

// Initialize the trails count to draw
int trails = trailSize;

// If player direction step duration is not zero, let's get a better trails count
if (u_PlayerStepDuration > 0)
{
    trails = int(clamp(750.0 * u_MapShaderTime / u_PlayerStepDuration, 0, trailSize));
}

// Iterate over all trail tiles to determine if it should be drawn now or not
for (int i = 0; i < trails; ++i)
{
    // Get the target trail tile
    ivec2 targetTile = middleTile + trailDirection * (i + 1);

    // If the target tile is the current tile being drawn
    if (tile == targetTile)
    {
        // Get dash pixel color
        vec4 sample = texture2D(u_Tex1, (vec2(u_PlayerDirection * tileSize, 1.0) + offset) /
dashSize);

        // If it is visible, add transparency to it based on the distance from the player

```

```

// The farther, the more transparent
if (sample.a > 0.5)
{
    float alpha = 1.0 - i * fade;

    gl_FragColor = vec4(mix(gl_FragColor.rgb, vec3(1.0, 0.0, 0.0), alpha), 1.0);
}
}
}
}

```

Also we will need to create a `q6_outfit.frag` file inside `<client root>\modules\game_shaders\shaders\fragment\q6_outfit.frag` with the content:

```

// DISCLAIMER
//
// I am no expert at shaders and it is my first attempt to create such effect. So,
// as long as I replicated the desired effect, it is probably not the right way of
// achieving it. I hope my efforts are taken into account (laughing with a tear)
//
// It was based on the already existing Outfit - Outline fragment shader from the
// Mehah's OT Client

uniform sampler2D u_Tex0;           // Player texture
varying vec2      v_TexCoord;      // Texture coordinates
uniform float     u_Time;          // Ellapsed time

const float       offset = 1.0 / 64.0; // Offset to check for empty pixels

void main()
{
    // Sample the current pixel
    vec4 color = texture2D(u_Tex0, v_TexCoord);

    // If it is more opaque than transparent, just draw it
    if (color.a > 0.5)
    {
        gl_FragColor = color;
    }
    // Otherwise, check if it is opaque and all four neighbours are transparent. If
    // it is not opaque or all four neighbours are transparent, draw it. Otherwise,
    // paint it red
    else
    {
        float a =
            texture2D(u_Tex0, vec2(v_TexCoord.x + offset, v_TexCoord.y          )).a +
            texture2D(u_Tex0, vec2(v_TexCoord.x          , v_TexCoord.y - offset)).a +
            texture2D(u_Tex0, vec2(v_TexCoord.x - offset, v_TexCoord.y          )).a +
            texture2D(u_Tex0, vec2(v_TexCoord.x          , v_TexCoord.y + offset)).a;

        if (color.a < 1.0 && a > 0.0)
        {
            gl_FragColor = vec4(1.0, 0.0, 0.0, 1.0);
        }
    }
}

```

```

    }
    else
    {
        gl_FragColor = color;
    }
}
}

```

We introduced three new uniforms `u_MapShaderTime`, `u_PlayerDirection` and `u_PlayerStepDuration` inside `q6_map.frag`, so let's add it to the client source code

Inside the `setupMapShader` in `<client root>\src\framework\graphics\shadermanager.cpp`, let's add these three lines along the other `bindUniformLocation`:

```

shader->bindUniformLocation(MAP_SHADER_TIME, "u_MapShaderTime");
shader->bindUniformLocation(PLAYER_DIRECTION, "u_PlayerDirection");
shader->bindUniformLocation(PLAYER_STEP_DURATION, "u_PlayerStepDuration");

```

Also let's add the three new enums `MAP_SHADER_TIME`, `PLAYER_DIRECTION` and `PLAYER_STEP_DURATION` in `<client root>\src\framework\graphics\shadermanager.h`, right below the `MAP_GLOBAL_COORD`:

```

MAP_SHADER_TIME = 18,
PLAYER_DIRECTION = 19,
PLAYER_STEP_DURATION = 20,

```

Now we need to update these values before calling the shader. Let's modify the `registerEvents` function in `<client root>\src\client\mapview.cpp` and add these lines right before setting the shader program:

```

m_shader->setUniformValue(ShaderManager::MAP_SHADER_TIME, g_clock.seconds() - m_shaderStartTime);

LocalPlayerPtr player          = g_game.getLocalPlayer();
Otc::Direction playerDirection = (player ? player->getDirection() :
Otc::Direction::InvalidDirection);
uint16_t          playerStepDuration = (player ? player->getStepDuration(true, playerDirection) : 0);

m_shader->setUniformValue(ShaderManager::PLAYER_DIRECTION, playerDirection);
m_shader->setUniformValue(ShaderManager::PLAYER_STEP_DURATION, playerStepDuration);

```

At this point the local player should not be `nullptr`, but... Is better to be safe than sorry

Also we will need to include `localplayer.h`, so we should add this line on the top of the `<client root>\src\client\mapview.cpp`, right below `#include "lightview.h"`:

```

#include "localplayer.h"

```

These modifications make use of the member `m_shaderStartTime`, so let's add it

Inside `<client root>\src\client\mapview.h`, let's add this line right below `Position m_shaderPosition;`:

```

float    m_shaderStartTime;

```

Back to `<client root>\src\client\mapview.cpp`, let's modify the `setShader` function and add this line right below `m_shaderPosition = getCameraPosition();`:

```
m_shaderStartTime = g_clock.seconds();
```

So, now, every time a shader is set, its timer is reseted

To test/spawn this effect, let's add a GUI button

Create a folder named `game_q6` inside `<client root>\mods`

Add 2 files to it:

```
<client root>\mods\game_q6\q6.otmod:
```

```
Module
  name: game_q6
  description: Interview Question 6
  author: Wagner Barongello
  website: https://github.com/barongello
  sandboxed: true
  scripts: [ q6 ]
  @onLoad: init()
  @onUnload: terminate()
```

```
<client root>\mods\game_q6\q6.lua:
```

```
-- Configurations
local EFFECT_STEPS = 5

-- References
local q6Button = nil

-- Events
local toggleEvent = nil

-- Run on game start
function online()
  -- If can find the toggle button, show it
  if q6Button ~= nil then
    q6Button:show()
  end
end

-- Run on game end
function offline()
  resetEffect()
end

-- Module initialization on load
function init()
  -- Connect the signals/callbacks
  connect(g_game, {
```



```

        onGameStart = online,
        onGameEnd = offline
    })

    -- Create the trigger button, get its reference and set it as not toggled
    q6Button = modules.client_topmenu.addRightGameToggleButton('q6Button', tr('Q6'),
'/images/topbuttons/q6', toggle)

    -- If can find the toggle button, set it off
    if q6Button ~= nil then
        q6Button:setOn(false)
    end

    -- If initialized after game started, call the callback manually
    if g_game.isOnline() then
        online()
    end
end

-- Module termination on unload
function terminate()
    -- Remove the signals/callbacks
    disconnect(g_game, {
        onGameStart = online,
        onGameEnd = offline
    })

    -- Reset the effect
    resetEffect()

    -- Destroy trigger button
    if q6Button ~= nil then
        q6Button:destroy()

        q6Button = nil
    end
end

-- Toggle function of the trigger button
function toggle()
    -- If q6Button is not found, this should not have been called
    if q6Button == nil then
        return
    end

    -- If the toggle state is on, reset the effect. Otherwise, start it
    if q6Button:isOn() then
        resetEffect()
    else
        startEffect()
    end
end

-- Start effect and toggle on
function startEffect()

```

```

-- If can't get the map, nothing to do
local map = modules.game_interface.getMapPanel()

if map == nil then
    return
end

-- If can't get the player, nothing to do
local player = g_game:getLocalPlayer()

if player == nil then
    return
end

-- Get player's actual position as target position and player direction
local position = player:getPosition()
local direction = player:getDirection()

-- Modify target position based on player's direction
-- 0 - North
-- 1 - East
-- 2 - South
-- 3 - West
if direction == 0 then
    position.y = position.y - EFFECT_STEPS
elseif direction == 1 then
    position.x = position.x + EFFECT_STEPS
elseif direction == 2 then
    position.y = position.y + EFFECT_STEPS
elseif direction == 3 then
    position.x = position.x - EFFECT_STEPS
end

-- Disable player walking animation
player:setDisableWalkAnimation(true)

-- Make player auto walk to the target position
player:autoWalk(position)

-- Set the dash shader for the map
map:setShader('Map - Question 6')

-- Set the outline shader for the player
player:setShader('Outfit - Question 6')

-- Get the effect duration
local duration = player:getStepDuration(true, direction) * EFFECT_STEPS

-- Schedule an event to stop the effect
toggleEvent = scheduleEvent(resetEffect, duration)

-- If can find the q6Button, toggle it on
if q6Button ~= nil then
    q6Button:setOn(true)
end

```

```

end

-- Toggle off and stop effect
function resetEffect()
    -- If the toggleEvent is going on, remove it
    if toggleEvent ~= nil then
        removeEvent(toggleEvent)

        toggleEvent = nil
    end

    -- If can get the map, reset the shader
    local map = modules.game_interface.getMapPanel()

    if map ~= nil then
        map:setShader('Map - Default')
    end

    -- If can get the player, reset the shader and re-enable walk animation
    local player = g_game:getLocalPlayer()

    if player ~= nil then
        player:setShader('Outfit - Default')

        player:setDisableWalkAnimation(false)
    end

    -- If can get the q6Button, set its toggle state off
    if q6Button ~= nil then
        q6Button:setOn(false)
    end
end
end

```

Add a 16x16 PNG file named q6.png to `data\images\topbuttons\q6.png`

Edit the `<client root>\modules\game_interface\interface.otmod` and add `- game_q6` under the `load-later` section

Reload the client and click on the Q6 button on the top right corner

Question 7

Dialog with jump button

Solution

Result video: <https://youtu.be/UFGY8DrLrQg>

Create a folder named `game_q7` inside `<client root>\mods`

Add 3 files to it:

```
<client root>\mods\game_q7\q7.otmod:
```

```

Module
  name: game_q7
  description: Interview Question 7
  author: Wagner Barongello

```

```
website: https://github.com/barongello
sandboxed: true
scripts: [ q7 ]
@onLoad: init()
@onUnload: terminate()
```

<client root>\mods\game_q7\q7.otui:

```
MainWindow
  id: q7Window
  !text: tr('Q7')
  size: 320 240
  @onEscape: toggle()

  Button
    id: jumpButton
    !text: tr('JUMP')
    width: 50
    @onClick: jumpButtonClick()
```

<client root>\mods\game_q7\q7.lua:

```
-- Configurations
local JUMP_BUTTON_AUTO_MOVE_TIMEOUT = 100
local JUMP_BUTTON_STEP_X            = 10
local JUMP_BUTTON_SPACING_X        = 10
local JUMP_BUTTON_SPACING_TOP      = 30
local JUMP_BUTTON_SPACING_BOTTOM   = 10

-- References
local q7Window = nil
local q7Button = nil
local jumpButton = nil

-- Events
local autoMoveEvent = nil

-- Run on game start
function online()
  -- If trigger button is found, show it
  if q7Button ~= nil then
    q7Button:show()
  end
end

-- Run on game end
function offline()
  resetWindow()
end
```

```

-- Module initialization on load
function init()
    -- Connect the signals/callbacks
    connect(g_game, {
        onGameStart = online,
        onGameEnd = offline
    })

    -- Create the window, get its reference and hide it
    q7Window = g_ui.displayUI('q7', modules.game_interface.getRightPanel())

    -- If window is found, hide it
    if q7Window ~= nil then
        q7Window:hide()
    end

    -- Create the trigger button, get its reference and set it as not toggled
    q7Button = modules.client_topmenu.addRightGameToggleButton('q7Button', tr('Q7'),
'/images/topbuttons/q7', toggle)

    -- If trigger button is found, set its toggle state to off
    if q7Button ~= nil then
        q7Button:setOn(false)
    end

    -- Get the reference to the jumpButton
    jumpButton = q7Window:getChildById('jumpButton')

    -- If initialized after game started, call the callback manually
    if g_game.isOnline() then
        online()
    end
end

-- Module termination on unload
function terminate()
    -- Remove the signals/callbacks
    disconnect(g_game, {
        onGameStart = online,
        onGameEnd = offline
    })

    -- Reset window
    resetWindow()

    -- If window is found, destroy it
    if q7Window ~= nil then
        q7Window:destroy()

        q7Window = nil
    end

    -- If trigger button is found, destroy it
    if q7Button ~= nil then

```

```

        q7Button:destroy()

        q7Button = nil
    end
end

-- Toggle function of the trigger button
function toggle()
    -- If q7Button is not found, this should not have been called
    if q7Button == nil then
        return
    end

    -- If the toggle button is on, stop window. Otherwise, start it
    if q7Button:isOn() then
        resetWindow()
    else
        startWindow()
    end
end

-- Start window and button states
function startWindow()
    -- If window, trigger button or jump button can't be found, nothing to do
    if q7Window == nil or q7Button == nil then
        return
    end

    -- Toggle on
    q7Button:setOn(true)

    -- Show window, bring it to front and focus it
    q7Window:show()
    q7Window:raise()
    q7Window:focus()

    -- Randomize jumpButton start position
    jumpButtonClick()

    -- Set the timer to auto move the jumpButton
    autoMoveEvent = periodicalEvent(jumpButtonAutoMove, nil, JUMP_BUTTON_AUTO_MOVE_TIMEOUT)
end

-- Reset window and trigger button states
function resetWindow()
    -- If there is a periodical event, remove it
    if autoMoveEvent ~= nil then
        removeEvent(autoMoveEvent)

        autoMoveEvent = nil
    end

    -- If q7Window is found, hide it
    if q7Window ~= nil then
        q7Window:hide()
    end
end

```

```

end

-- If q7Button is found, set its toggle state to off
if q7Button ~= nil then
    q7Button:setOn(false)
end
end

-- Send jumpButton back to the right side
function resetJumpButtonX()
    -- If window or jump button couldn't be find, nothing to do
    if q7Window == nil or jumpButton == nil then
        return
    end

    -- Get window and jump button current position
    windowPos = q7Window:getPosition()
    buttonPos = jumpButton:getPosition()

    -- Calculate the new jump button X position based on window position and size and jump button
    size
    buttonPos.x = windowPos.x + q7Window:getWidth() - jumpButton:getWidth() -
    JUMP_BUTTON_SPACING_X

    -- Set the jump button new position
    jumpButton:setPosition(buttonPos)
end

-- Send jumpButton to a random Y position
function resetJumpButtonY()
    -- If window or jump button couldn't be find, nothing to do
    if q7Window == nil or jumpButton == nil then
        return
    end

    -- Get window and jump button current position
    windowPos = q7Window:getPosition()
    buttonPos = jumpButton:getPosition()

    -- Get the maximum Y that the jump button can have and randomize it
    local maxY = q7Window:getHeight() - jumpButton:getHeight() - JUMP_BUTTON_SPACING_BOTTOM
    local randomY = math.random(JUMP_BUTTON_SPACING_TOP, maxY)

    -- Calculate the new jump button Y position based on window position and randomized Y
    buttonPos.y = windowPos.y + randomY

    -- Set the jump button new position
    jumpButton:setPosition(buttonPos)
end

-- Auto move jumpButton horizontally
function jumpButtonAutoMove()
    -- If window or jump button couldn't be find, nothing to do
    if q7Window == nil or jumpButton == nil then
        return
    end

```

```

end

-- Get window and jump button current position
windowPos = q7Window:getPosition()
buttonPos = jumpButton:getPosition()

-- Calculate jump button new X position based on its velocity
buttonPos.x = buttonPos.x - JUMP_BUTTON_STEP_X

-- If hit the left side, send it back to right side
if buttonPos.x <= windowPos.x + JUMP_BUTTON_SPACING_X then
    resetJumpButtonX()
end

-- Set the jump button new position
jumpButton:setPosition(buttonPos)
end

-- On click jumpButton, send it back to right side and randomize Y position
function jumpButtonClick()
    -- If jump button can't be found, nothing to do
    if jumpButton == nil then
        return
    end

    -- Reset jump button X position
    resetJumpButtonX()

    -- Randomize jump button Y position
    resetJumpButtonY()
end

```

Add a 16x16 PNG file named `q7.png` to ``data\images\topbuttons\q7.png`

Edit the `<client root>\modules\game_interface\interface.otmod` and add `- game_q7` under the `load-later` section

Reload the client and click on the `Q7` button on the top right corner

Final thoughts

I know that it is not good to connect to MySQL as `root`, but it is just a local test server. Never do it in production

Server messages that are being sent to the players have no support for translations in the branch I developed it, but it is a thing to pay attention in production (and that was probably already been addressed in your code base)

On `Question 3` it does not work if the player name has spaces. I tried it manually and even a `Player` instance reporting the right name with `player:getName()`, if you try to instantiate through `Player("name with spaces")`, it always returns `nil`. Probably a thing that will need to be addressed in production

On `Question 4`, deleting an already `nullptr` pointer is a `no-op`, so it is useless or does no harm, depending on the scenario (in the `bad` scenario, it is useless. In the `ugly` scenario, it does no harm). Also the "smart pointers" of the `beauty` scenario handle the pointer deletion when their destructors are called when they go out of the scope and their reference count reach 0

Questions `1`, `2`, `3`, `4` and `7` were the best ones for me, since I think that shaders are my Achilles' heel (in the past, the only things I have done with shaders were moving clouds and tinting sprites)

For the questions 5 and 6 I tried my best to achieve similar results, although I don't think they are done in the more "right way"

Specially for question 6, I tried, but couldn't figure out how to pass the current player's texture to the shader (I always ended up with all black textures), so I tried a workaround: create a new texture with 4 masks, one for each player's direction

This workaround introduced a new problem: since they are solid images, the transparency effect would be layered one over the another, making it ugly and very difficult to see that the desired result was achieved, so I decided to space them more, making one "shadow clone" per tile, instead of making them appear closer to the player as in the question's video

Hope that efforts are also taken into account ❤️

Also shaders could have been registered differently, but I added it to the shaders GUI to ease the development/testing process

The code comments are a tricky thing, since it depends more on the code conventions that the team is following and the preferred code flavor. Some will find it too much, some will think that every line should be commented, some will think that comments are needed only in not so obvious lines/functions. It is totally adjustable, I have no hard opinions over things like that, I go with what the code base and team demand/seem fit

Also I think that there are too much safety checks, to the point of being overkill. But we never know when the caller will change, or if it will be called from other places in the future, or if things were already sanitized before calling the next function, etc.. Always better to be safe than sorry (and is not that much of an overhead that is added to the final product)

While being on the sanitization topic, `string.format` is not a proper sanitization function to SQL queries (to not say that it is not a sanitization function at all, because it isn't). But, since we ensured that the parameter that players have control over is a number, it should cause no harms. But always good to have sanitization in mind when dealing with users' inputs

Made scripts configurable, like permissions, effects durations, speeds, spacings, etc.

Questions 1, 2, 3, 4 and 7 I have finished as I like in 2 days, then questions 5 and 6 took 2 days each 😅 I took my time to revisit every question, read it calmly again, re-check my sources and documentation, rest a bit to come back with a fresh mind. But I was on it full time for the entire week

That's all, I think. Hope you like it and come back with good news. I really, REALLY, **REALLY** want the job ❤️