map.cpp

```
#include "headers.hpp"
#include "map.hpp"


Map::Map(int dist_from_side, sf::RenderWindow& window) : window(window)
{
■sf::Image map_texture;
■map_texture.loadFromFile("sprites/map.png");

■width = map_texture.getSize().x;
■height = map_texture.getSize().y;

■
■data = new int[width * height];
■
■for(int i = 0; i < width; i++)
■■for (int j = 0; j < height; j++)
■■{
■■■if (map_texture.getPixel(i, j).r == 255)
■■■■data[i + j * width] = 1;
■■■else
■■■■data[i + j * width] = 0;
■■}



■map_texture.~Image();

■position.x = dist_from_side;
■position.y = HEIGHT - dist_from_side - cell_size * height;

■sky_color = sf::Color(70, 170, 255);
■ground_color = sf::Color(33, 43, 35);

■sky_tex.loadFromFile("sprites/sky.png");
■sky_sprite.setTexture(sky_tex);
■//sky_sprite.setTextureRect(sf::IntRect(0, 30, 1833, 460));
■sky_sprite.setScale(sky_scale, sky_scale);

}




int Map::getCell(int x, int y)
{
■return data[x + width * y];
}

void Map::drawMap()
{
■sf::RectangleShape cell(v2f(cell_size, cell_size));

■// cell.setOutlineColor(sf::Color(70, 70, 80));
```

map.cpp

```
■// cell.setOutlineThickness(1);

■sf::Color colors[]{
■■sf::Color::Black,
■■sf::Color::White
■};



■for (int i = 0; i < width; i++)
■{
■■for (int j = 0; j < height; j++)
■■{
■■■int color_index = getCell(i, j);
■■■cell.setFillColor(colors[color_index]);
■■■cell.setPosition(position.x + cell_size * i, position.y + cell_size
* j);
■■■window.draw(cell);
■■}
■}
}

void Map::drawPoint(float x, float y)
{
■sf::CircleShape dot(5);

■dot.setFillColor(sf::Color::Red);

■dot.setPosition(position.x + cell_size * x - 2.5f, position.y +
cell_size * y - 2.5f);
■
■window.draw(dot);
■■
}

void Map::drawSky()
{
■float sky_y = floor_level - 950;
■sky_sprite.setPosition(sky_offset - sky_width * sky_scale, sky_y);
■window.draw(sky_sprite);

■sky_sprite.setPosition(sky_offset, sky_y);
■window.draw(sky_sprite);

■//cout << "Floor Level: " << floor_level << "\n";
}

void Map::shiftSky(float offset)
{
■sky_offset += offset * sky_sensitivity;

■if(sky_offset > sky_width * sky_scale)
■■sky_offset -= sky_width * sky_scale;
```

map.cpp

```
■if (sky_offset < WIDTH - sky_width * sky_scale)
■■sky_offset += sky_width * sky_scale;
}

void Map::drawGround()
{
■//cout << width << " "<< height << "\n";
■float ground_height = HEIGHT - floor_level;

■sf::RectangleShape ground(v2f(WIDTH, ground_height));

■ground.setFillColor(ground_color);
■ground.setPosition(v2f(0, floor_level));
■window.draw(ground);
}
```