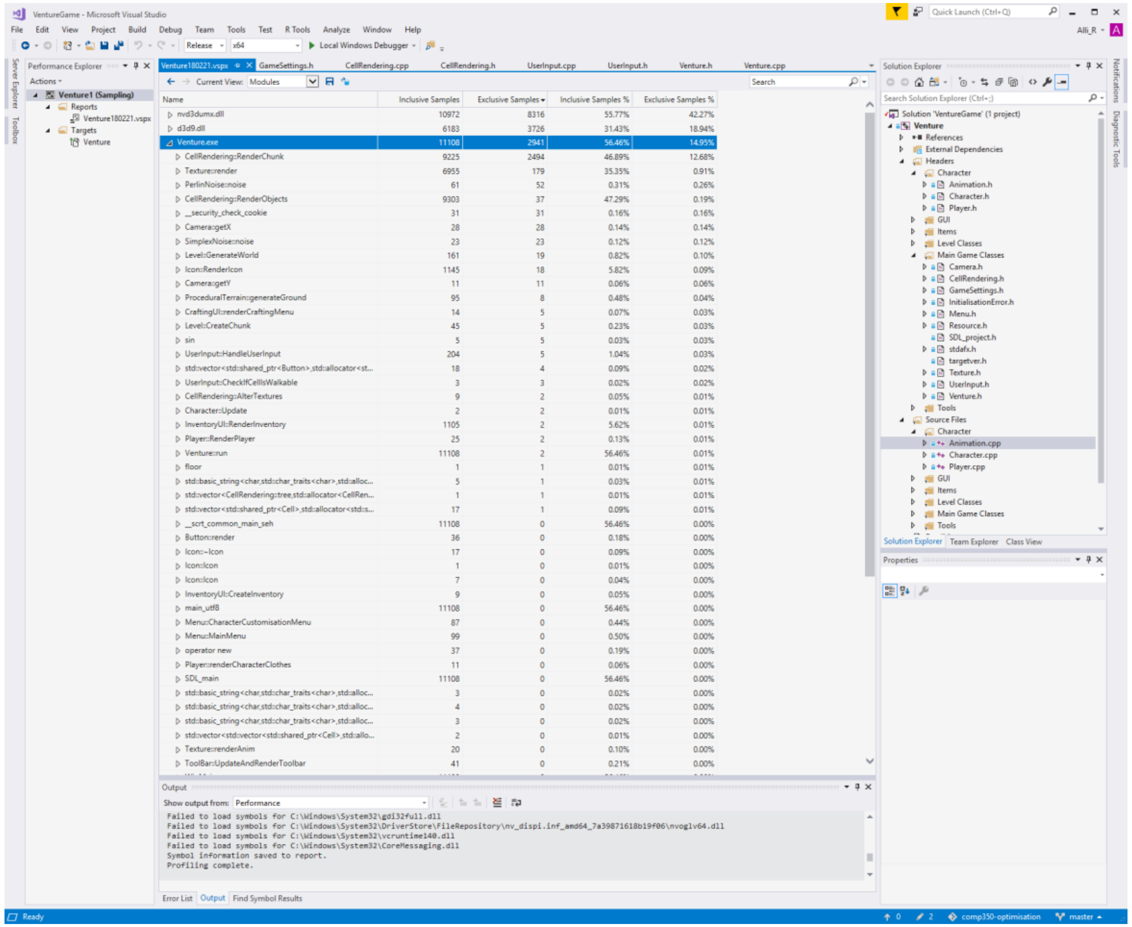
Optimisation task

Identifying the issues:



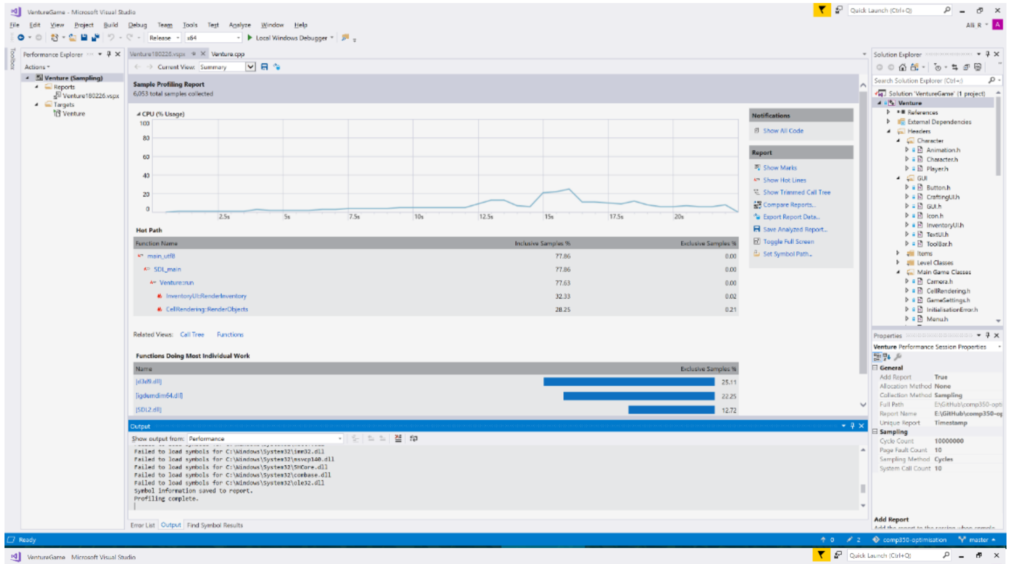
Name	Inclusive Samples	Exclusive Samples	Inclusive Samples %	Exclusive Samples %
▷ nvd3dumx.dll	10972	8316	55.77%	42.27%
▷ d3d9.dll	6183	3726	31.43%	18.94%
◿ Venture.exe	11108	2941	56.46%	14.95%
▷ CellRendering::RenderChunk	9225	2494	46.89%	12.68%
▷ Texture::render	6955	179	35.35%	0.91%
▷ PerlinNoise::noise	61	52	0.31%	0.26%
▷ CellRendering::RenderObjects	9303	37	47.29%	0.19%
▷ __security_check_cookie	31	31	0.16%	0.16%
▷ Camera::getX	28	28	0.14%	0.14%
▷ SimplexNoise::noise	23	23	0.12%	0.12%
▷ Level::GenerateWorld	161	19	0.82%	0.10%
▷ Icon::RenderIcon	1145	18	5.82%	0.09%
▷ Camera::getY	11	11	0.06%	0.06%
▷ ProceduralTerrain::generateGround	95	8	0.48%	0.04%
▷ CraftingUI::renderCraftingMenu	14	5	0.07%	0.03%
▷ Level::CreateChunk	45	5	0.23%	0.03%
▷ sin	5	5	0.03%	0.03%
▷ UserInput::HandleUserInput	204	5	1.04%	0.03%
▷ std::vector<std::shared_ptr<Button>,std::allocator<st...	18	4	0.09%	0.02%
▷ UserInput::CheckIfCellIsWalkable	3	3	0.02%	0.02%
▷ CellRendering::AlterTextures	9	2	0.05%	0.01%
▷ Character::Update	2	2	0.01%	0.01%
▷ InventoryUI::RenderInventory	1105	2	5.62%	0.01%
▷ Player::RenderPlayer	25	2	0.13%	0.01%
▷ Venture::run	11108	2	56.46%	0.01%
▷ floor	1	1	0.01%	0.01%
▷ std::basic_string<char,std::char_traits<char>,std::alloc...	5	1	0.03%	0.01%
▷ std::vector<CellRendering::tree,std::allocator<CellRen...	1	1	0.01%	0.01%
▷ std::vector<std::shared_ptr<Cell>,std::allocator<std::s...	17	1	0.09%	0.01%
▷ __scrt_common_main_seh	11108	0	56.46%	0.00%
▷ Button::render	36	0	0.18%	0.00%
▷ Icon::~Icon	17	0	0.09%	0.00%
▷ Icon::Icon	1	0	0.01%	0.00%
▷ Icon::Icon	7	0	0.04%	0.00%
▷ InventoryUI::CreateInventory	9	0	0.05%	0.00%
▷ main_utf8	11108	0	56.46%	0.00%
▷ Menu::CharacterCustomisationMenu	87	0	0.44%	0.00%
▷ Menu::MainMenu	99	0	0.50%	0.00%
▷ operator new	37	0	0.19%	0.00%
▷ Player::renderCharacterClothes	11	0	0.06%	0.00%
▷ SDL_main	11108	0	56.46%	0.00%
▷ std::basic_string<char,std::char_traits<char>,std::alloc...	3	0	0.02%	0.00%
▷ std::basic_string<char,std::char_traits<char>,std::alloc...	4	0	0.02%	0.00%
▷ std::basic_string<char,std::char_traits<char>,std::alloc...	3	0	0.02%	0.00%
▷ std::vector<std::vector<std::shared_ptr<Cell>,std::allo...	2	0	0.01%	0.00%
▷ Texture::renderAnim	20	0	0.10%	0.00%
▷ ToolBar::UpdateAndRenderToolbar	41	0	0.21%	0.00%

Output

Show output from: Performance

```
Failed to load symbols for C:\Windows\System32\gdi32full.dll
Failed to load symbols for C:\Windows\System32\DriverStore\FileRepository\nv_dispi.inf_amd64_7a39871618b19f06\nvoglv64.dll
Failed to load symbols for C:\Windows\System32\vcruntime140.dll
Failed to load symbols for C:\Windows\System32\CoreMessaging.dll
Symbol information saved to report.
Profiling complete.
```

This shows that a lot of time is dedicated to the rendering of the chunks.



Sample Profiling Report
6,053 total samples collected

Hot Path

Function Name	Inclusive Samples %	Exclusive Samples %
main_utf8	77.86	0.00
SDL_main	77.86	0.00
Venture::run	77.63	0.00
InventoryUI::RenderInventory	32.33	0.02
CellRendering::RenderObjects	28.25	0.21

Related Views: Call Tree Functions

Functions Doing Most Individual Work

Name	Exclusive Samples %
[d3d9.dll]	25.11
[igdumdim64.dll]	22.25
[SDL2.dll]	12.72

Venture Performance Session Properties

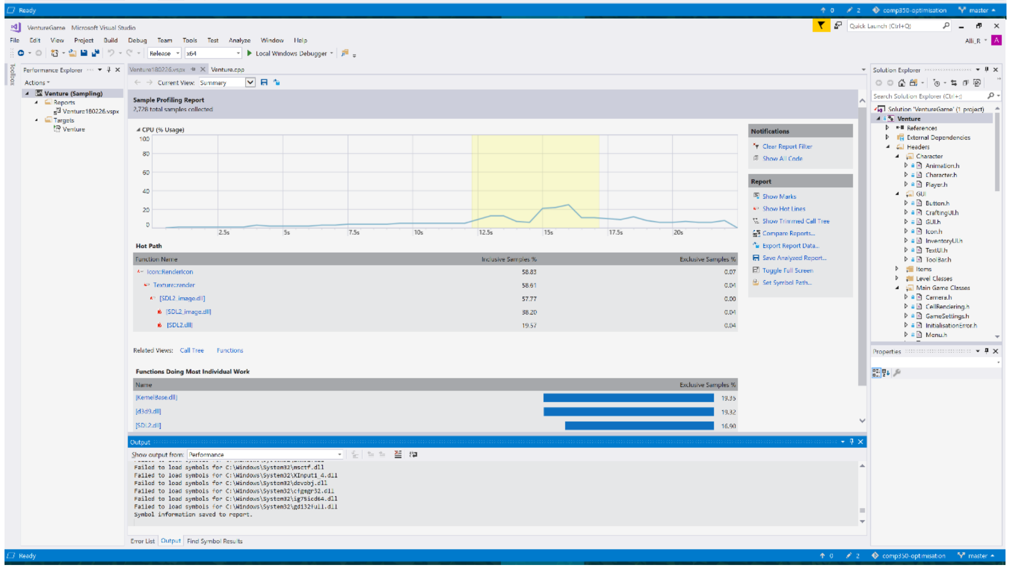
General	
Add Report	True
Allocation Method	None
Collection Method	Sampling
Full Path	E:\GitHub\comp350-opti
Report Name	E:\GitHub\comp350-op
Unique Report	Timestamp
Sampling	
Cycle Count	10000000
Page Fault Count	10
Sampling Method	Cycles
System Call Count	10

Output

Show output from: Performance

```
Failed to load symbols for C:\Windows\System32\imm32.dll
Failed to load symbols for C:\Windows\System32\msvcp140.dll
Failed to load symbols for C:\Windows\System32\SHCore.dll
Failed to load symbols for C:\Windows\System32\combase.dll
Failed to load symbols for C:\Windows\System32\ole32.dll
Symbol information saved to report.
Profiling complete.
```

This shows the spikes at where I was opening the inventory in the game.



Sample Profiling Report
2,728 total samples collected

Hot Path

Function Name	Inclusive Samples %	Exclusive Samples %
Icon::RenderIcon	58.83	0.07
Texture::render	58.61	0.04
[SDL2_image.dll]	57.77	0.00
[SDL2_image.dll]	38.20	0.04
[SDL2.dll]	19.57	0.04

Related Views: Call Tree Functions

Functions Doing Most Individual Work

Name	Exclusive Samples %
[KernelBase.dll]	19.35
[d3d9.dll]	19.32
[SDL2.dll]	16.90

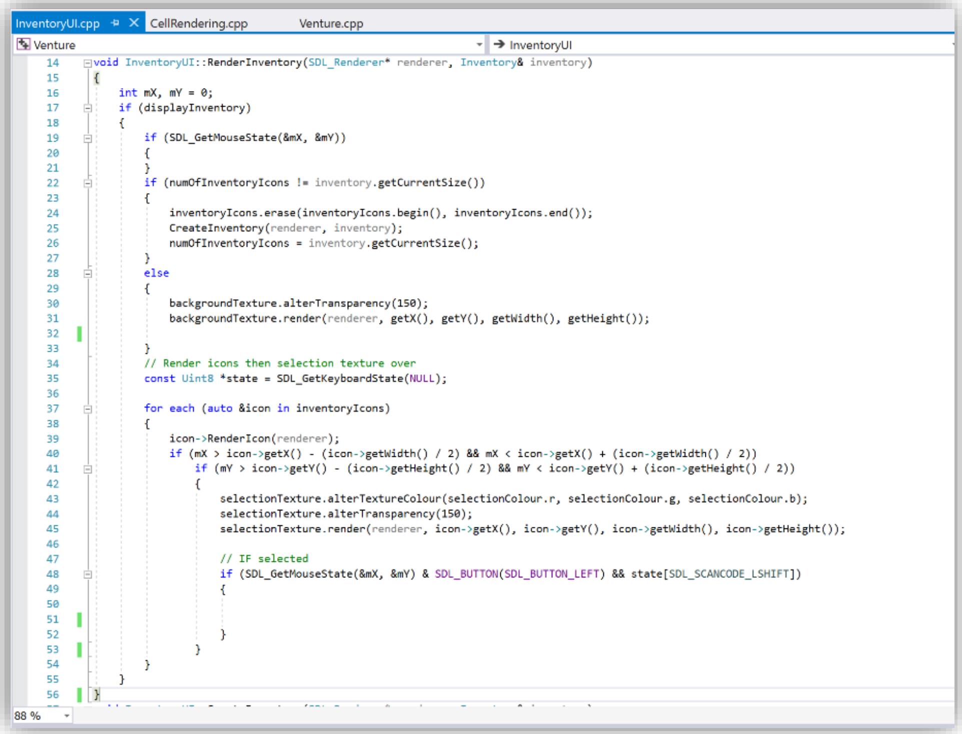
Output

Show output from: Performance

```
Failed to load symbols for C:\Windows\System32\msctf.dll
Failed to load symbols for C:\Windows\System32\XInput1_4.dll
Failed to load symbols for C:\Windows\System32\devobj.dll
Failed to load symbols for C:\Windows\System32\cfgmgr32.dll
Failed to load symbols for C:\Windows\System32\ig75icd64.dll
Failed to load symbols for C:\Windows\System32\gdi32full.dll
Symbol information saved to report.
```

This shows that the renderIcon function is causing a lot of cpu usage, when it shouldn’t be that costly to render some icons.

This is because every time the inventory is opened, it loops through all the icons to see if they have changed.

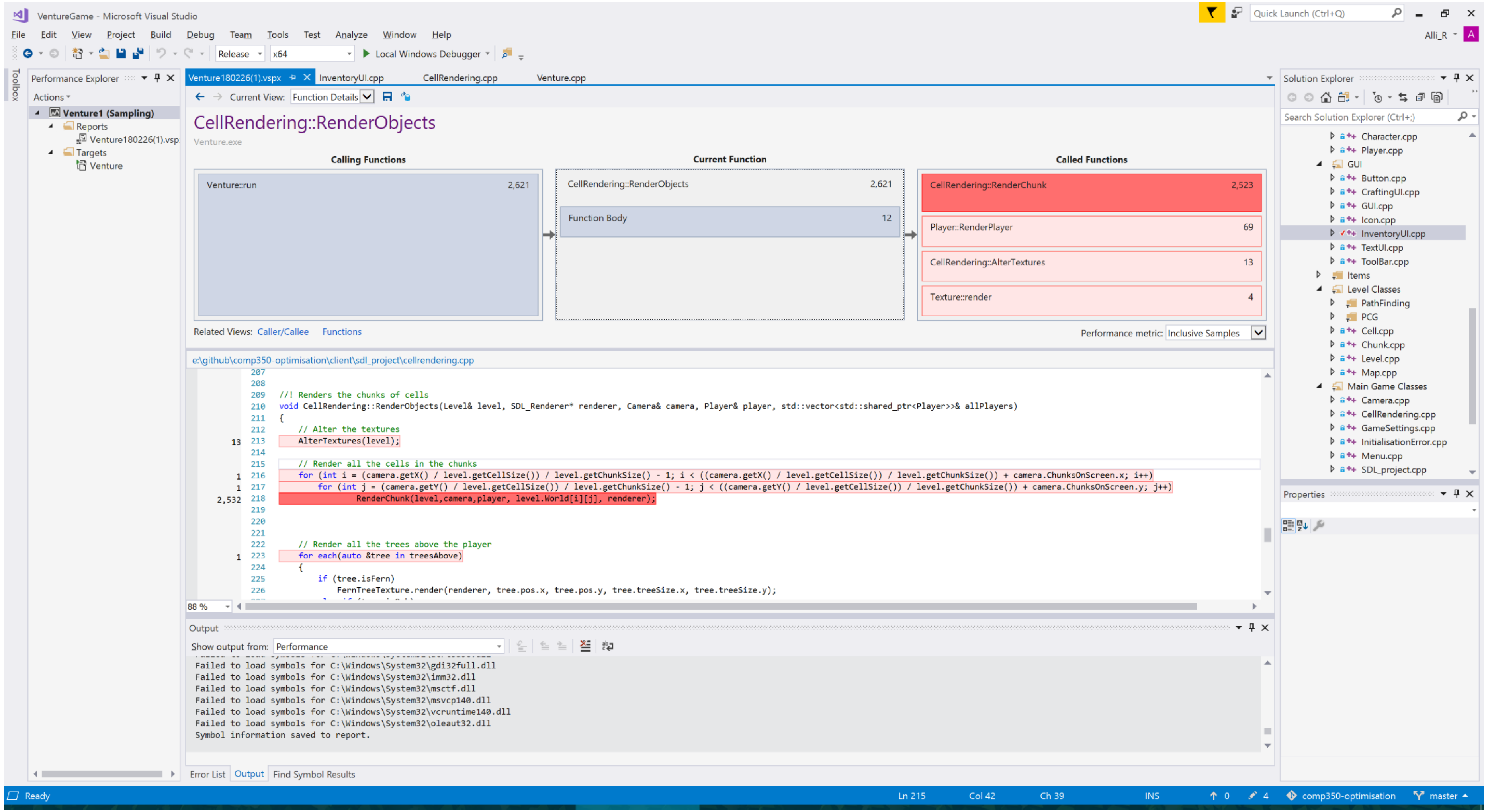


```
void InventoryUI::RenderInventory(SDL_Renderer* renderer, Inventory& inventory)
{
    int mX, mY = 0;
    if (displayInventory)
    {
        if (SDL_GetMouseState(&mX, &mY))
        {
        }
        if (numOfInventoryIcons != inventory.getCurrentSize())
        {
            inventoryIcons.erase(inventoryIcons.begin(), inventoryIcons.end());
            CreateInventory(renderer, inventory);
            numOfInventoryIcons = inventory.getCurrentSize();
        }
        else
        {
            backgroundTexture.alterTransparency(150);
            backgroundTexture.render(renderer, getX(), getY(), getWidth(), getHeight());

        }
        // Render icons then selection texture over
        const Uint8 *state = SDL_GetKeyboardState(NULL);

        for each (auto &icon in inventoryIcons)
        {
            icon->RenderIcon(renderer);
            if (mX > icon->getX() - (icon->getWidth() / 2) && mX < icon->getX() + (icon->getWidth() / 2))
                if (mY > icon->getY() - (icon->getHeight() / 2) && mY < icon->getY() + (icon->getHeight() / 2))
                {
                    selectionTexture.alterTextureColour(selectionColour.r, selectionColour.g, selectionColour.b);
                    selectionTexture.alterTransparency(150);
                    selectionTexture.render(renderer, icon->getX(), icon->getY(), icon->getWidth(), icon->getHeight());

                    // IF selected
                    if (SDL_GetMouseState(&mX, &mY) & SDL_BUTTON(SDL_BUTTON_LEFT) && state[SDL_SCANCODE_LSHIFT])
                    {


                    }
                }
        }
    }
}
```

This is the code that renders the icons in the toolbar, and this function is being called every time the inventory is opened.

Another issue is that there is not a time out on when the inventory button is opened, so if you hold the button down, it will open and close the inventory each frame.

One optimisation I will do is to remove the chunks that are outside the players range after a set time.



Current View: Function Details

CellRendering::RenderObjects

Venture.exe

Calling Functions		Current Function		Called Functions	
Venture::run	2,621	CellRendering::RenderObjects	2,621	CellRendering::RenderChunk	2,523
		Function Body	12	Player::RenderPlayer	69
				CellRendering::AlterTextures	13
				Texture::render	4

Related Views: Caller/Callee Functions

Performance metric: Inclusive Samples

e:\github\comp350-optimisation\client\sdl_project\cellrendering.cpp

```
      //! Renders the chunks of cells
      void CellRendering::RenderObjects(Level& level, SDL_Renderer* renderer, Camera& camera, Player& player, std::vector<std::shared_ptr<Player>>& allPlayers)
      {
          // Alter the textures
   13     AlterTextures(level);

          // Render all the cells in the chunks
    1     for (int i = (camera.getX() / level.getCellSize()) / level.getChunkSize() - 1; i < ((camera.getX() / level.getCellSize()) / level.getChunkSize()) + camera.ChunksOnScreen.x; i++)
    1         for (int j = (camera.getY() / level.getCellSize()) / level.getChunkSize() - 1; j < ((camera.getY() / level.getCellSize()) / level.getChunkSize()) + camera.ChunksOnScreen.y; j++)
2,532               RenderChunk(level,camera,player, level.World[i][j], renderer);



          // Render all the trees above the player
    1     for each(auto &tree in treesAbove)
          {
              if (tree.isFern)
                  FernTreeTexture.render(renderer, tree.pos.x, tree.pos.y, tree.treeSize.x, tree.treeSize.y);
```

88 %

Output

Show output from: Performance

```
Failed to load symbols for C:\Windows\System32\gdi32full.dll
Failed to load symbols for C:\Windows\System32\imm32.dll
Failed to load symbols for C:\Windows\System32\msctf.dll
Failed to load symbols for C:\Windows\System32\msvcp140.dll
Failed to load symbols for C:\Windows\System32\vcruntime140.dll
Failed to load symbols for C:\Windows\System32\oleaut32.dll
Symbol information saved to report.
```

I will also try and reduce the times that RenderChunk gets called.