# Game Development with SFML

Simple and Fast Multimedia Library

C++ Usergroup Zentralschweiz 2019 | Lukas Dürrenberger | @DarkCisum

# About Me

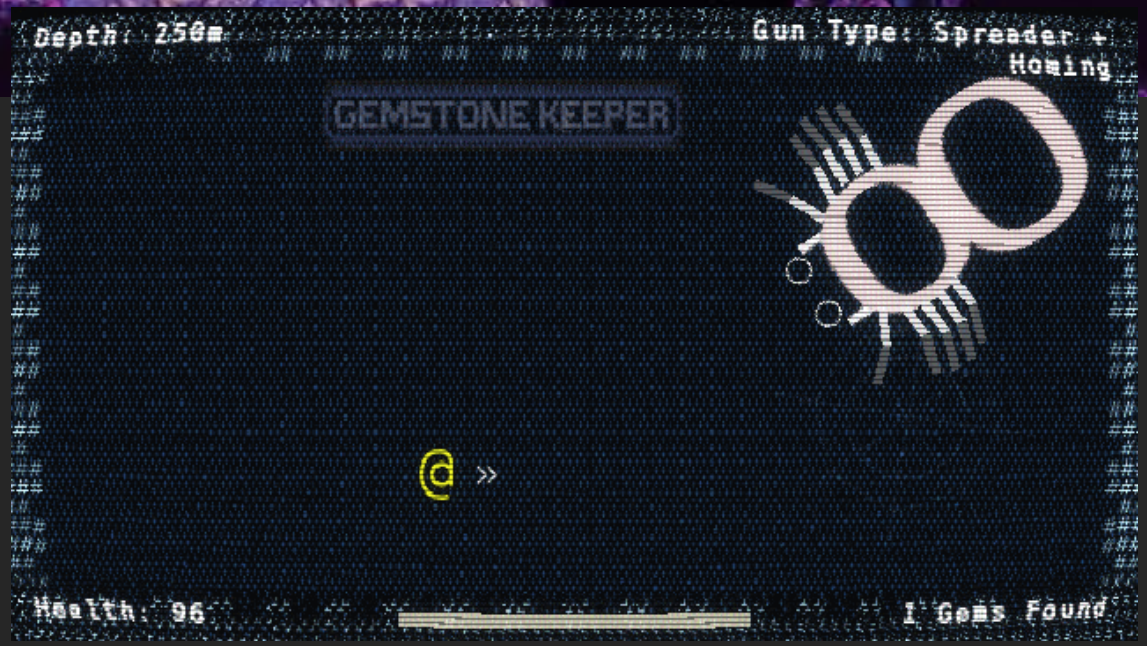
Lukas Dürrenberger

eXpl0it3r




REMNANTS
NAEZITH


MOONQUEST


Depth: 250m
Gun Type: Spreader + Homing
GEMSTONE KEEPER
@ »
Health: 96
1 Gems Found

# What is…?


Simple and Fast Multimedia Library



# SFML Modules

**System**
- Threads
- Mutex
- Clock

**Audio**
- Sound
- Music
- 3D Listener

**Window**
- Window
- OpenGL Context
- Input
  - Keyboard
  - Mouse
  - Joystick / Gamepad
  - Touch

**Graphics**
- Texture
- Shapes
- Vertex Array/Buffer
- Shaders
- Text
- View
- RenderTexture

**Network**
- UDP Socket
- TCP Socket
- FTP
- HTTP 1.0
- Packet




REMNANTS OF
NAEZITH


Castle Clamber


CENDRIC
AN RPG-PLATFORMER


AWAY TEAM


ZERAN'S
FOLLY


CUIT


GEMSTONE KEEPER
Gamepopper
Alpha Build
Start Game
Score Attack
Daily Run
Boss Rush
Exit


OCTO
NAUT
ODYSSEY


TURNOVER


MOONQUEST


DISPERSIO


DoodleBob
Virtual Companion


CREA

# Community

**Forum Statistics** (since 2012)

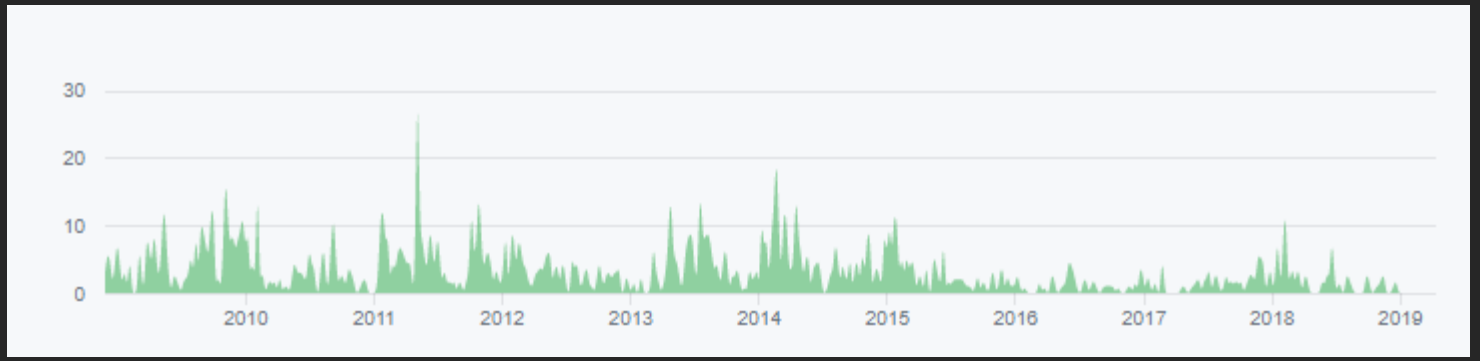
- Average posts per day: 48.3
- Average topics per day: 7.15
- Average online per day: 99.6
- Total posts: >150k
- Total topics: >20k

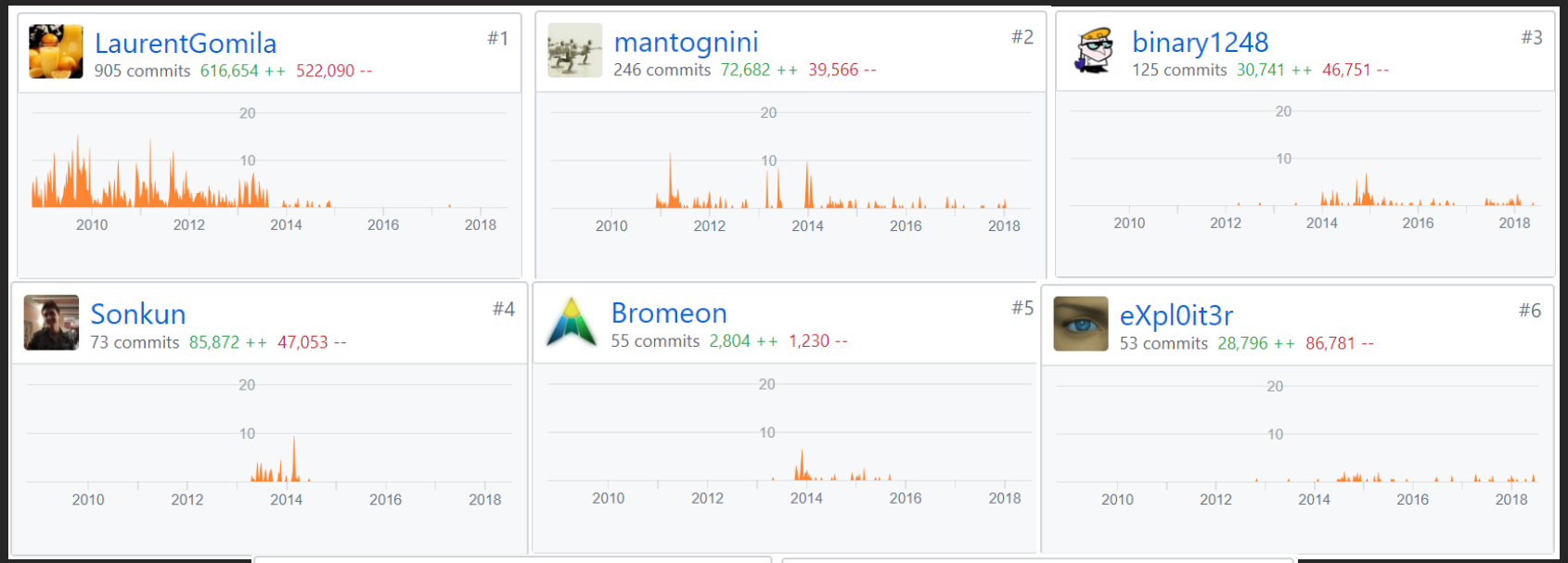
**Website Statistics**

- ~100k visits per month
- ~400k page views per month
- ~16k downloads per month



# The SFML Team






# Future

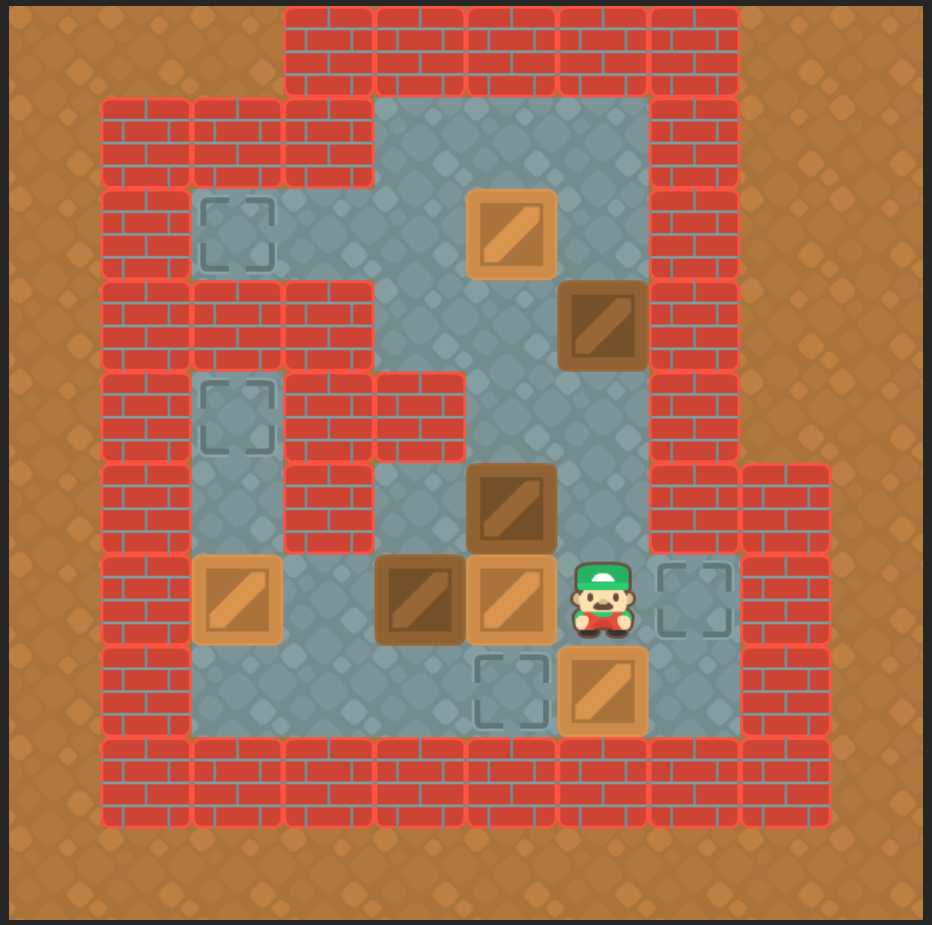
- Expanding Community Contributions
- Modern OpenGL Backends
- Nintendo Switch
- SFML 3
- …


# Workshop

Lets create a game!



# Goal: Basic Sokoban Clone





# 01 – Setup

Install & Compile



# 01 – Getting Started

https://www.sfml-dev.org/tutorials/2.5/#getting-started


https://github.com/eXpl0it3r/SFML-Workshop



# 01 – Package Managers



- `git clone https://github.com/SFML/SFML.git`
  `cmake . [options]`
  `cmake --build . --target install [options]`
- `conan install sfml/2.5.1@bincrafters/stable`
- `vcpkg install sfml`
- `apt-get install libsfml-dev`
- `pacman -S sfml`
- `brew install sfml`
- Download from https://www.sfml-dev.org/
- …



# 01 – SFML Basics

- SFML 2 uses C++03
- RAII Classes
- Event Processing is Required
- Flyweight Pattern for Resources



# 01 – Game Loop

- As long as the windows is open
  - As long as there are new events
    - Process sf::Event::Closed
  - Clear the window
  - (Draw objects onto the window)
  - Display the window

**Advanced**

- window.setFramerateLimit(60)
- FPS: 1.f / clock.restart().asSeconds()

**Tips**

- Use #include <SFML/Graphics.hpp>
- Use sf::RenderWindow



# 02 – Movement

Input & Rendering



# 02 – Input & Movement

- If key is pressed, move just once
- Depending on the key, move in that direction
- Always move a specific distance
- next_position = current_position + (direction_vector * distance)

**Tips**

- Store the keyboard state in std::map<sf::Keyboard::Key, bool>
- Use sf::Event::KeyPressed & sf::Event::KeyReleased
- Use a sf::Vector2i as direction_vector



# 03 – Structure

Scope Class



# 03 – Simple Refactoring

- main() should contain minimal start up code
- No global variables!
- No global stateful free functions

**Tips**

- Fine grain game loop:
  - Handle events
  - Update
  - Render

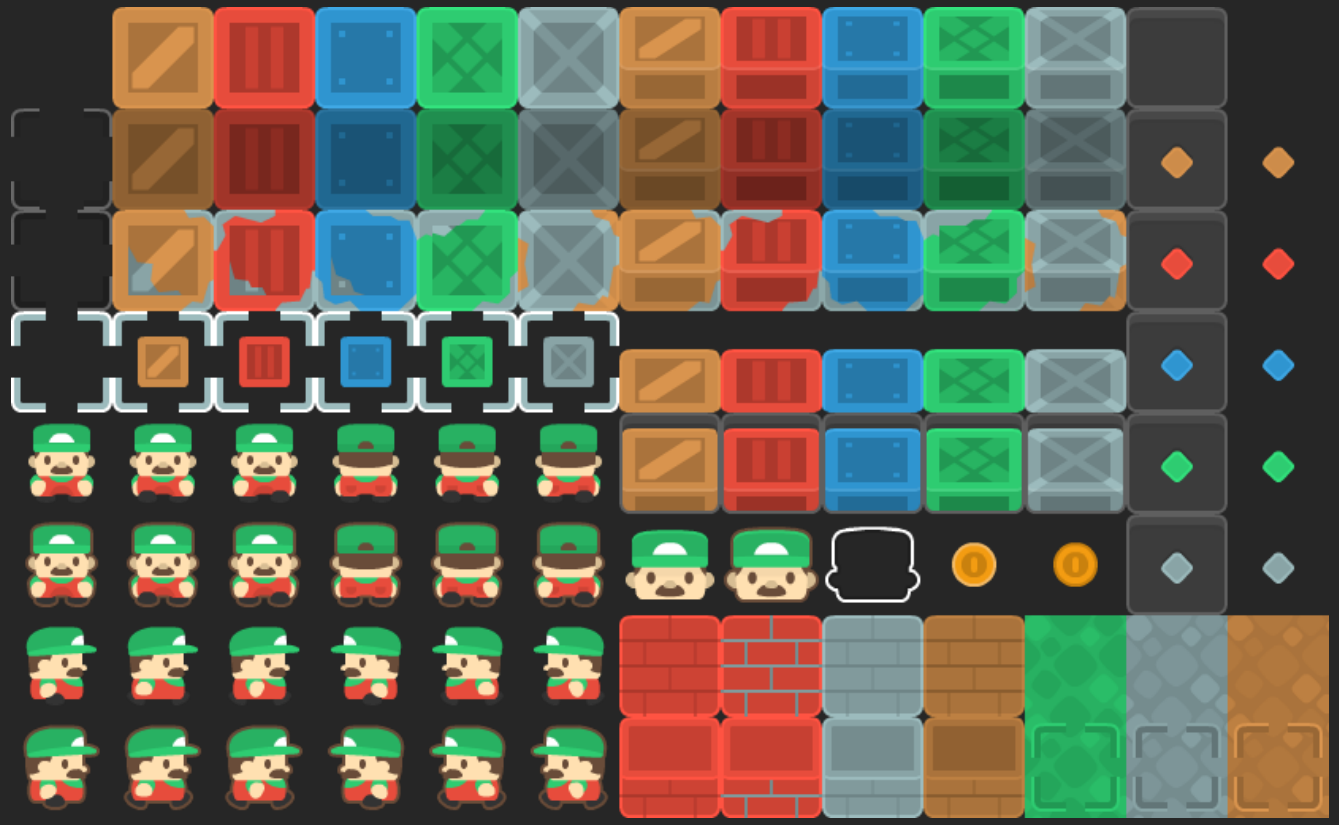


# 04 – Sprite & Texture

Display an Image



# Sprite & Texture Interaction



sf::Texture

sf::Rect<T>

sf::Sprite



# 04 – Handling Resources

- Heavy resource & light data object
- Always ensure loadFromFile() succeeds!
- Don’t copy sf::Texture

**Tips**

- Manage resources: std::map<std::string, std::unique_ptr<T>>

**Advanced**

- Add a reset function



# 05 – Map

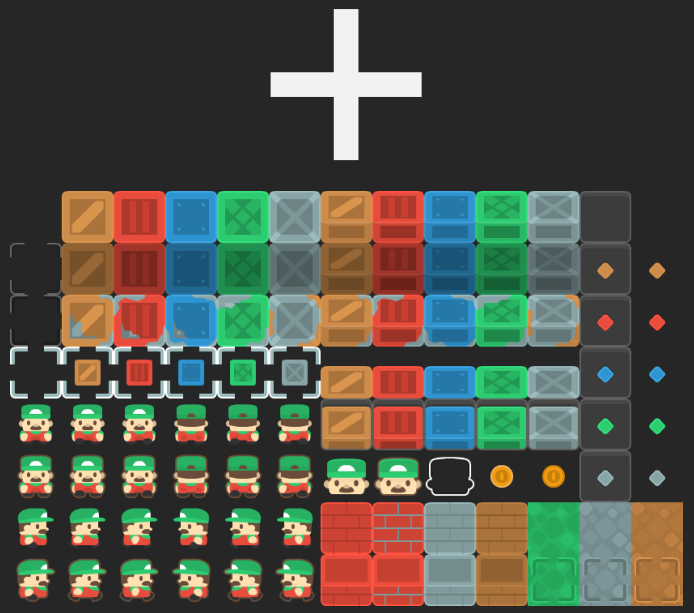
## Walk on Tiles



# Tile Map Creation

```
std::vector<int> data =
{
    90,  90,  90,  85,  85,  85,  85,  85,  90,  90,
    90,  85,  85,  85,  89,  89,  89,  85,  90,  90,
    90,  85, 102,  89,  89,  89,  89,  85,  90,  90,
    90,  85,  85,  85,  89,  89, 102,  85,  90,  90,
    90,  85, 102,  85,  85,  89,  89,  85,  90,  90,
    90,  85,  89,  85,  89, 102,  89,  85,  85,  90,
    90,  85,  89,  89, 102,  89,  89, 102,  85,  90,
    90,  85,  89,  89,  89, 102,  89,  89,  85,  90,
    90,  85,  85,  85,  85,  85,  85,  85,  85,  90,
    90,  90,  90,  90,  90,  90,  90,  90,  90,  90,
};
```

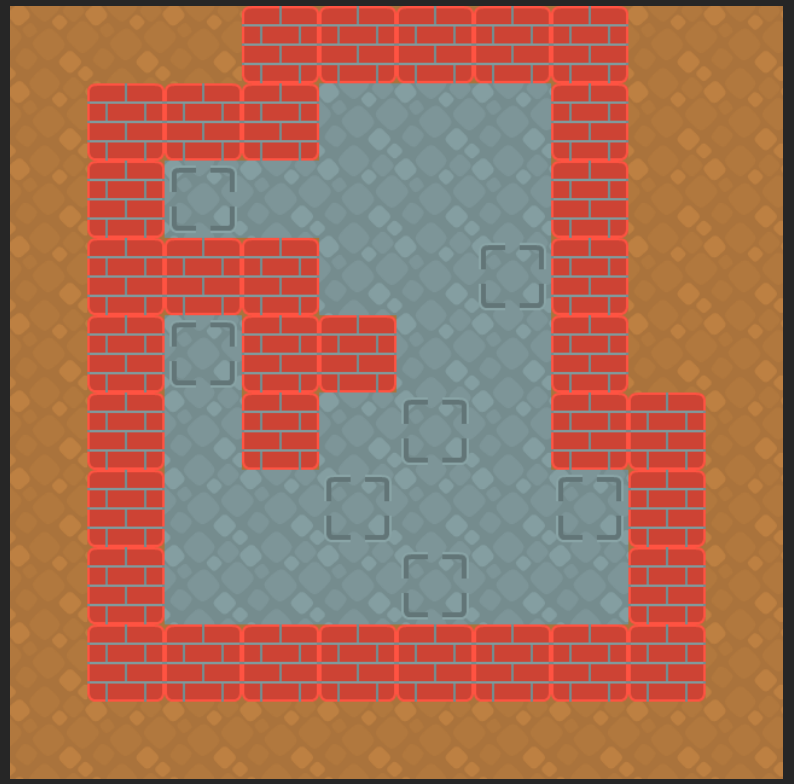
std::vector<int>



sf::Texture

sf::Rect<T>

sf::Sprite



# 05 – Tile Map

- std::vector<int> defines the map tiles
- Calculate texture rects for each tile
- Generate std::vector<sf::Sprite>
- Collision: If next_position in the map is a collision value
- Use a TileType enum

**Tips**

- texture_u = tile_number % (map_size.x / tile_size.x)
- texture_v = tile_number / (map_size.x / tile_size.x)

**Advanced**

- Use std::vector<sf::Vertex>



# 06 – Entity

Move Boxes!



# 06 – Boxes Rendering

- Not part of the map
- Only move it possible
- Change texture rect on target location
- Collision: Check position + 1 and position + 2 in move direction

**Tips**

- std::abs(vec1.x - vec2.x) <= std::numeric_limits<float>::epsilon() && …



# 07 – Audio

Music & Sound Effects



# 07 – Music & Sound

- Open music file
- Play sf::Music
- Load sf::SoundBuffer
- Play sf::Sound for event

**Tips**

- sf::Music doesn’t use loadFromFile()
- Use ResourceHolder for sf::SoundBuffer

**Advanced**

- Use std::deque<sf::Sound>



# 08 – Text

## Font Rendering



# 08 – Font & Text

- Load font file
- Adjust sf::Text properties
- Render text when win condition is reach

**Tips**

- Use ResourceHolder for sf::Font

**Advanced**

- Add a step counter



# Closing Notes

Game Over



# A few more things…

- Shaders (GLSL)
- Render Texture
- Views
- OpenGL Context
- Mouse Input
- Joystick/Gamepad Input
- 3D Spatial Audio
- TCP & UDP Sockets
- UTF-32/16/8 Conversion
- Windows, Linux & macOS
- Android & iOS & Nintendo Switch
- …



# Thanks!

https://www.sfml-dev.org

https://github.com/eXpl0it3r/Talks

@DarkCisum

@sfmldev

Stickers! Anyone?