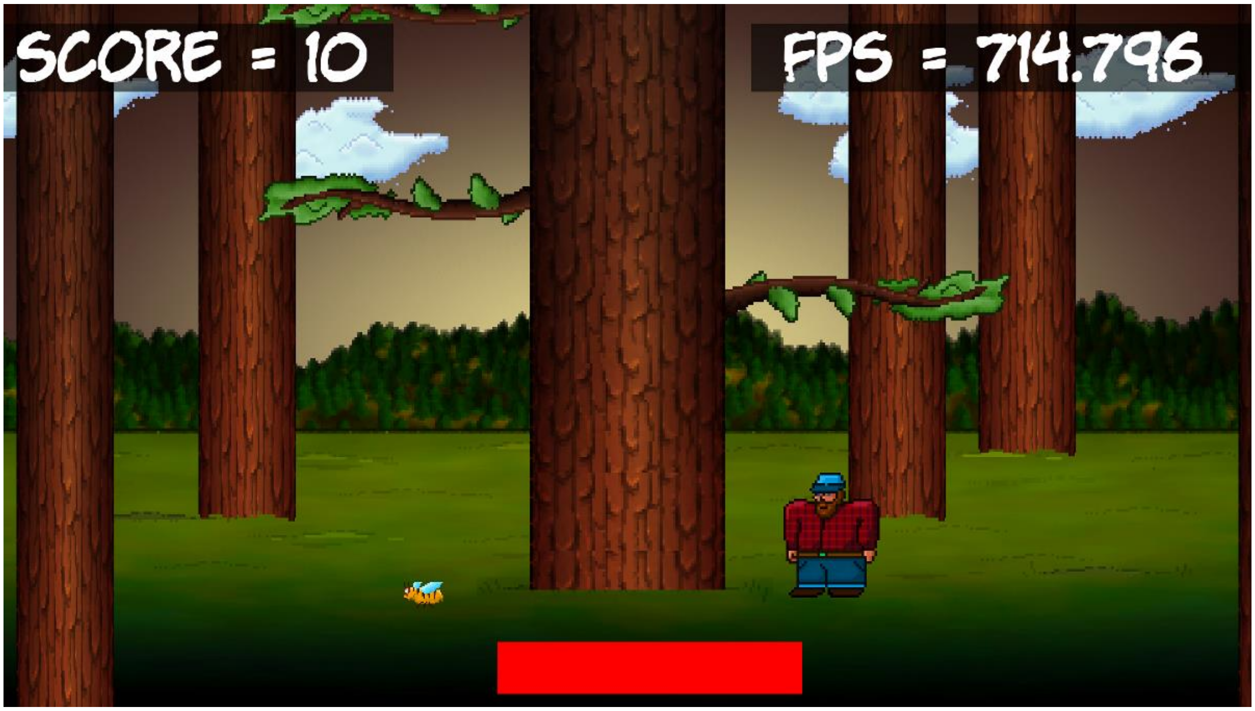
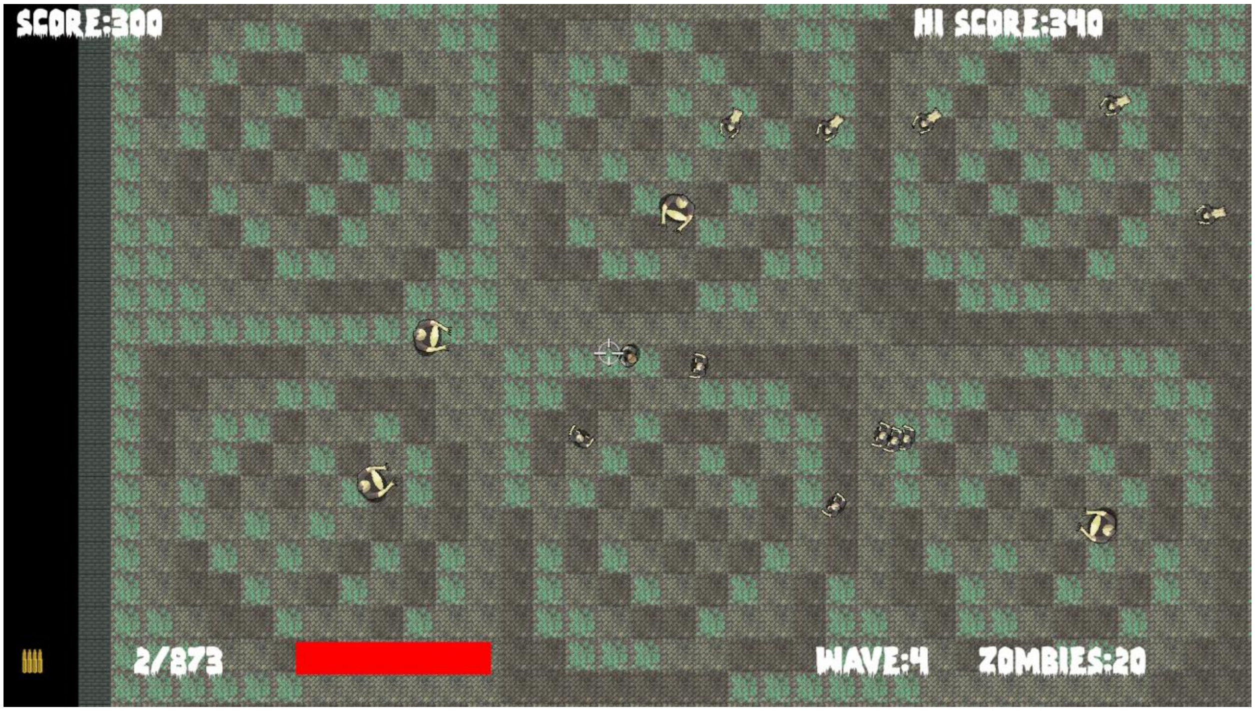
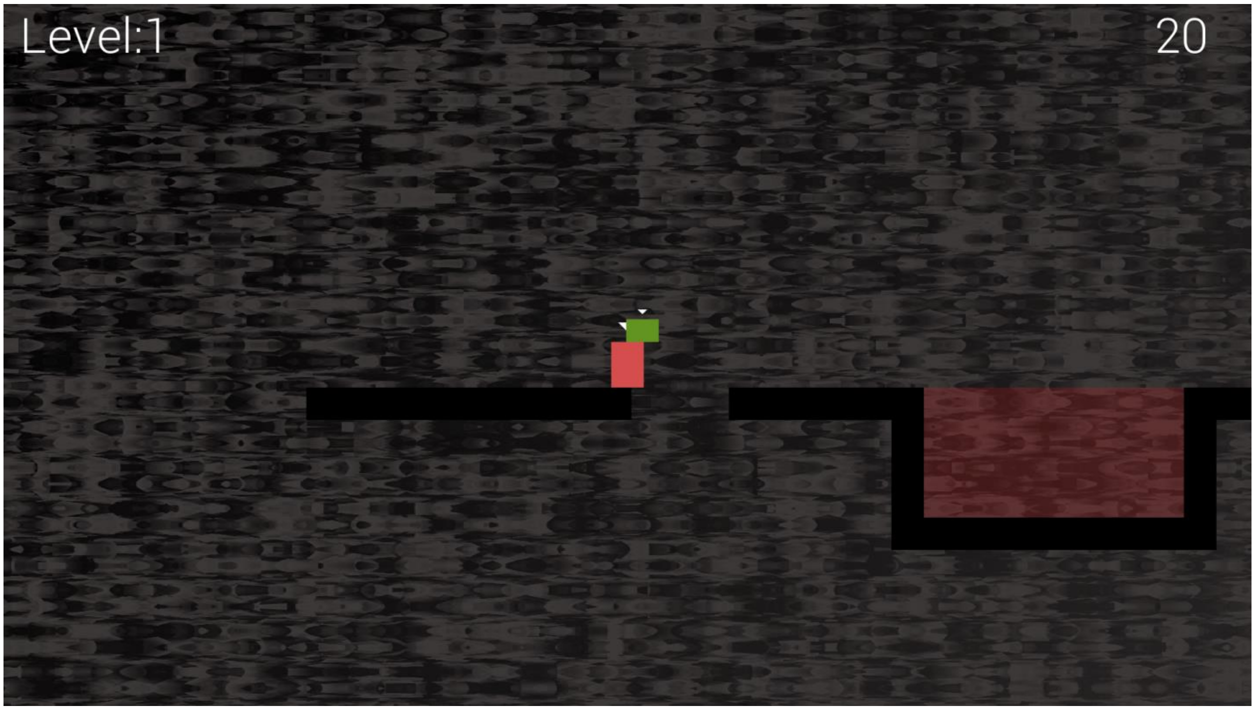
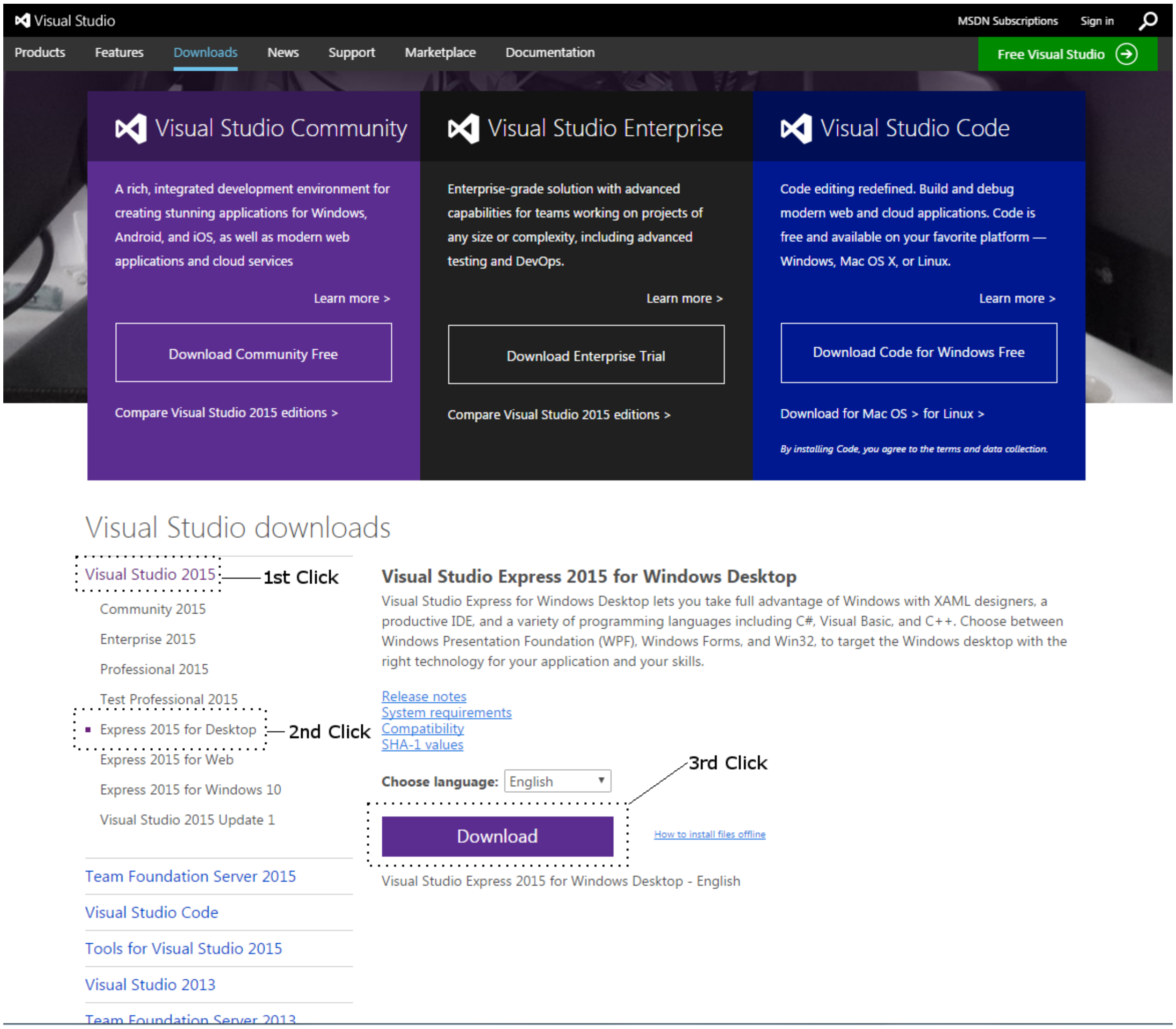
Chapter1: C++, SFML, Visual Studio and Starting the First Game






Level:1
20



Visual Studio Community

A rich, integrated development environment for creating stunning applications for Windows, Android, and iOS, as well as modern web applications and cloud services

Learn more >

Download Community Free

Compare Visual Studio 2015 editions >

Visual Studio Enterprise

Enterprise-grade solution with advanced capabilities for teams working on projects of any size or complexity, including advanced testing and DevOps.

Learn more >

Download Enterprise Trial

Compare Visual Studio 2015 editions >

Visual Studio Code

Code editing redefined. Build and debug modern web and cloud applications. Code is free and available on your favorite platform — Windows, Mac OS X, or Linux.

Learn more >

Download Code for Windows Free

Download for Mac OS > for Linux >

By installing Code, you agree to the terms and data collection.

Visual Studio downloads

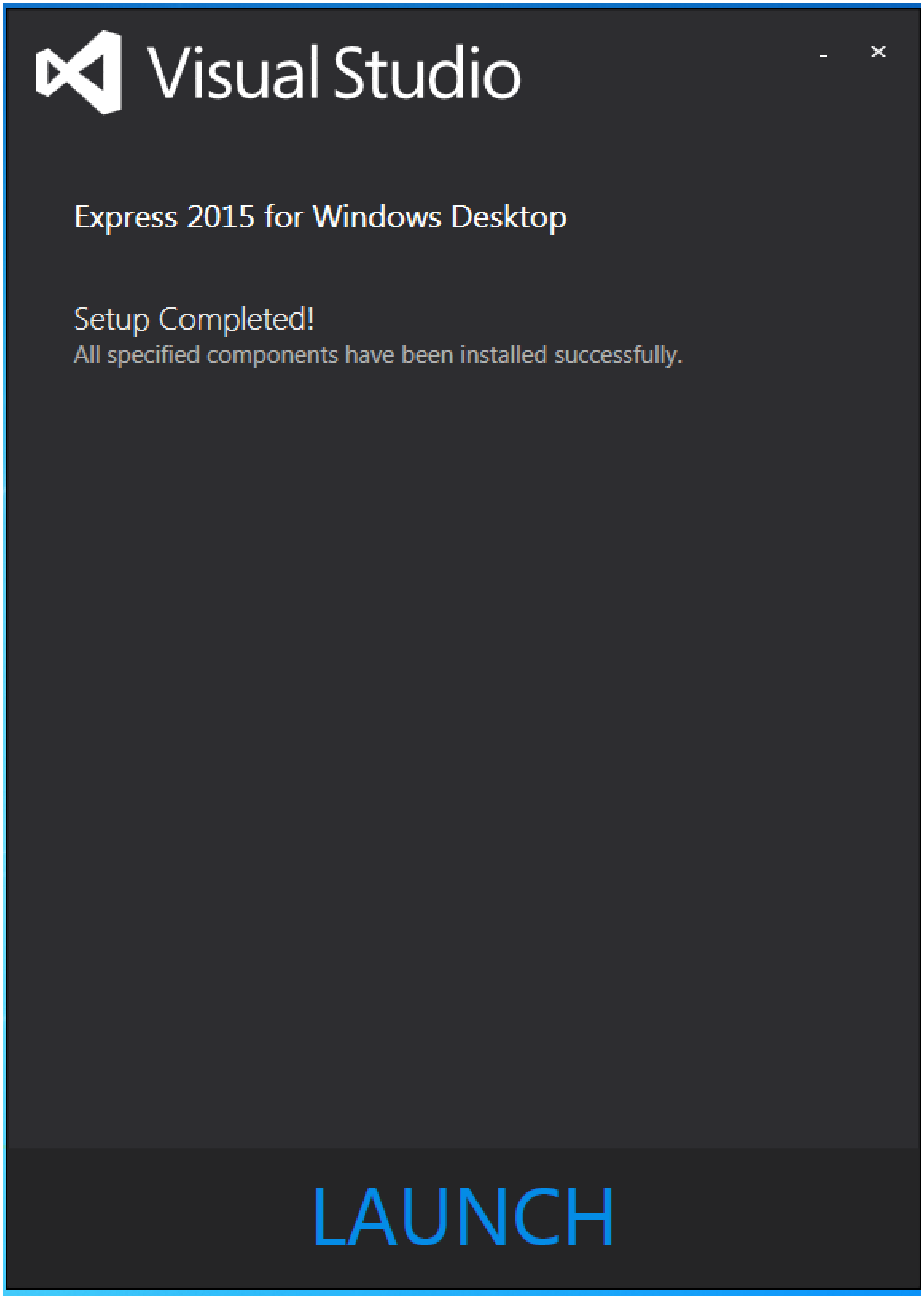
Visual Studio Express 2015 for Windows Desktop

Visual Studio Express for Windows Desktop lets you take full advantage of Windows with XAML designers, a productive IDE, and a variety of programming languages including C#, Visual Basic, and C++. Choose between Windows Presentation Foundation (WPF), Windows Forms, and Win32, to target the Windows desktop with the right technology for your application and your skills.

Release notes
System requirements
Compatibility
SHA-1 values

Choose language: English

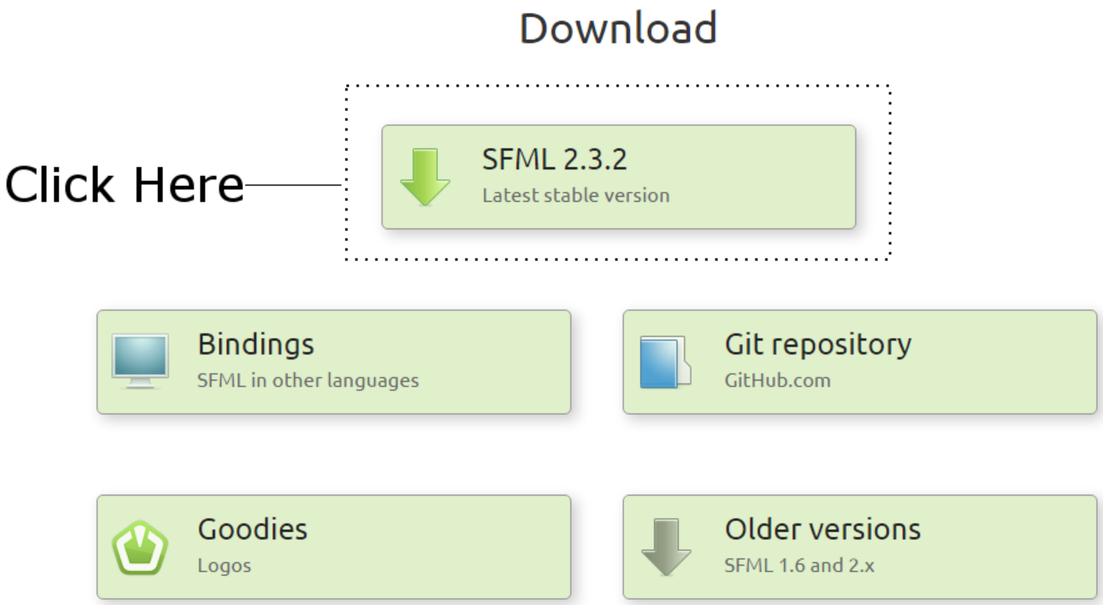
Visual Studio Express 2015 for Windows Desktop - English



Express 2015 for Windows Desktop

Setup Completed!

All specified components have been installed successfully.



Download

Click Here

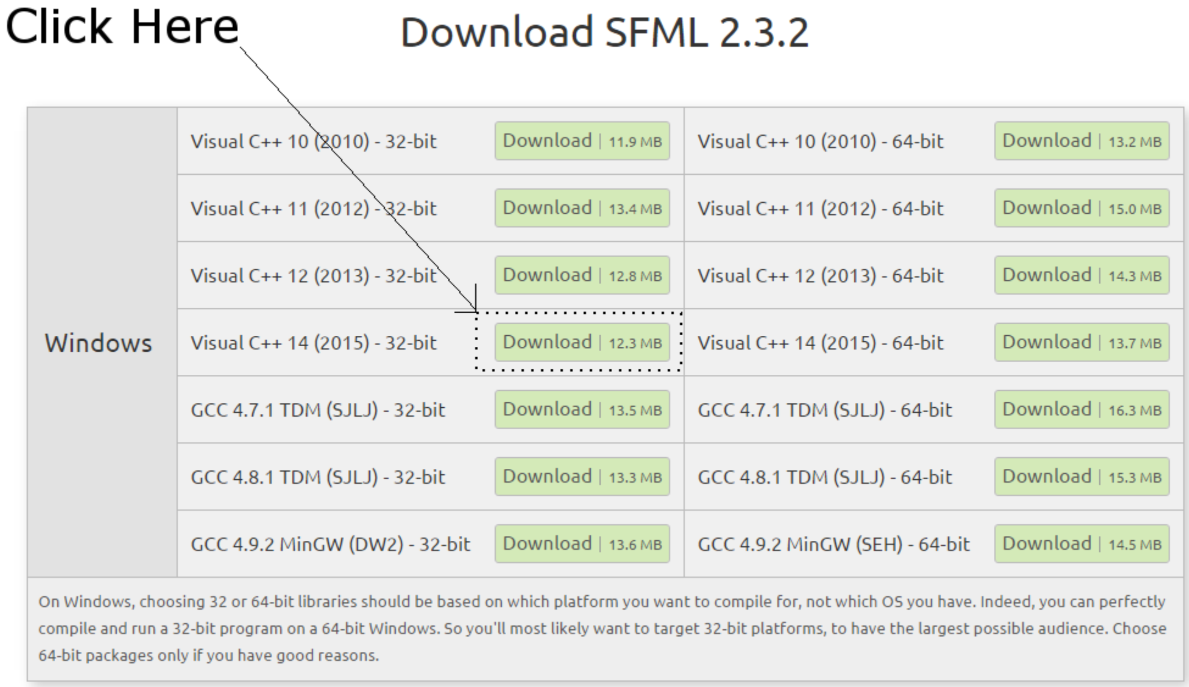
SFML 2.3.2
Latest stable version

Bindings
SFML in other languages

Git repository
GitHub.com

Goodies
Logos

Older versions
SFML 1.6 and 2.x

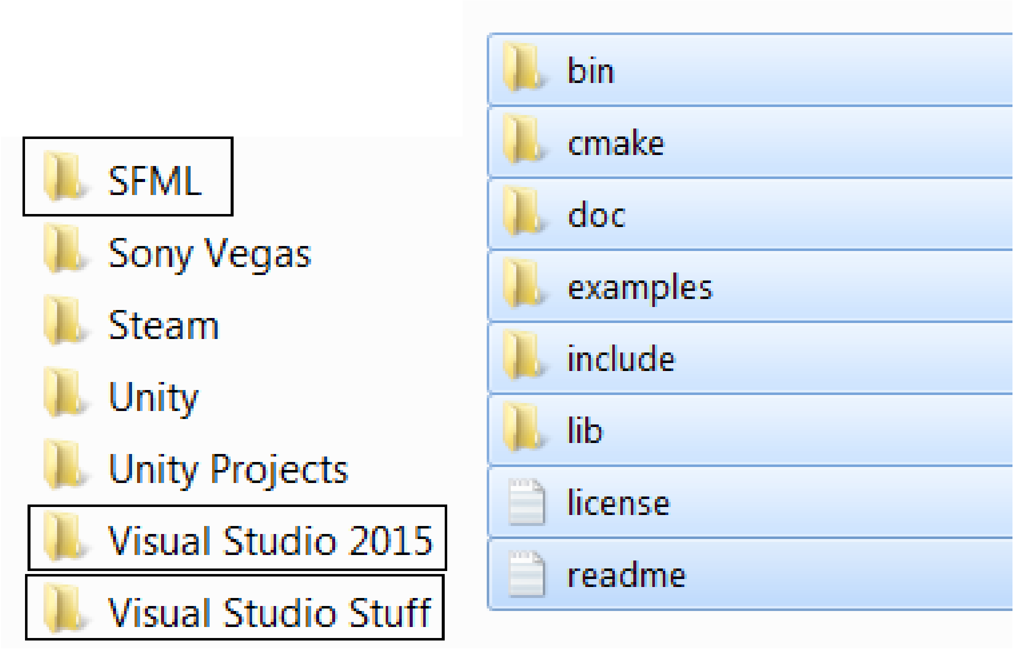


Click Here

Download SFML 2.3.2

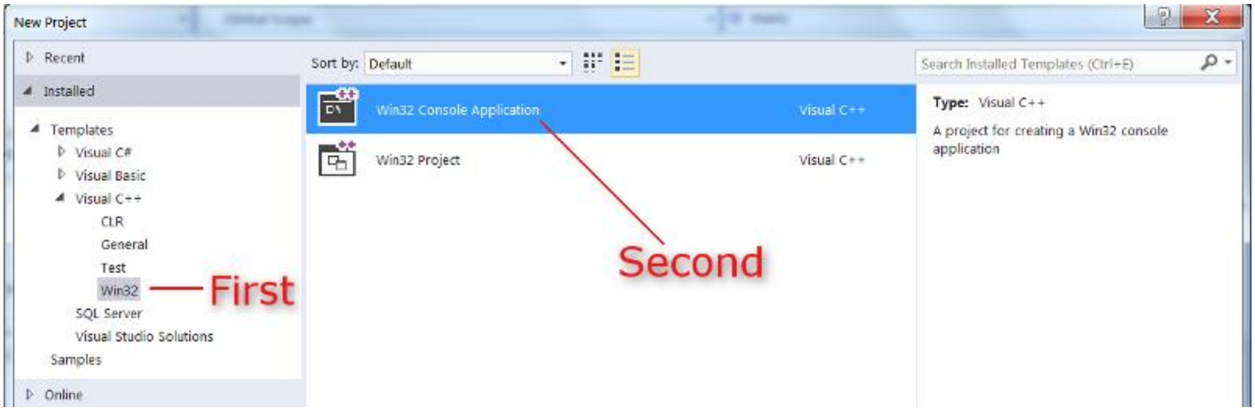
Windows	Visual C++ 10 (2010) - 32-bit	Download | 11.9 MB	Visual C++ 10 (2010) - 64-bit	Download | 13.2 MB
	Visual C++ 11 (2012) - 32-bit	Download | 13.4 MB	Visual C++ 11 (2012) - 64-bit	Download | 15.0 MB
	Visual C++ 12 (2013) - 32-bit	Download | 12.8 MB	Visual C++ 12 (2013) - 64-bit	Download | 14.3 MB
	Visual C++ 14 (2015) - 32-bit	Download | 12.3 MB	Visual C++ 14 (2015) - 64-bit	Download | 13.7 MB
	GCC 4.7.1 TDM (SJLJ) - 32-bit	Download | 13.5 MB	GCC 4.7.1 TDM (SJLJ) - 64-bit	Download | 16.3 MB
	GCC 4.8.1 TDM (SJLJ) - 32-bit	Download | 13.3 MB	GCC 4.8.1 TDM (SJLJ) - 64-bit	Download | 15.3 MB
	GCC 4.9.2 MinGW (DW2) - 32-bit	Download | 13.6 MB	GCC 4.9.2 MinGW (SEH) - 64-bit	Download | 14.5 MB

On Windows, choosing 32 or 64-bit libraries should be based on which platform you want to compile for, not which OS you have. Indeed, you can perfectly compile and run a 32-bit program on a 64-bit Windows. So you'll most likely want to target 32-bit platforms, to have the largest possible audience. Choose 64-bit packages only if you have good reasons.



SFML
Sony Vegas
Steam
Unity
Unity Projects
Visual Studio 2015
Visual Studio Stuff

bin
cmake
doc
examples
include
lib
license
readme



New Project

- Recent
- Installed
  - Templates
    - Visual C#
    - Visual Basic
    - Visual C++
      - CLR
      - General
      - Test
      - Win32
    - SQL Server
    - Visual Studio Solutions
  - Samples
- Online

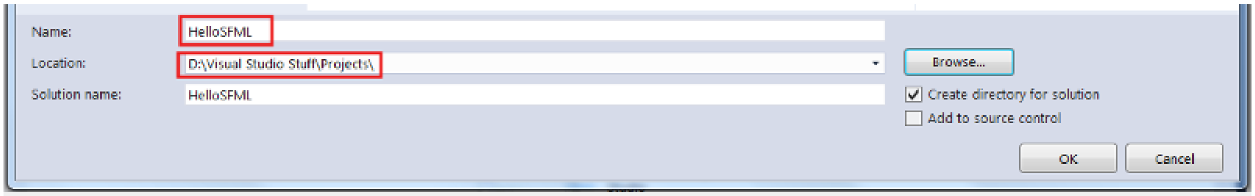
Sort by: Default

Search Installed Templates (Ctrl+E)

Win32 Console Application	Visual C++
Win32 Project	Visual C++

Type: Visual C++

A project for creating a Win32 console application



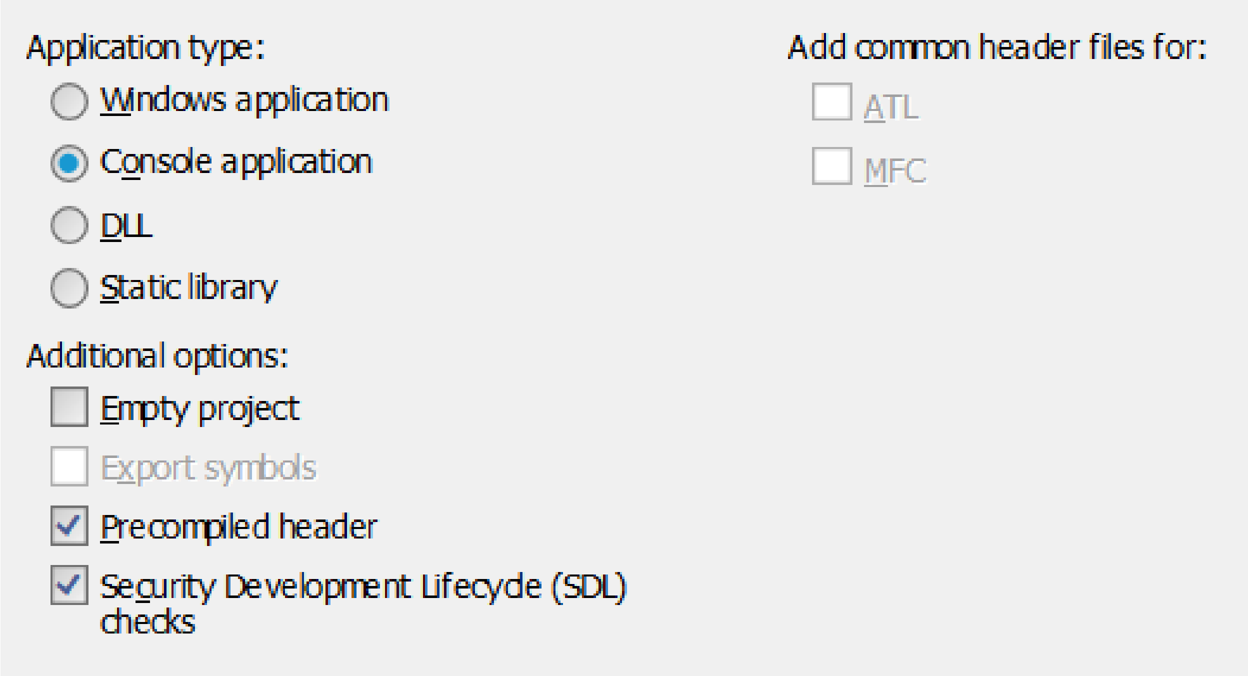
Name: HelloSFML

Location: D:\Visual Studio Stuff\Projects\

Solution name: HelloSFML

☑ Create directory for solution

□ Add to source control



Application type:

○ Windows application

● Console application

○ DLL

○ Static library

Add common header files for:

□ ATL

□ MFC

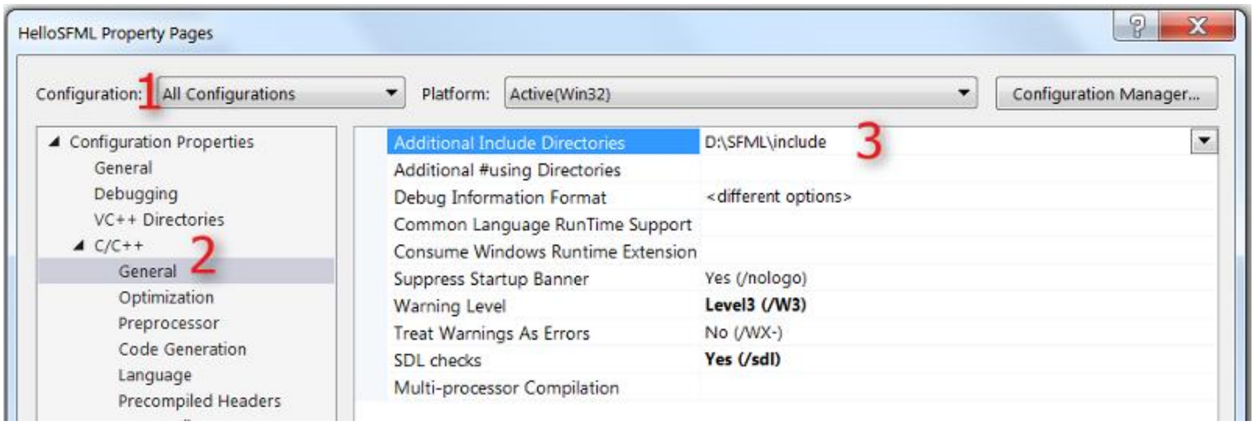
Additional options:

□ Empty project

□ Export symbols

☑ Precompiled header

☑ Security Development Lifecycle (SDL) checks

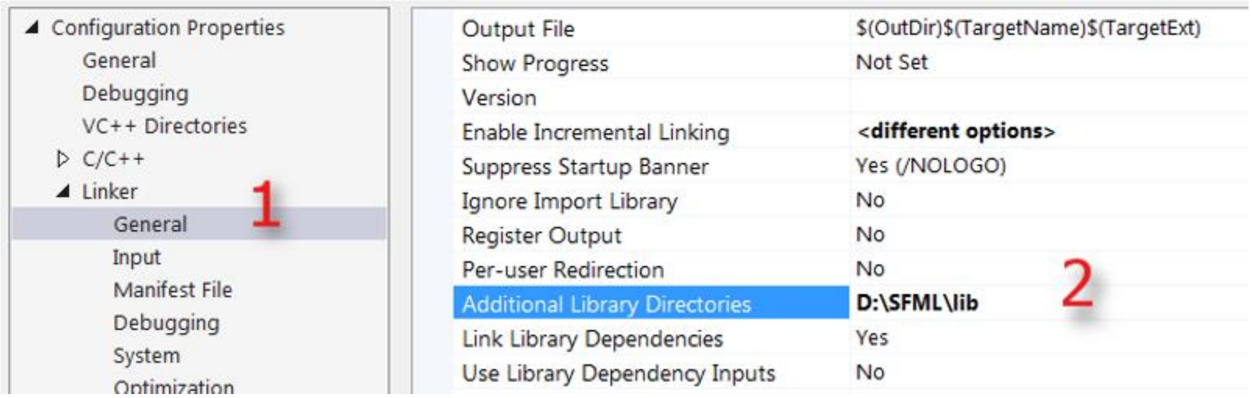


HelloSFML Property Pages

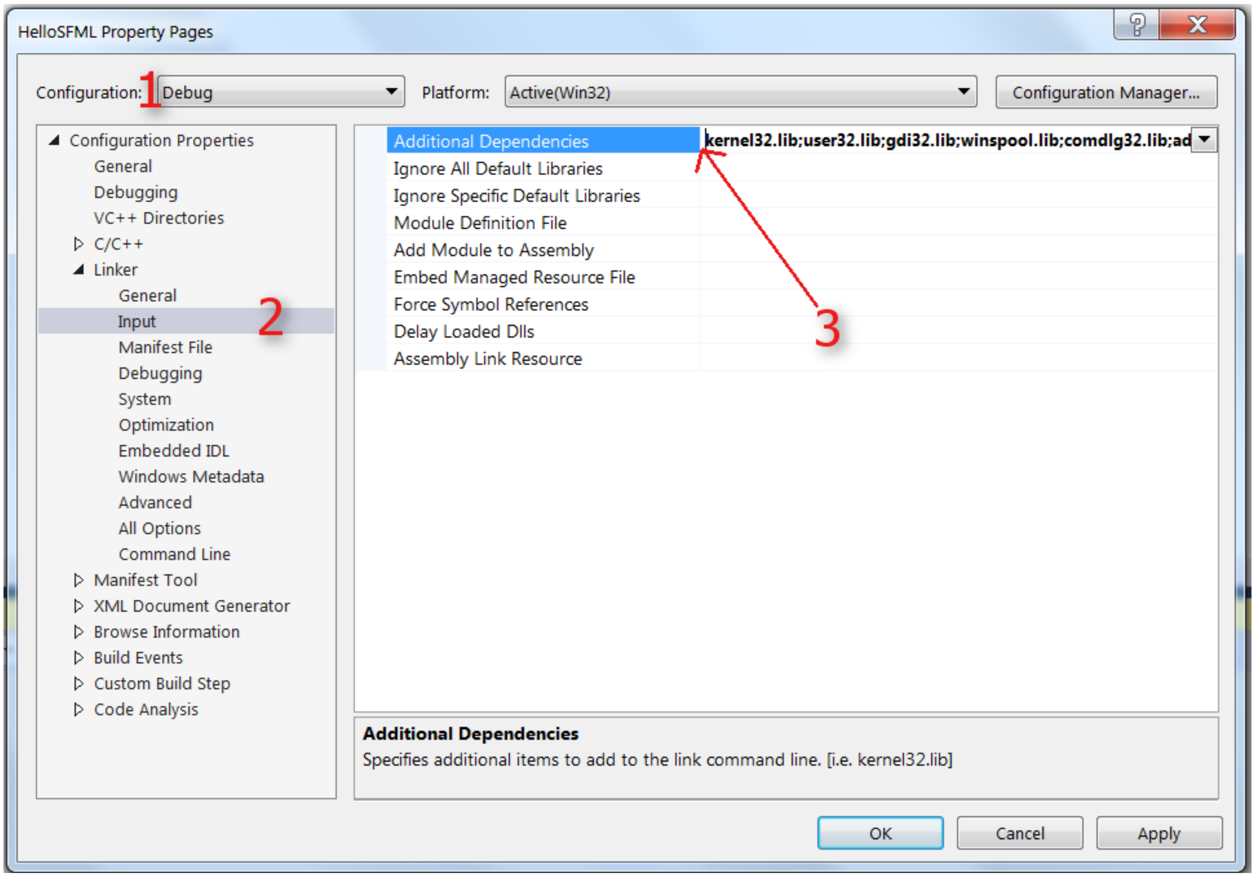
Configuration: All Configurations 1

Platform: Active(Win32)

Additional Include Directories	D:\SFML\include 3
Additional #using Directories	
Debug Information Format	<different options>
Common Language RunTime Support	
Consume Windows Runtime Extension	
Suppress Startup Banner	Yes (/nologo)
Warning Level	**Level3 (/W3)**
Treat Warnings As Errors	No (/WX-)
SDL checks	**Yes (/sdl)**
Multi-processor Compilation	



Output File	$(OutDir)$(TargetName)$(TargetExt)
Show Progress	Not Set
Version	
Enable Incremental Linking	**<different options>**
Suppress Startup Banner	Yes (/NOLOGO)
Ignore Import Library	No
Register Output	No
Per-user Redirection	No
Additional Library Directories	**D:\SFML\lib**
Link Library Dependencies	Yes
Use Library Dependency Inputs	No

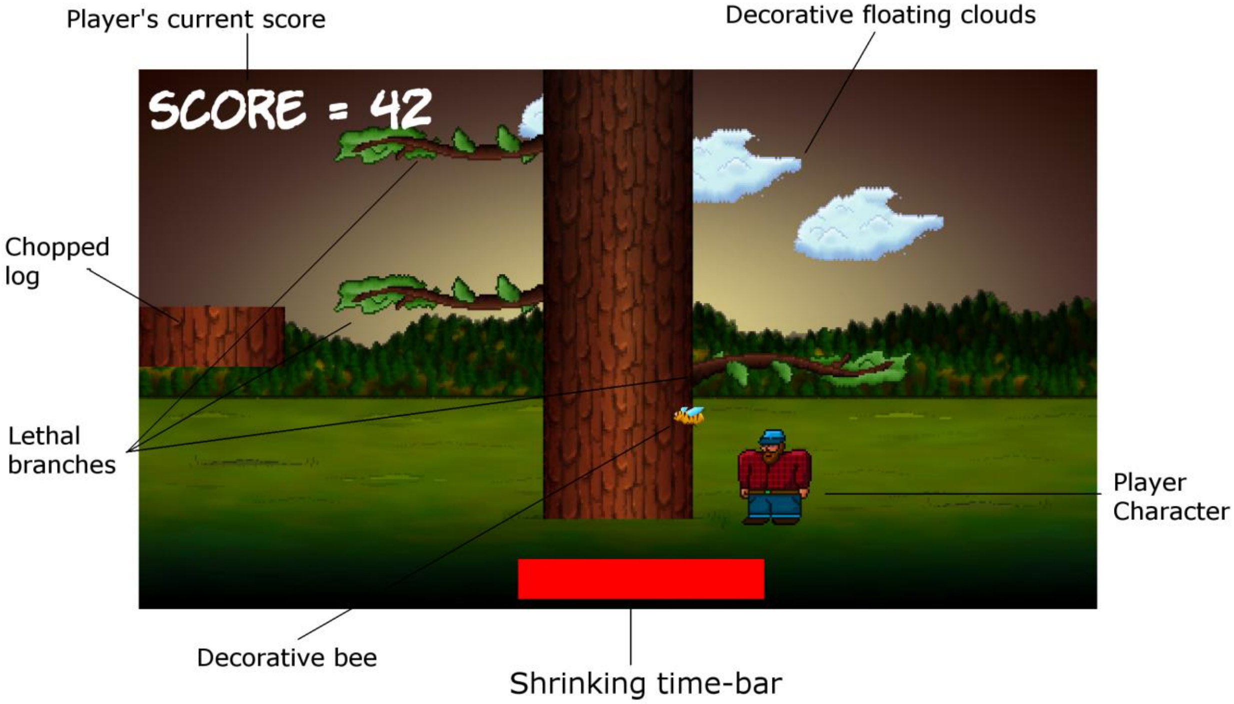


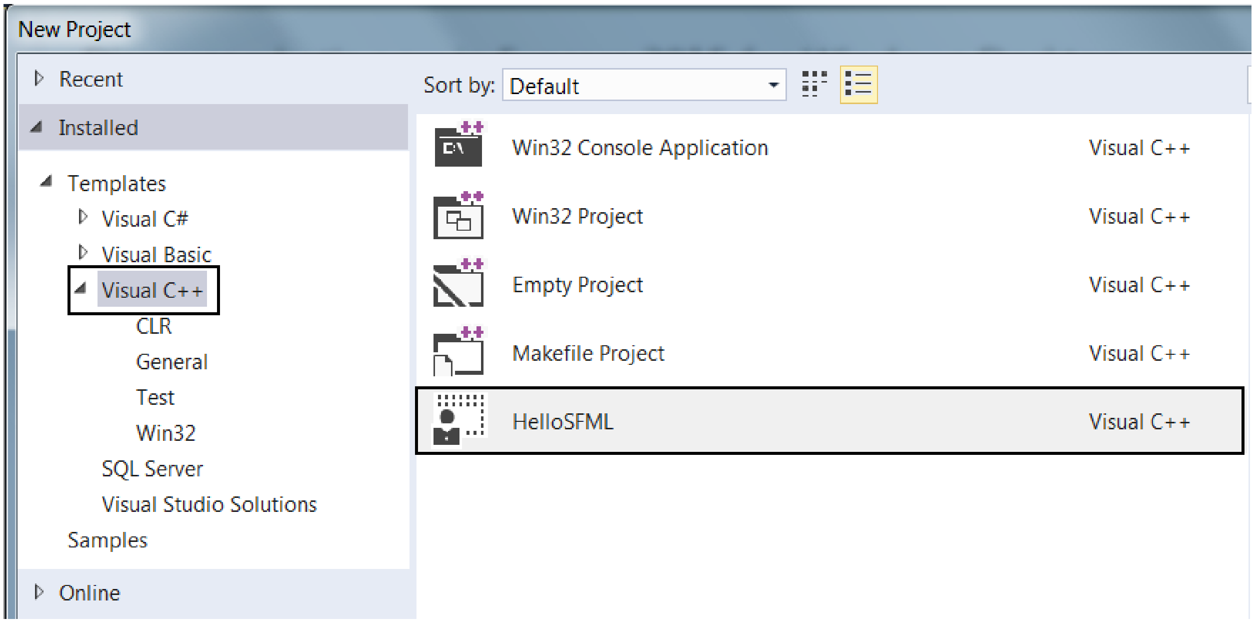
Configuration: Debug

Platform: Active(Win32)

Additional Dependencies	**kernel32.lib;user32.lib;gdi32.lib;winspool.lib;comdlg32.lib;ad**
Ignore All Default Libraries	
Ignore Specific Default Libraries	
Module Definition File	
Add Module to Assembly	
Embed Managed Resource File	
Force Symbol References	
Delay Loaded Dlls	
Assembly Link Resource	

Additional Dependencies
Specifies additional items to add to the link command line. [i.e. kernel32.lib]

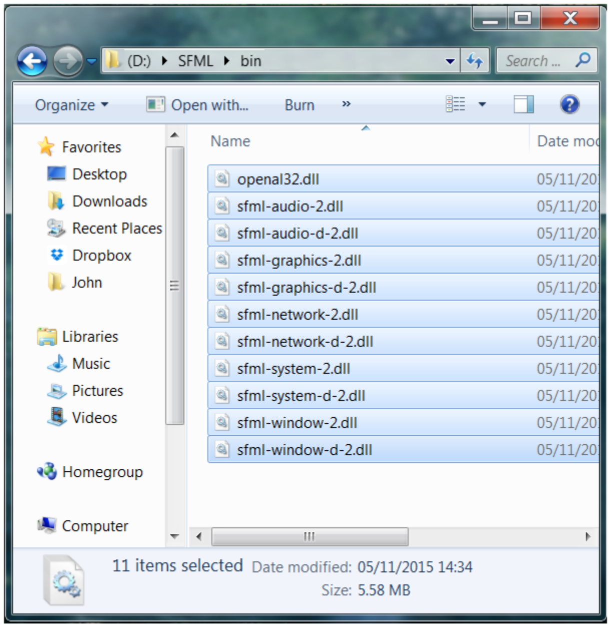



New Project

- Recent
- Installed
  - Templates
    - Visual C#
    - Visual Basic
    - Visual C++
      - CLR
      - General
      - Test
      - Win32
    - SQL Server
    - Visual Studio Solutions
  - Samples
- Online

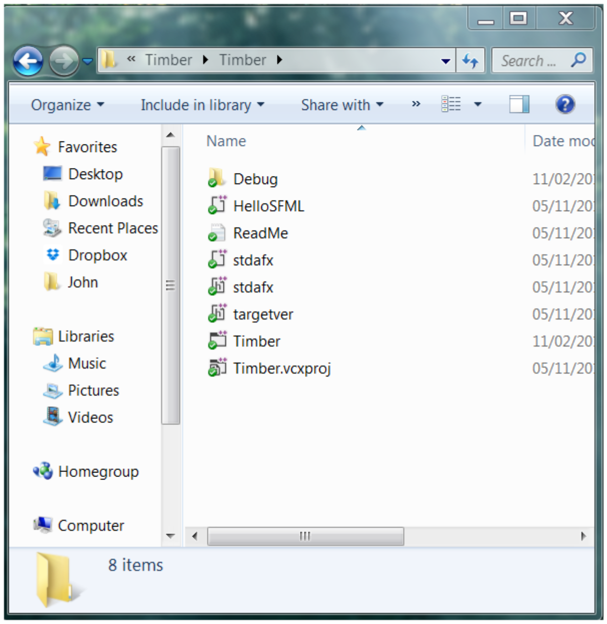
Sort by: Default

Win32 Console Application	Visual C++
Win32 Project	Visual C++
Empty Project	Visual C++
Makefile Project	Visual C++
HelloSFML	Visual C++



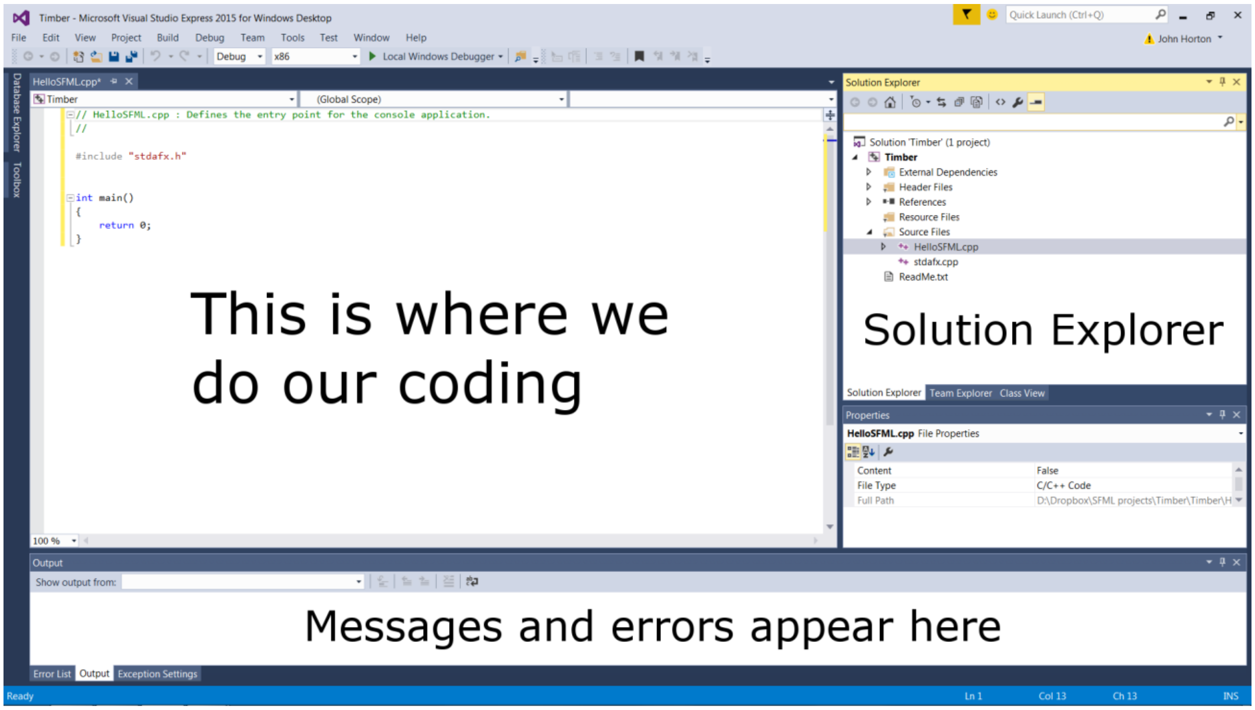
Name	Date mod
openal32.dll	05/11/20
sfml-audio-2.dll	05/11/20
sfml-audio-d-2.dll	05/11/20
sfml-graphics-2.dll	05/11/20
sfml-graphics-d-2.dll	05/11/20
sfml-network-2.dll	05/11/20
sfml-network-d-2.dll	05/11/20
sfml-system-2.dll	05/11/20
sfml-system-d-2.dll	05/11/20
sfml-window-2.dll	05/11/20
sfml-window-d-2.dll	05/11/20

11 items selected Date modified: 05/11/2015 14:34
Size: 5.58 MB



Name	Date mod
Debug	11/02/20
HelloSFML	05/11/20
ReadMe	05/11/20
stdafx	05/11/20
stdafx	05/11/20
targetver	05/11/20
Timber	11/02/20
Timber.vcxproj	05/11/20

8 items



```
// HelloSFML.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"


int main()
{
    return 0;
}
```

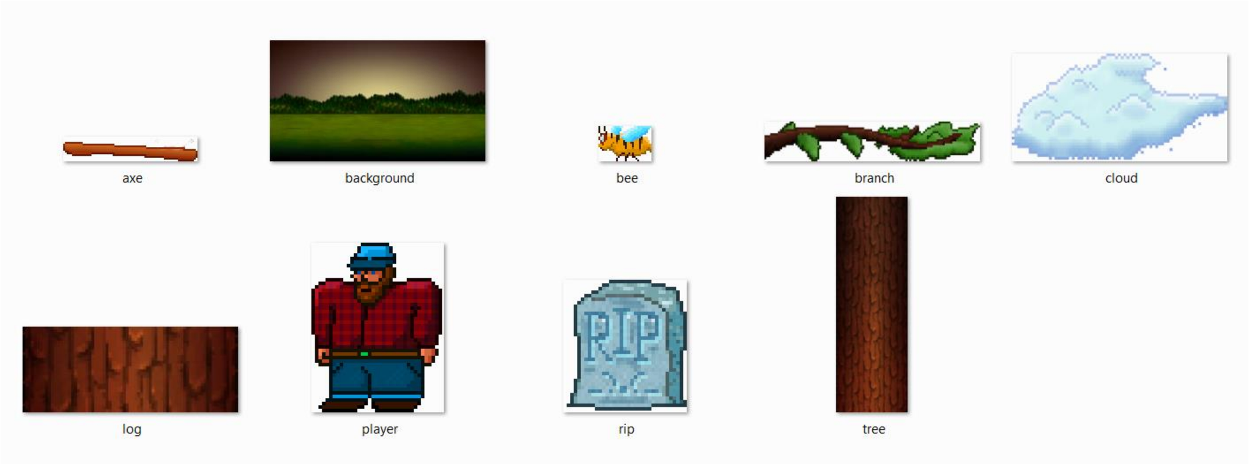
HelloSFML.cpp File Properties

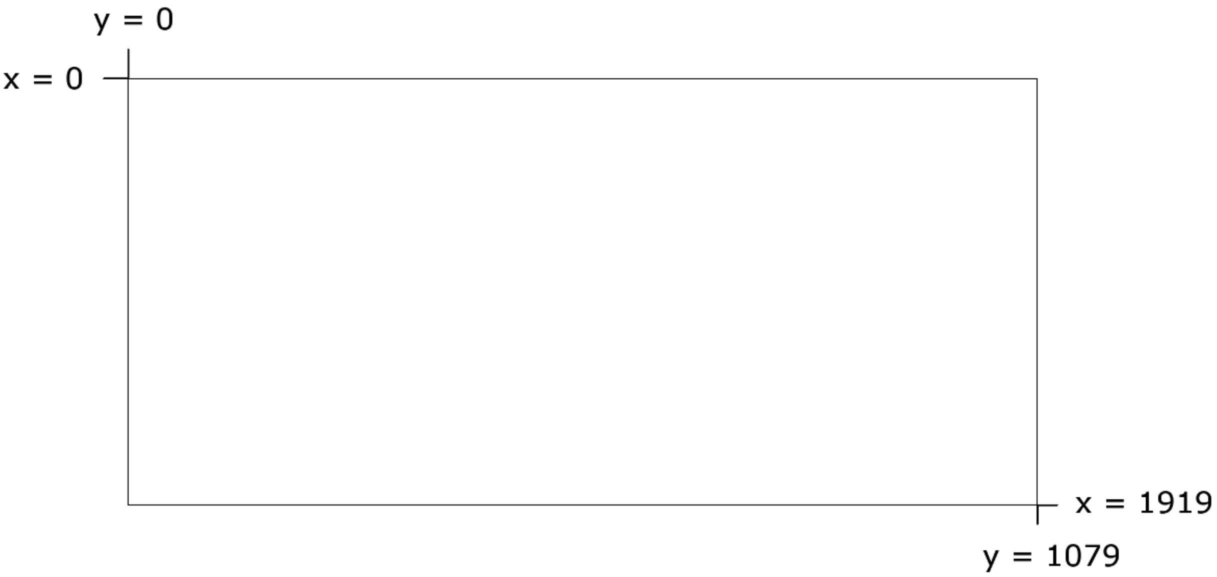
Content	False
File Type	C/C++ Code
Full Path	D:\Dropbox\SFML projects\Timber\Timber\H

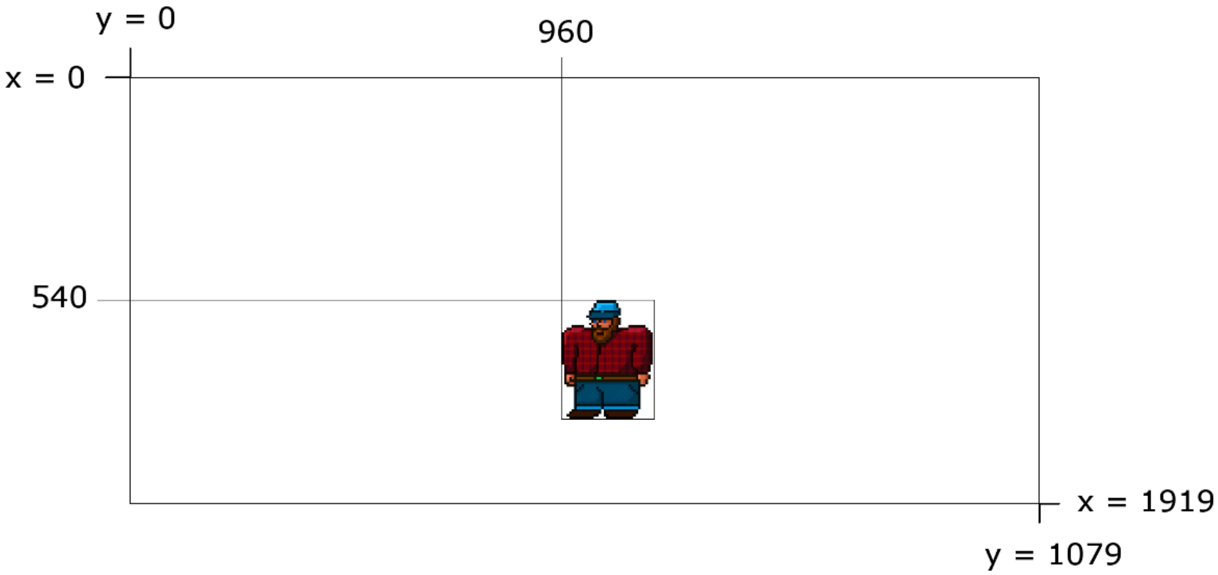
This is where we do our coding

Solution Explorer

Messages and errors appear here

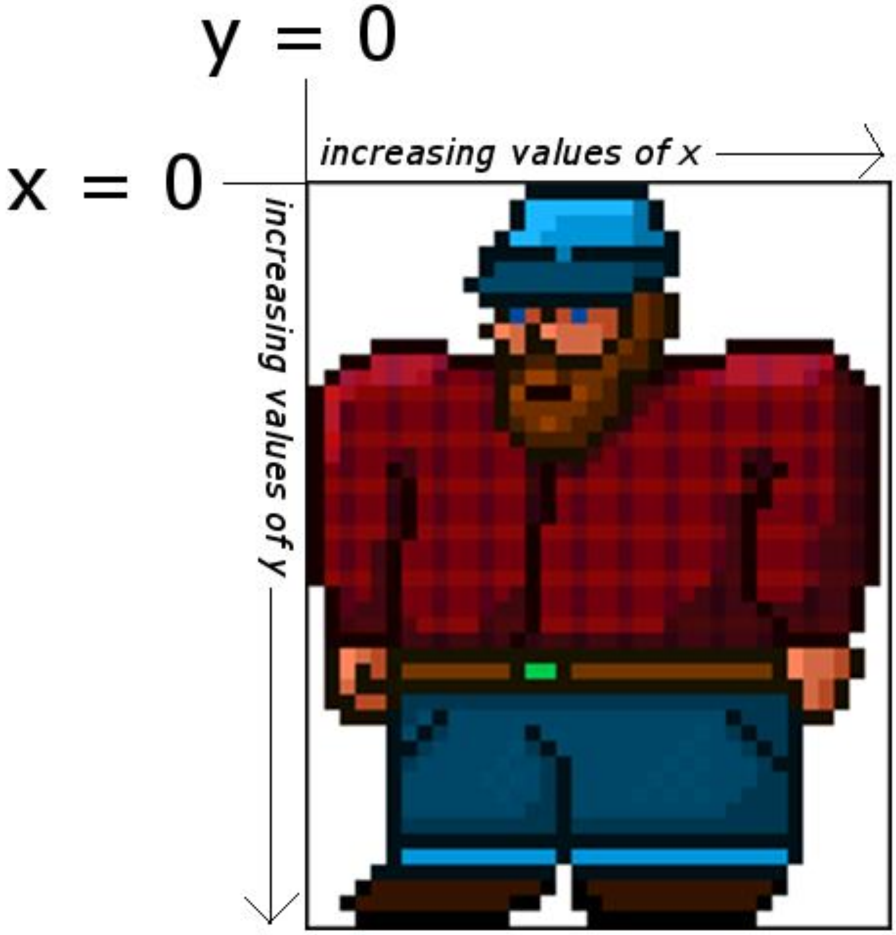

axe
background
bee
branch
cloud
log
player
rip
tree

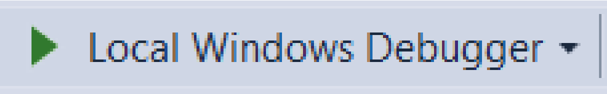

y = 0
x = 0
x = 1919
y = 1079


y = 0
960
x = 0
540
x = 1919
y = 1079

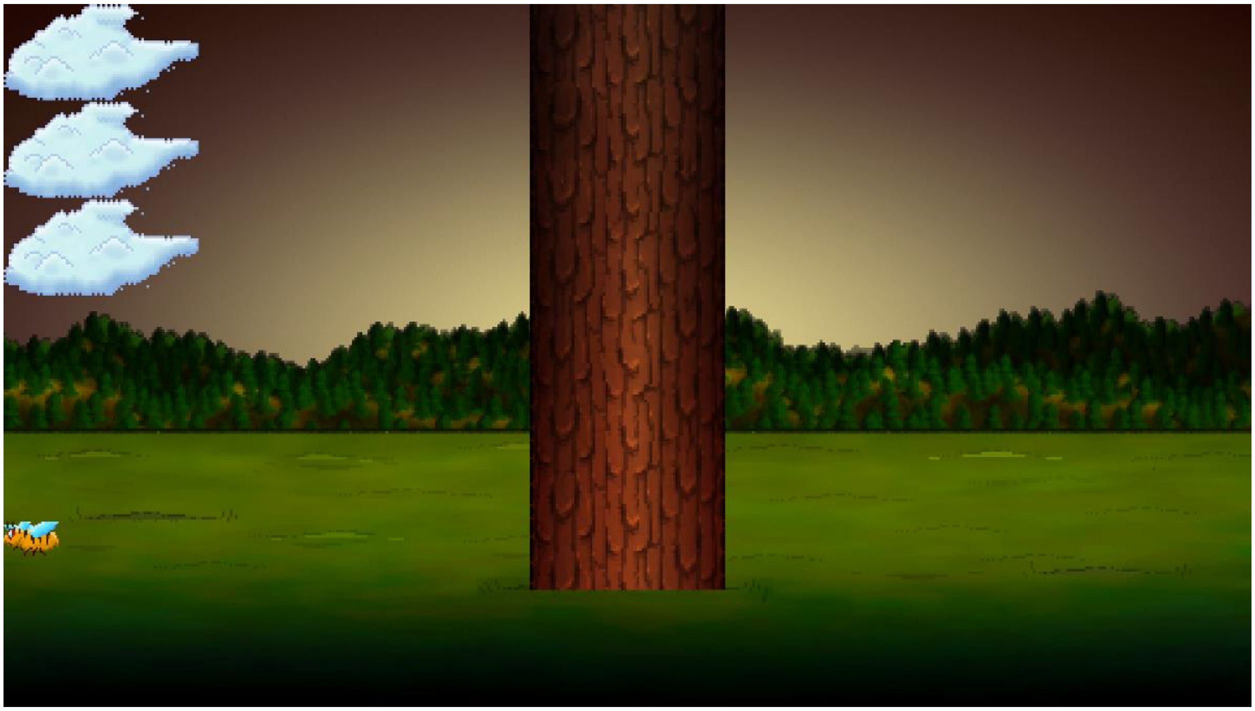
Internal Coordinates

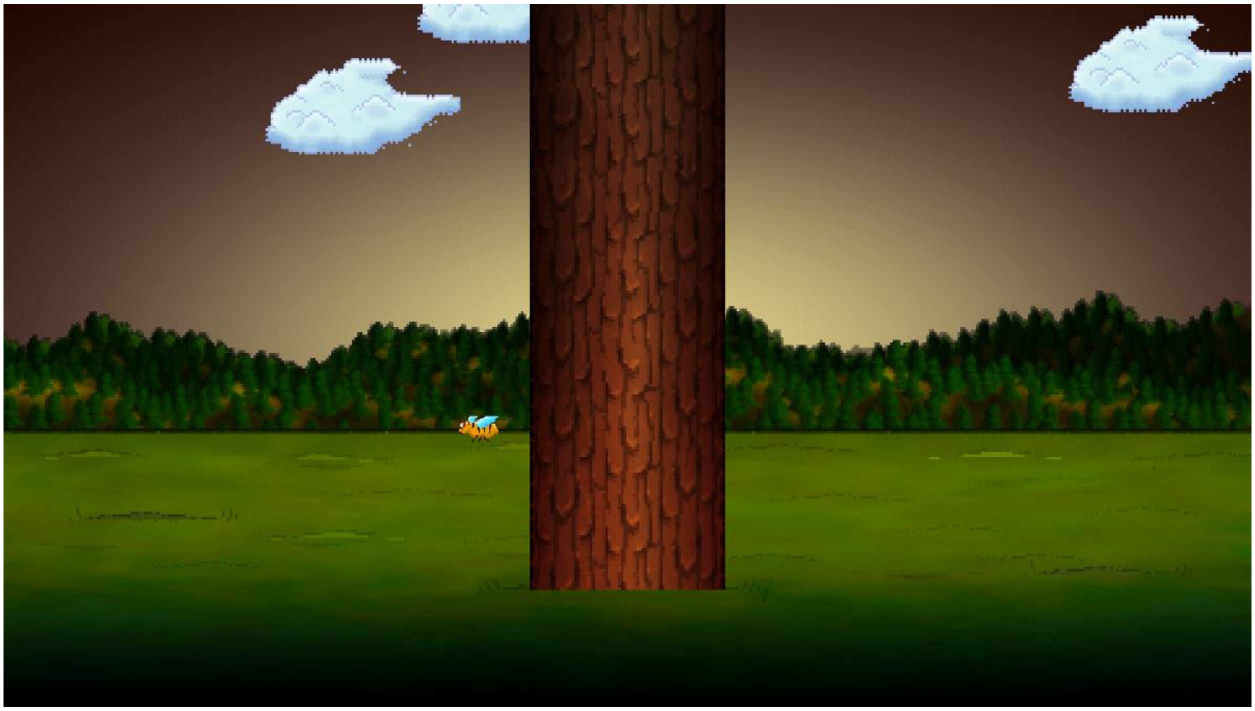
(Origin = 0, 0)



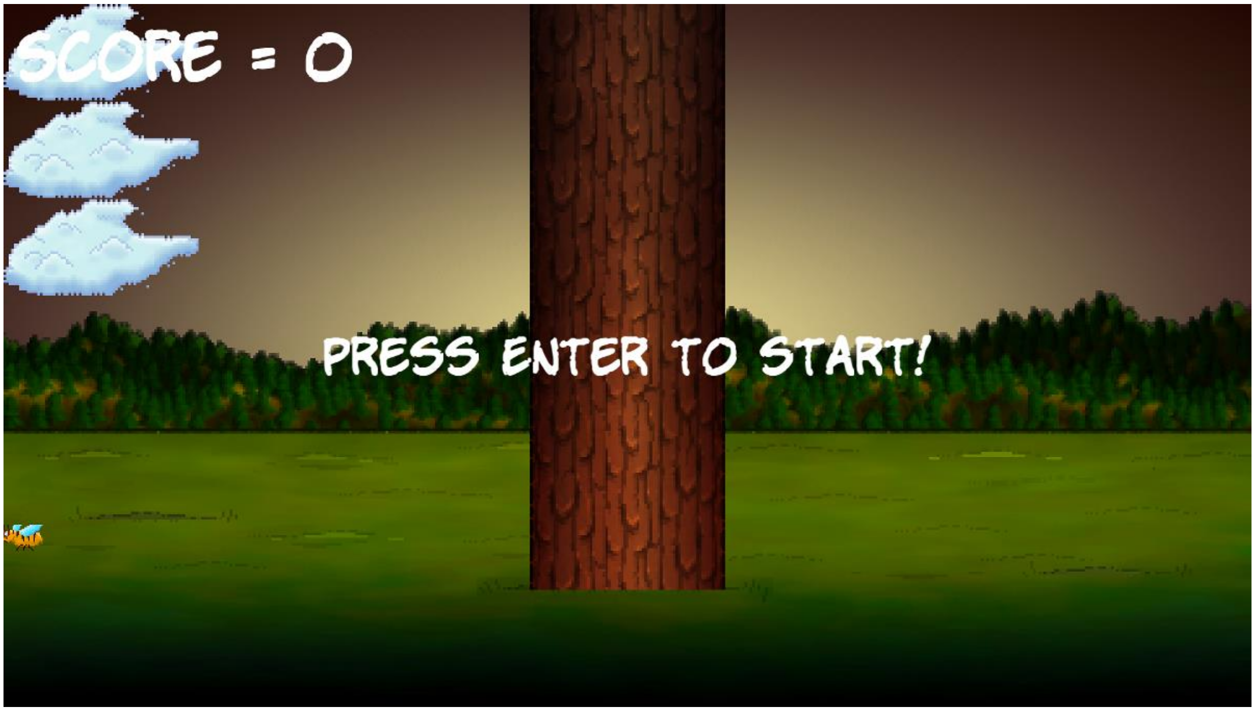


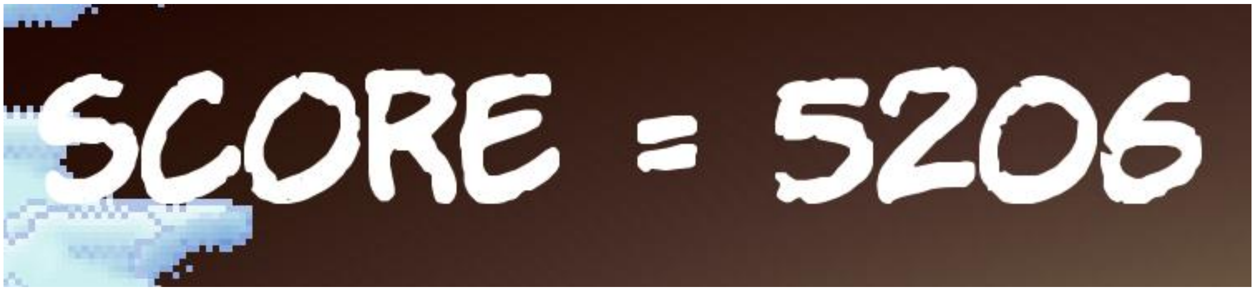
Chapter 2: Variables, Operators, and Decisions – Animating Sprites

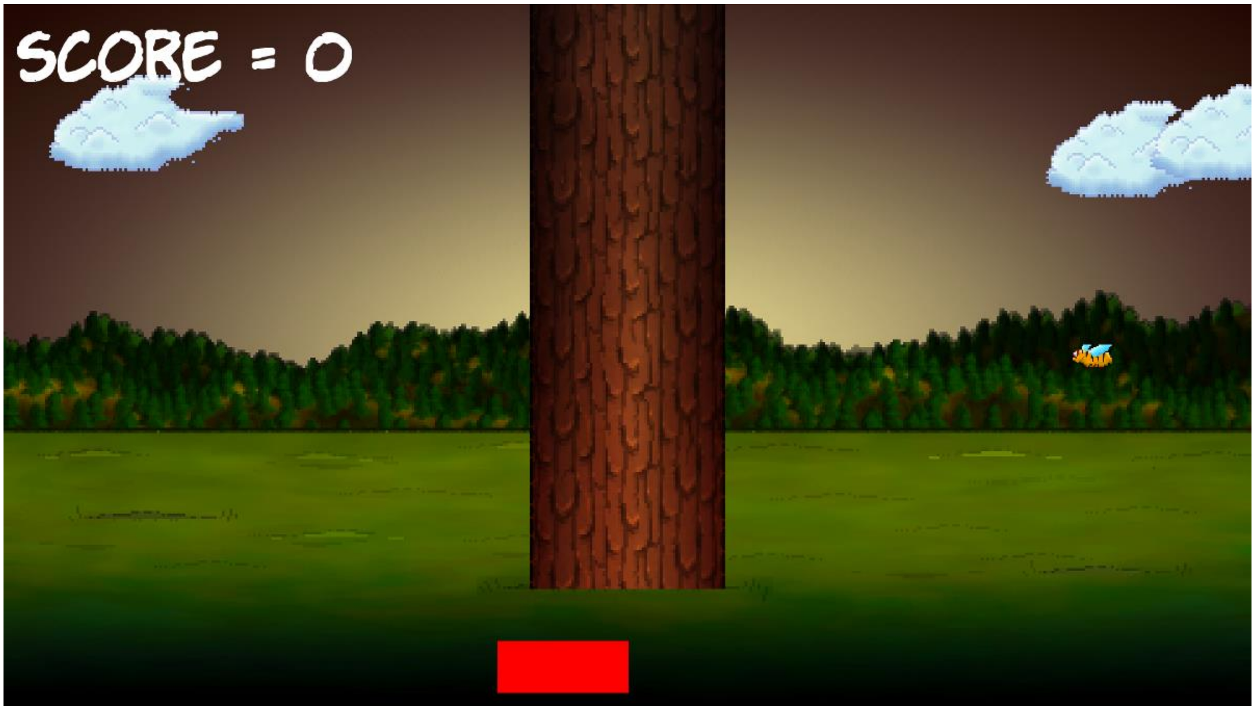




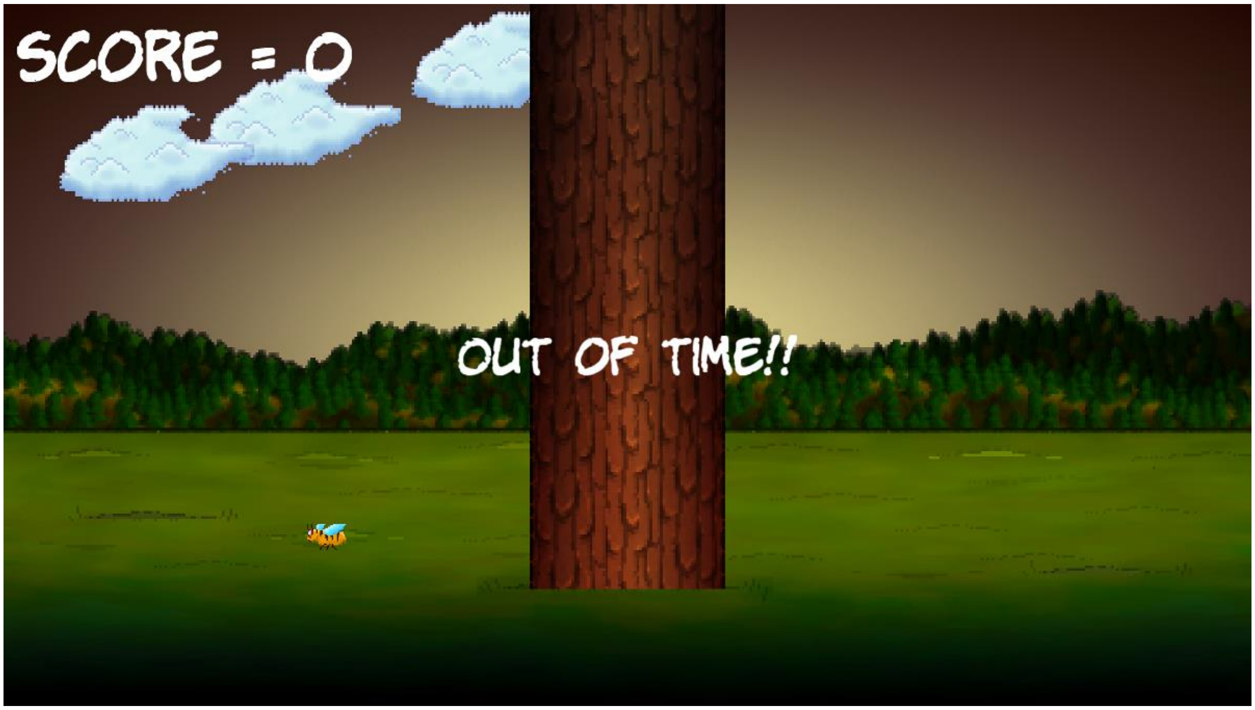
Chapter 3: C++ Strings, SFML Time – Player Input, and HUD







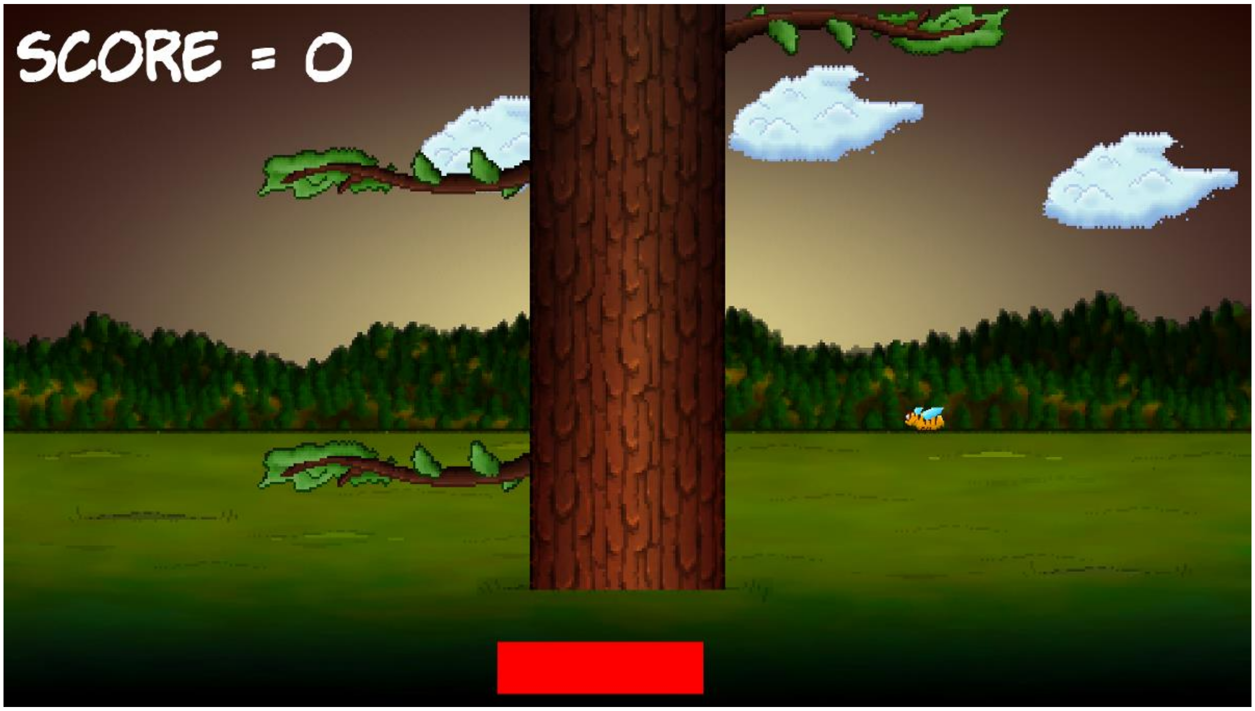
SCORE = 0



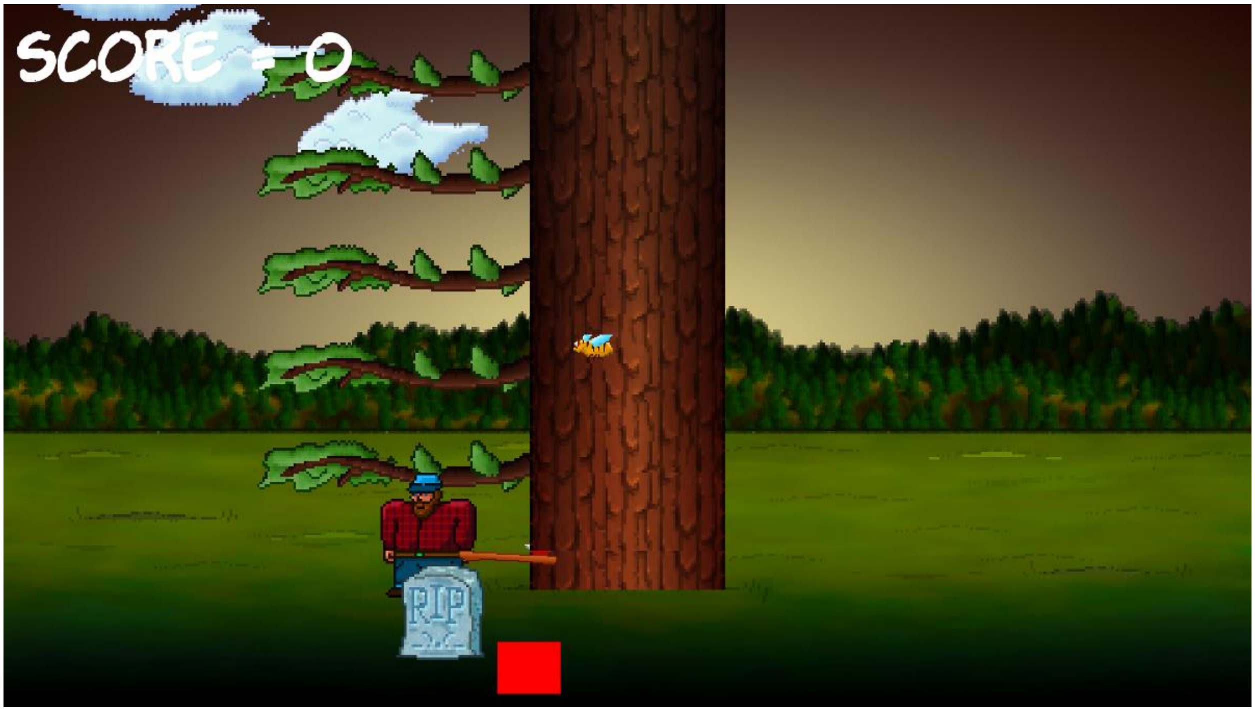
SCORE = 0

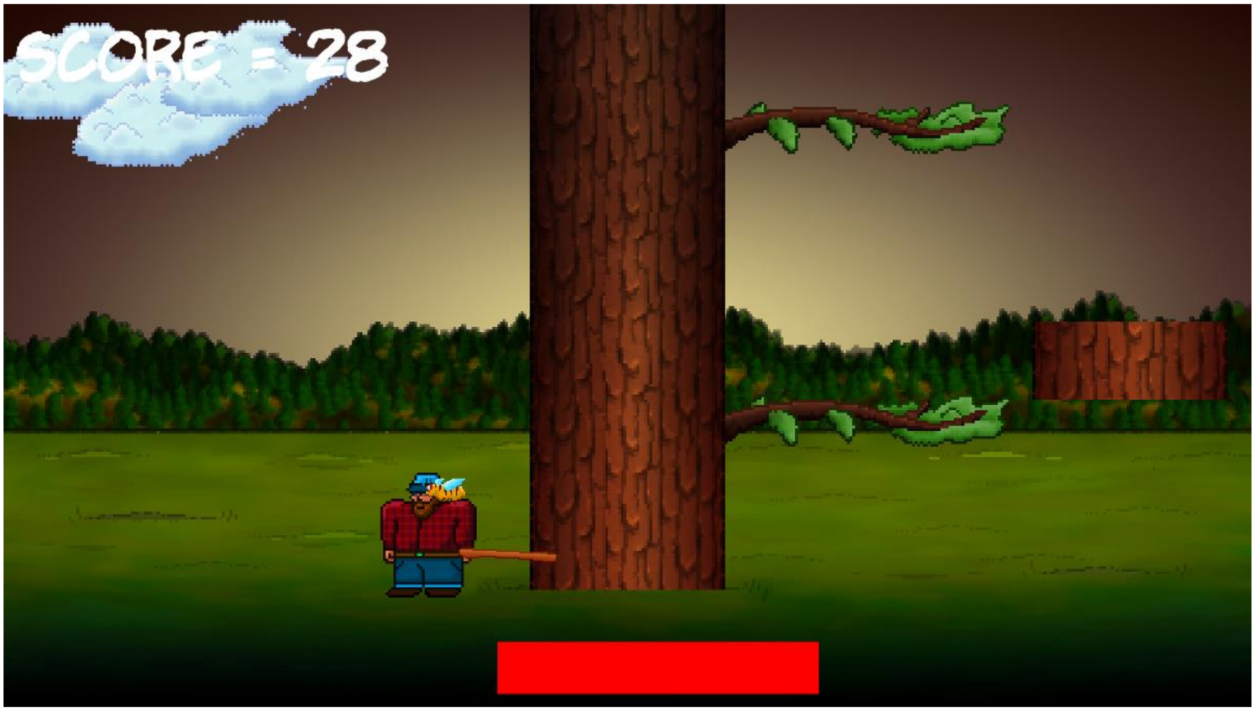
OUT OF TIME!!

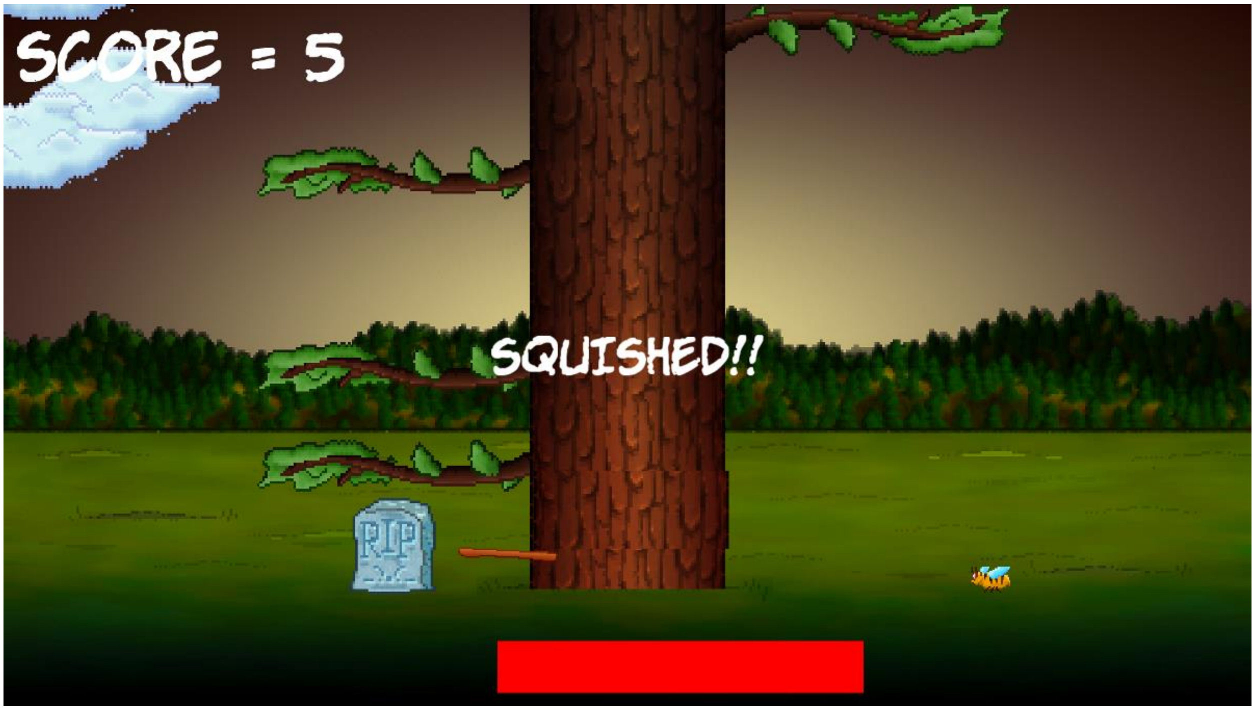
Chapter 4: Loops, Arrays, Switch, Enumerations and Functions – Implementing Game Mechanics

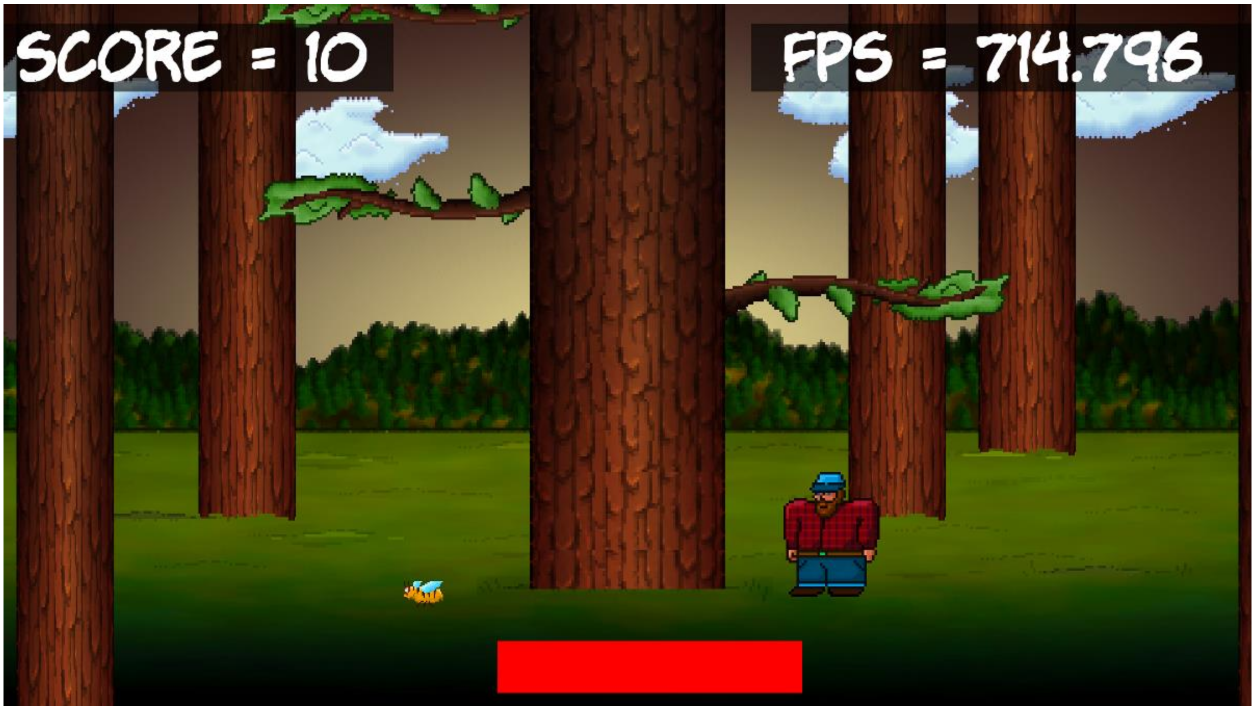


Chapter 5: Collisions, Sound, and, End Conditions – Making the Game Playable

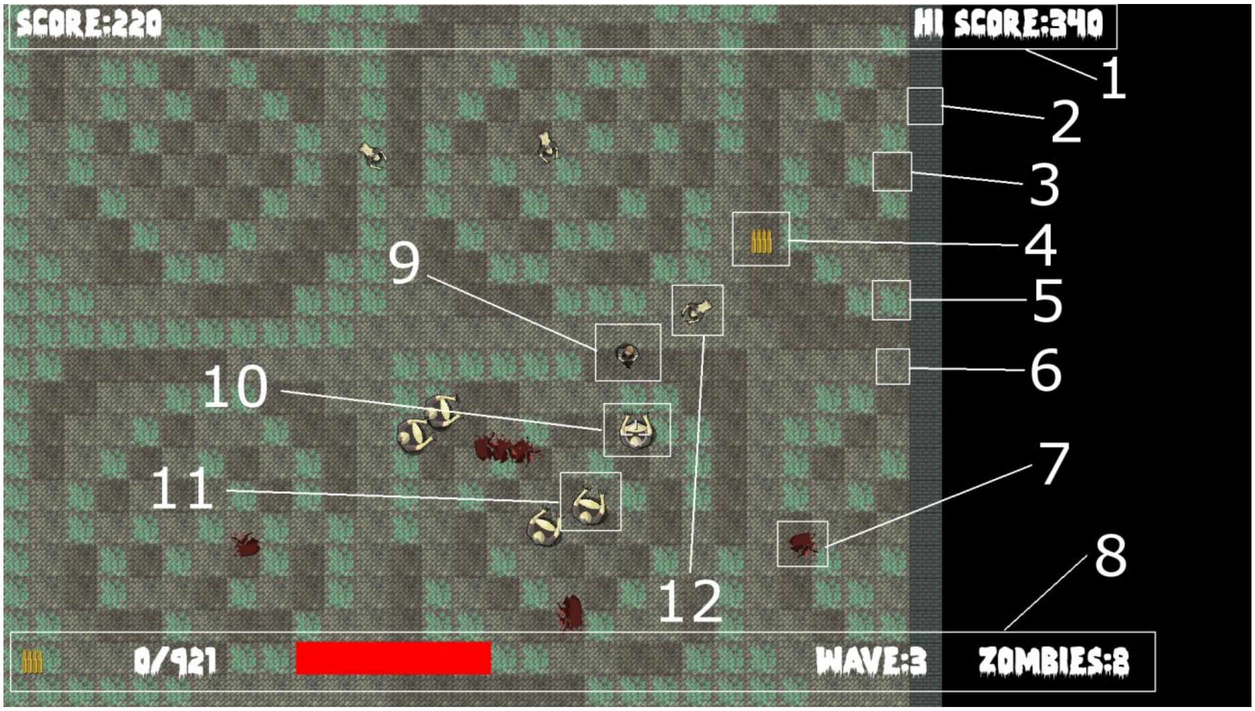


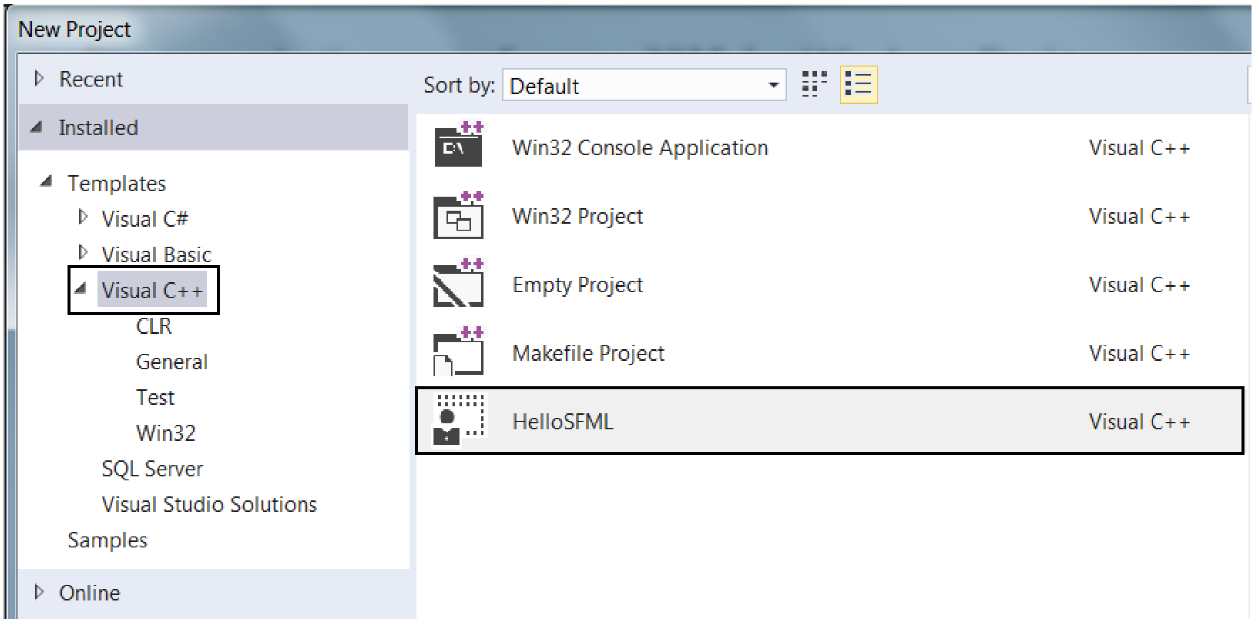



SCORE = 5
SQUISHED!!
RIP


SCORE = 10
FPS = 714.796

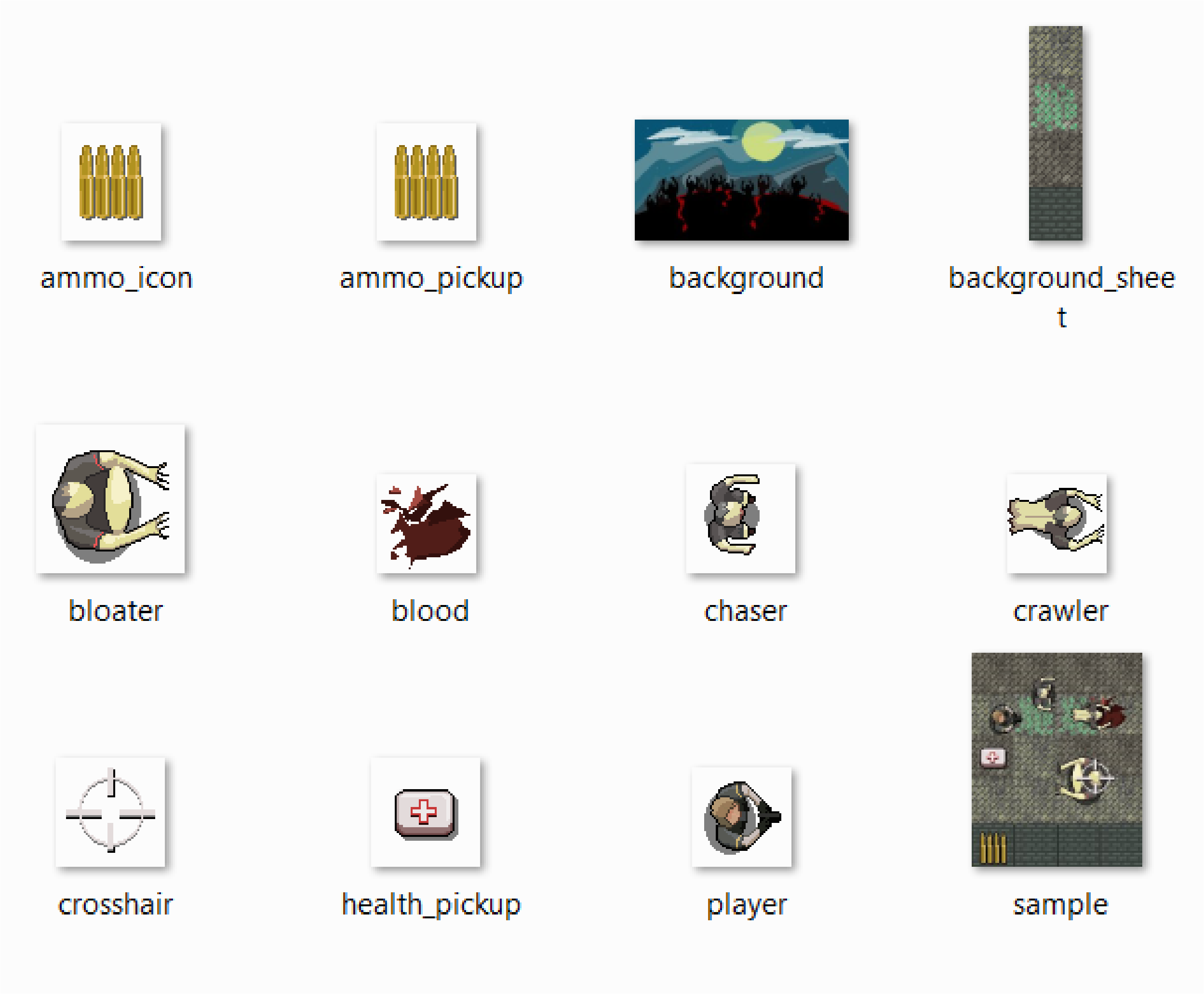
Chapter 6: Object-Oriented Programming, Classes, and SFML Views

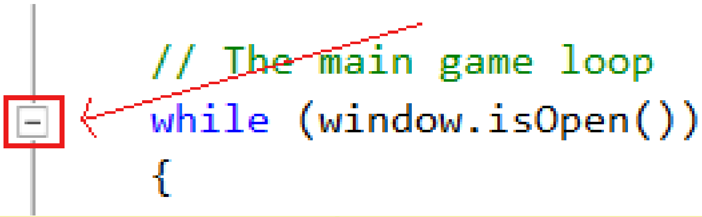




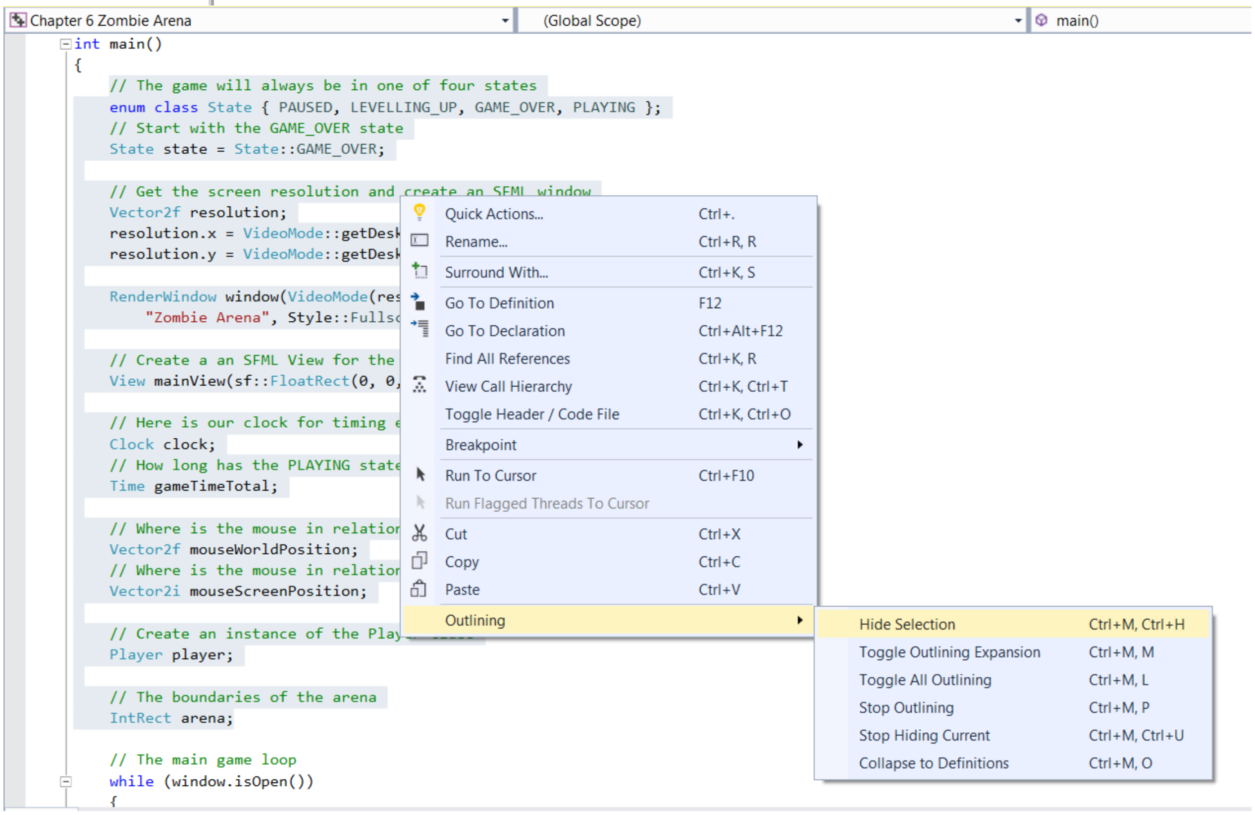
Sort by: Default

Win32 Console Application	Visual C++
Win32 Project	Visual C++
Empty Project	Visual C++
Makefile Project	Visual C++
HelloSFML	Visual C++


ammo_icon
ammo_pickup
background
background_sheet
bloater
blood
chaser
crawler
crosshair
health_pickup
player
sample

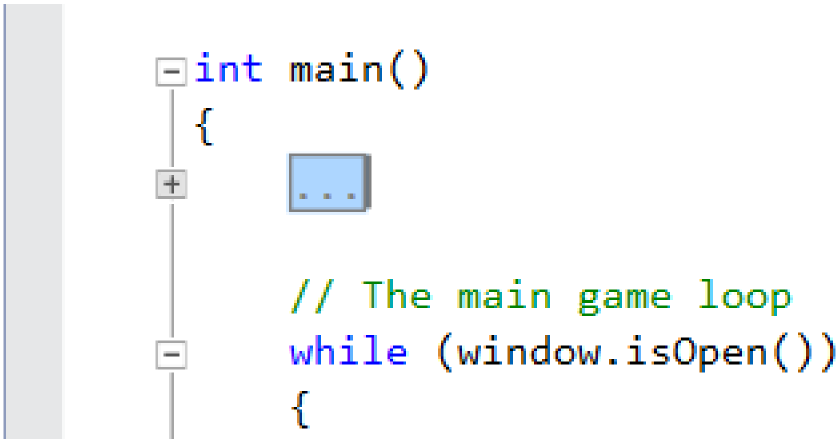


```
// The main game loop
while (window.isOpen())
{
```

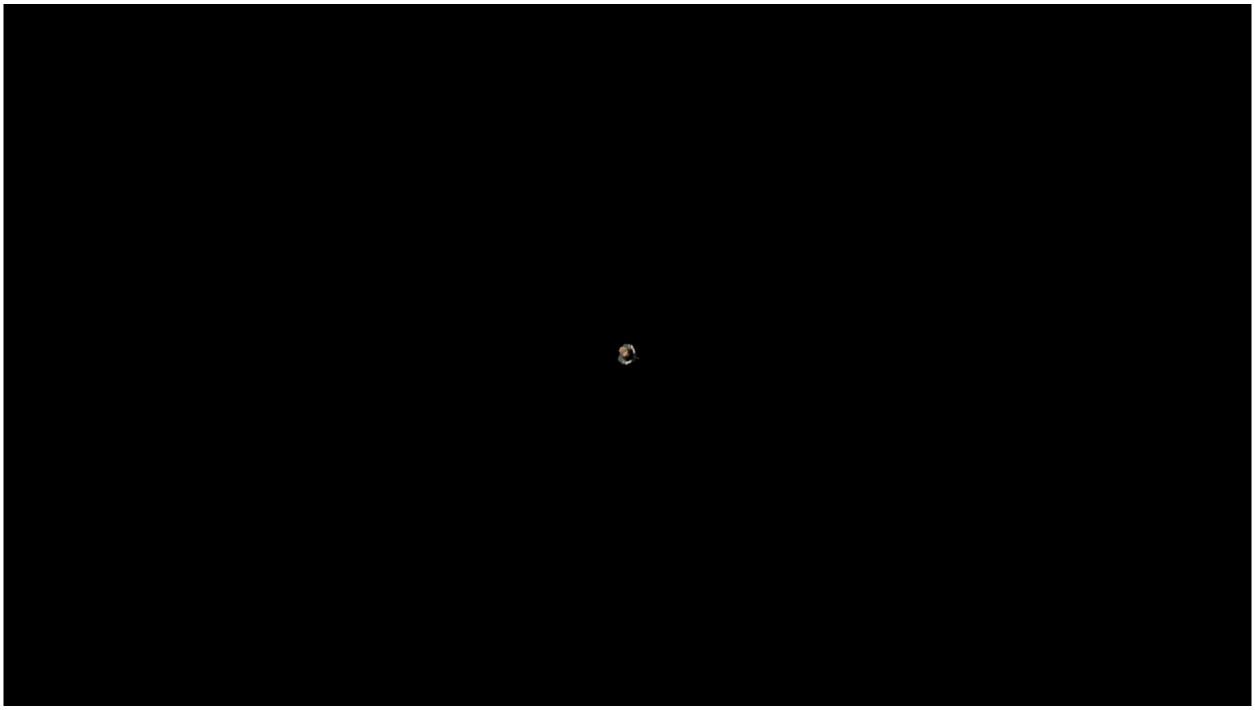


Chapter 6 Zombie Arena
(Global Scope)
main()
Quick Actions... Ctrl+.
Rename... Ctrl+R, R
Surround With... Ctrl+K, S
Go To Definition F12
Go To Declaration Ctrl+Alt+F12
Find All References Ctrl+K, R
View Call Hierarchy Ctrl+K, Ctrl+T
Toggle Header / Code File Ctrl+K, Ctrl+O
Breakpoint
Run To Cursor Ctrl+F10
Run Flagged Threads To Cursor
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Outlining
Hide Selection Ctrl+M, Ctrl+H
Toggle Outlining Expansion Ctrl+M, M
Toggle All Outlining Ctrl+M, L
Stop Outlining Ctrl+M, P
Stop Hiding Current Ctrl+M, Ctrl+U
Collapse to Definitions Ctrl+M, O

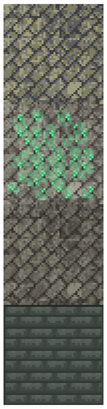
```
int main()
{
    // The game will always be in one of four states
    enum class State { PAUSED, LEVELLING_UP, GAME_OVER, PLAYING };
    // Start with the GAME_OVER state
    State state = State::GAME_OVER;

    // Get the screen resolution and create an SFML window
    Vector2f resolution;
    resolution.x = VideoMode::getDesk
    resolution.y = VideoMode::getDesk

    RenderWindow window(VideoMode(res
        "Zombie Arena", Style::Fullsc

    // Create a an SFML View for the
    View mainView(sf::FloatRect(0, 0,

    // Here is our clock for timing e
    Clock clock;
    // How long has the PLAYING state
    Time gameTimeTotal;

    // Where is the mouse in relation
    Vector2f mouseWorldPosition;
    // Where is the mouse in relation
    Vector2i mouseScreenPosition;

    // Create an instance of the Play
    Player player;

    // The boundaries of the arena
    IntRect arena;

    // The main game loop
    while (window.isOpen())
    {
```

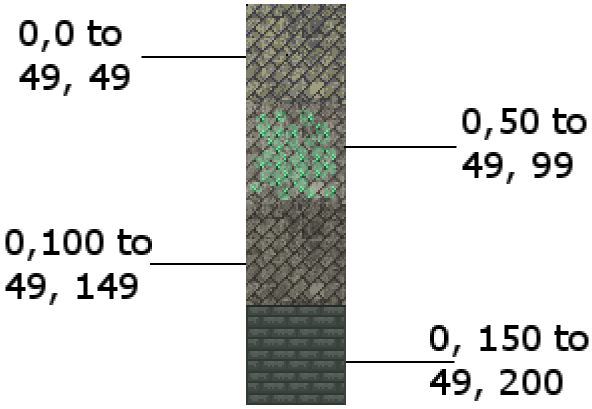


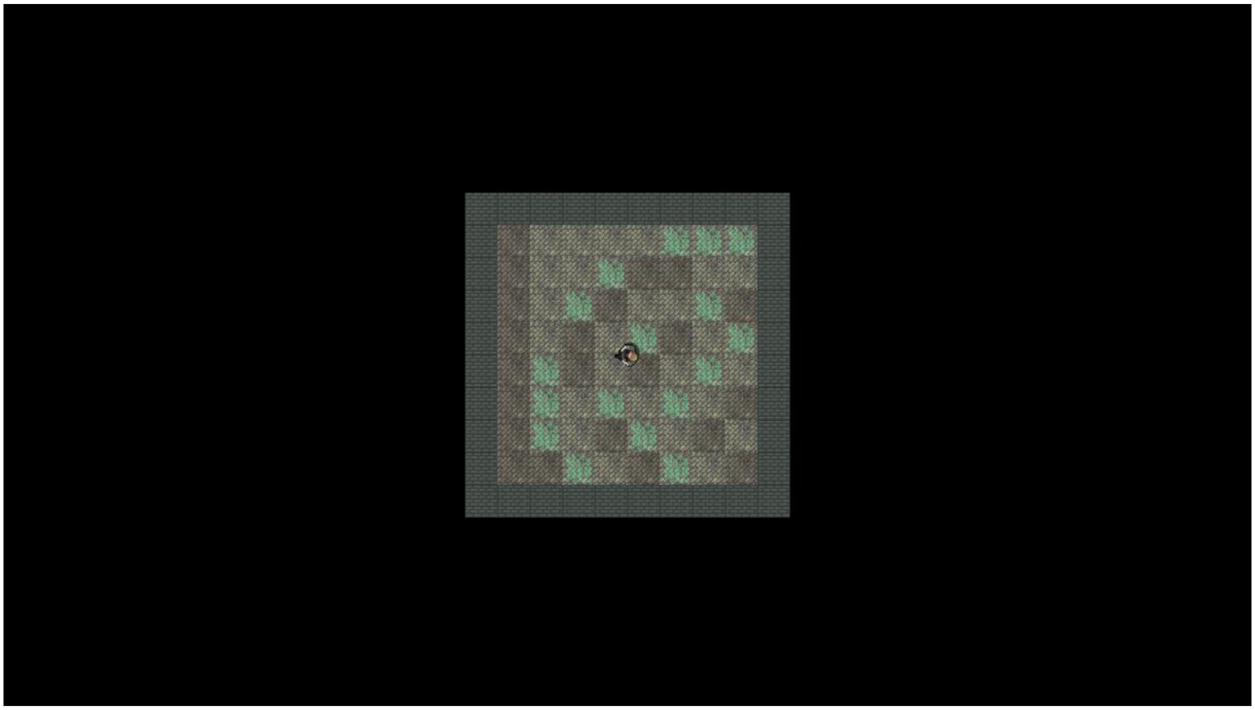
```
int main()
{
        ...

        // The main game loop
        while (window.isOpen())
        {
```



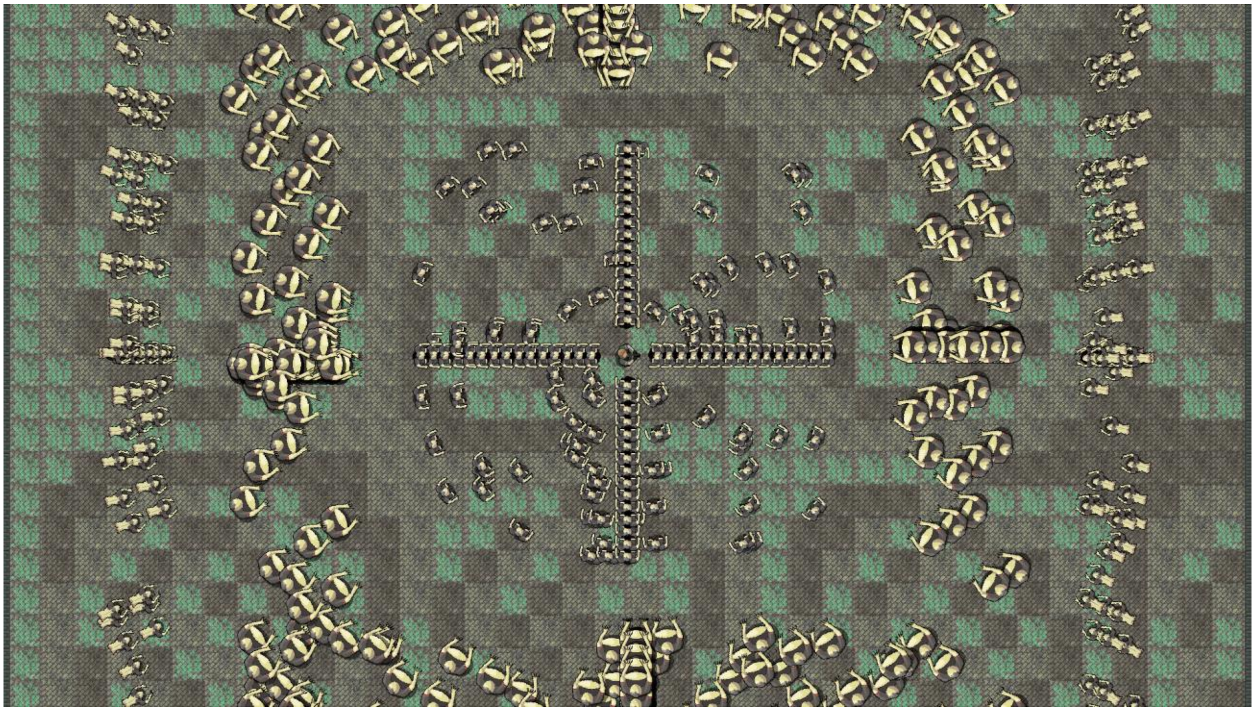
Chapter 7: C++ References, Sprite Sheets and Vertex Arrays



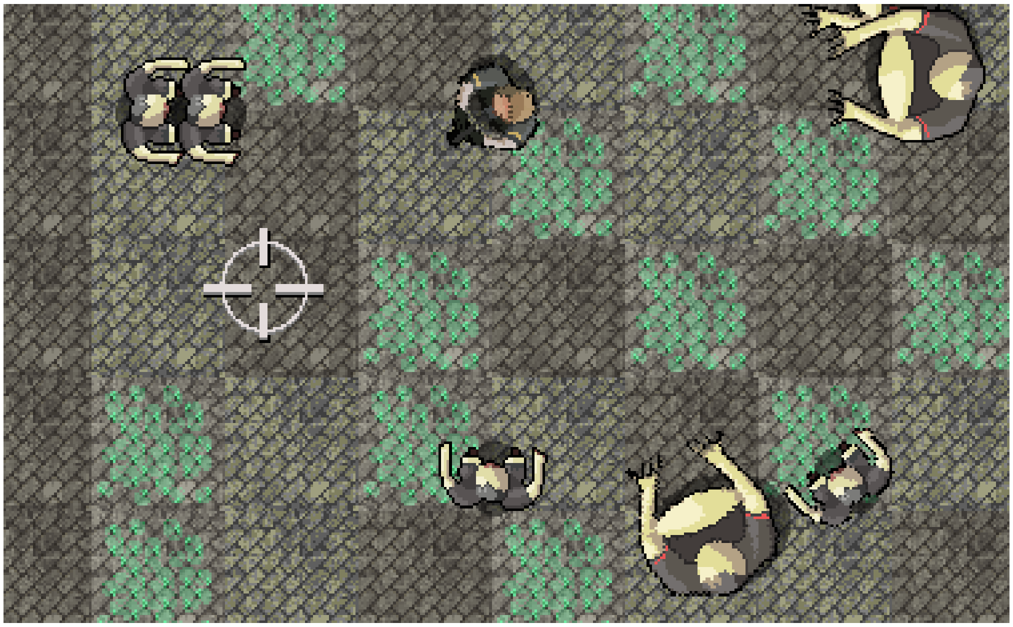


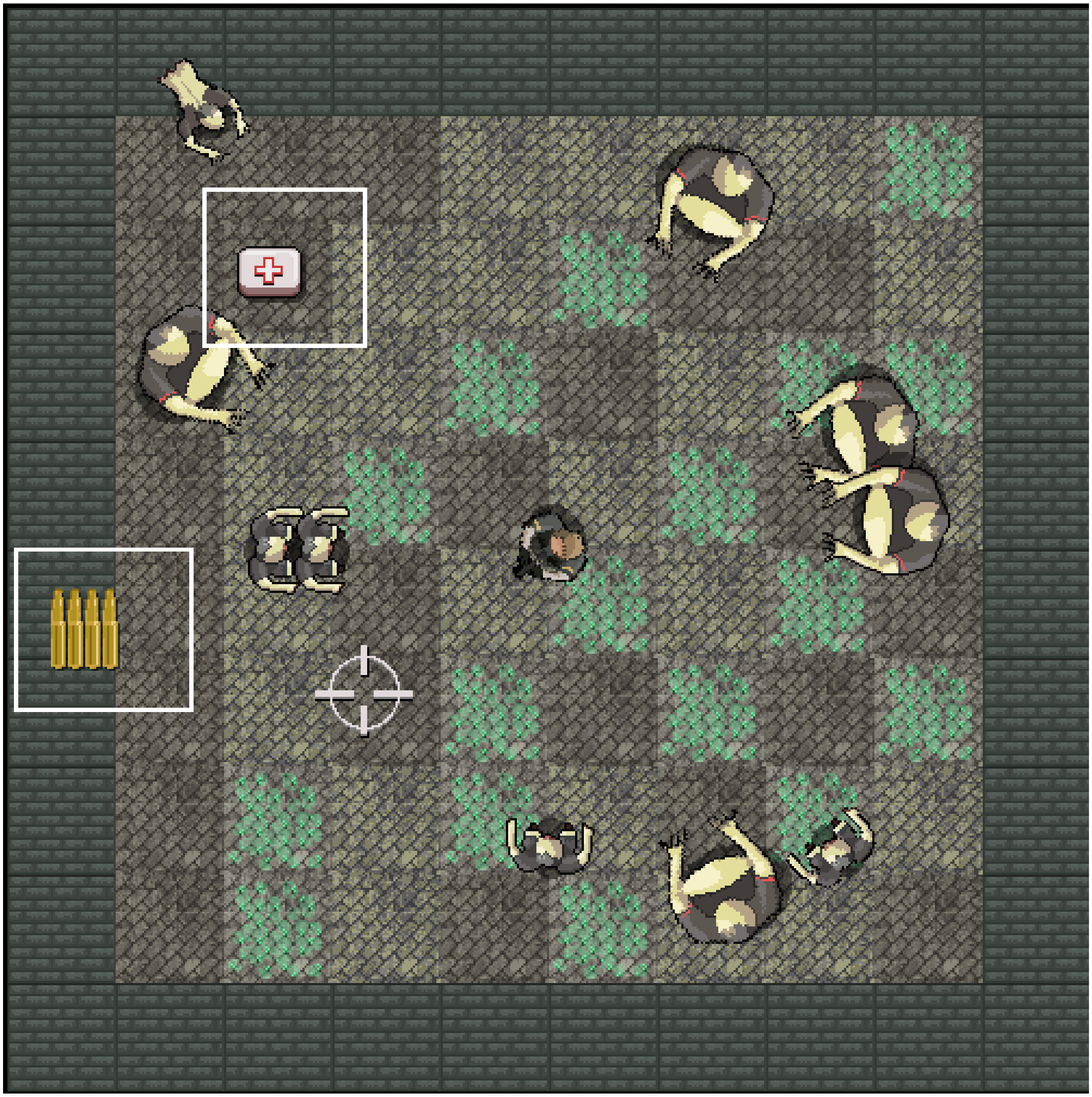


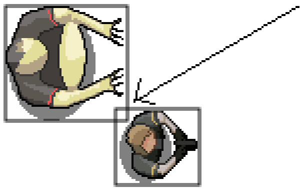
Chapter 8: Pointers, the Standard Template Library, and Texture Management



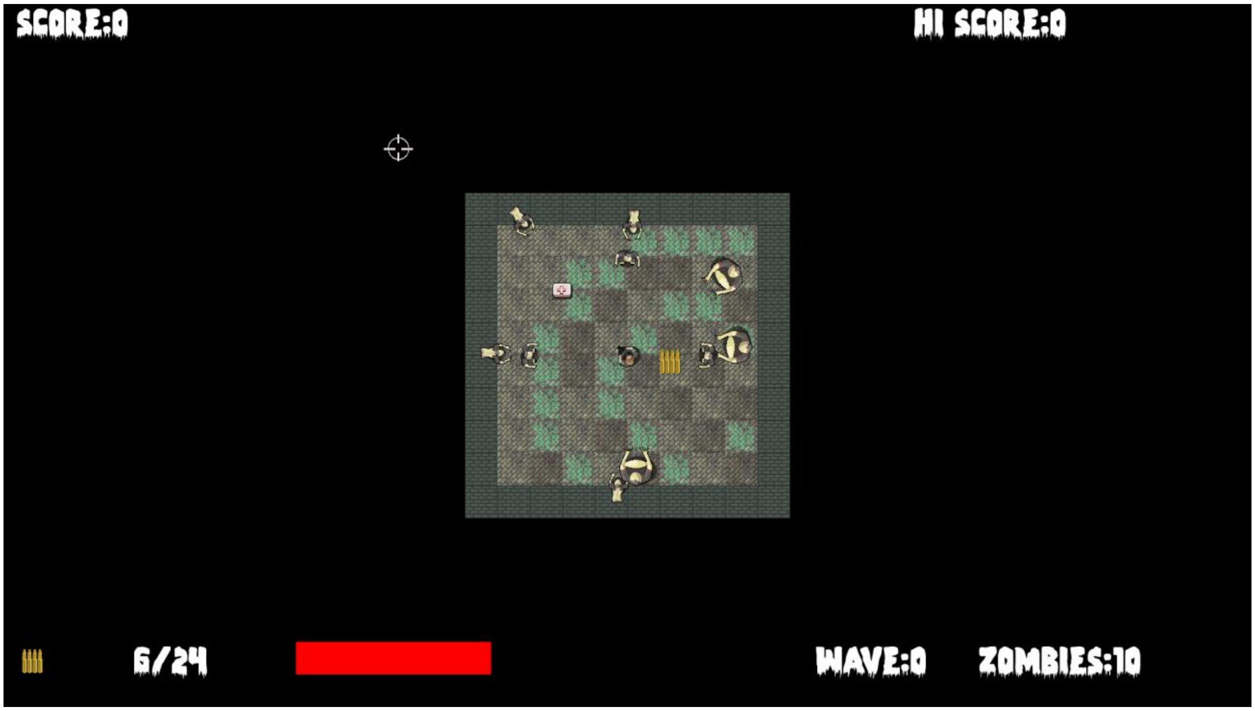
Chapter 9: Collision Detection, Pickups and Bullets

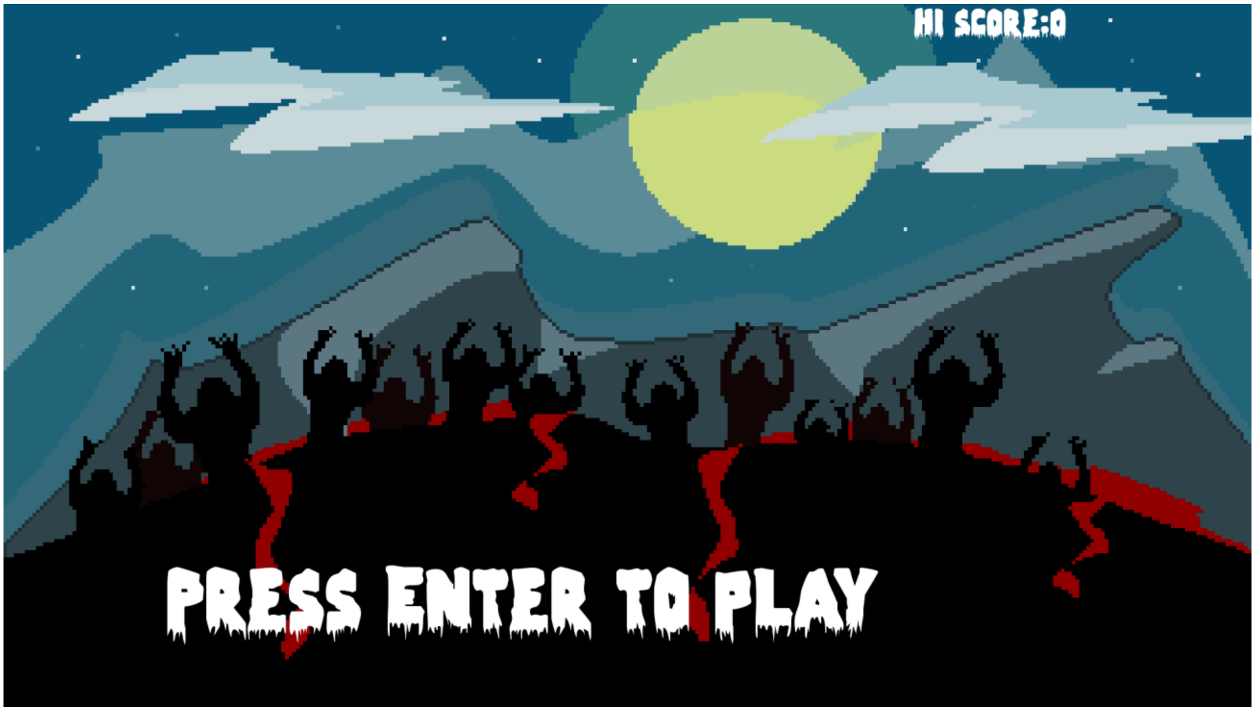


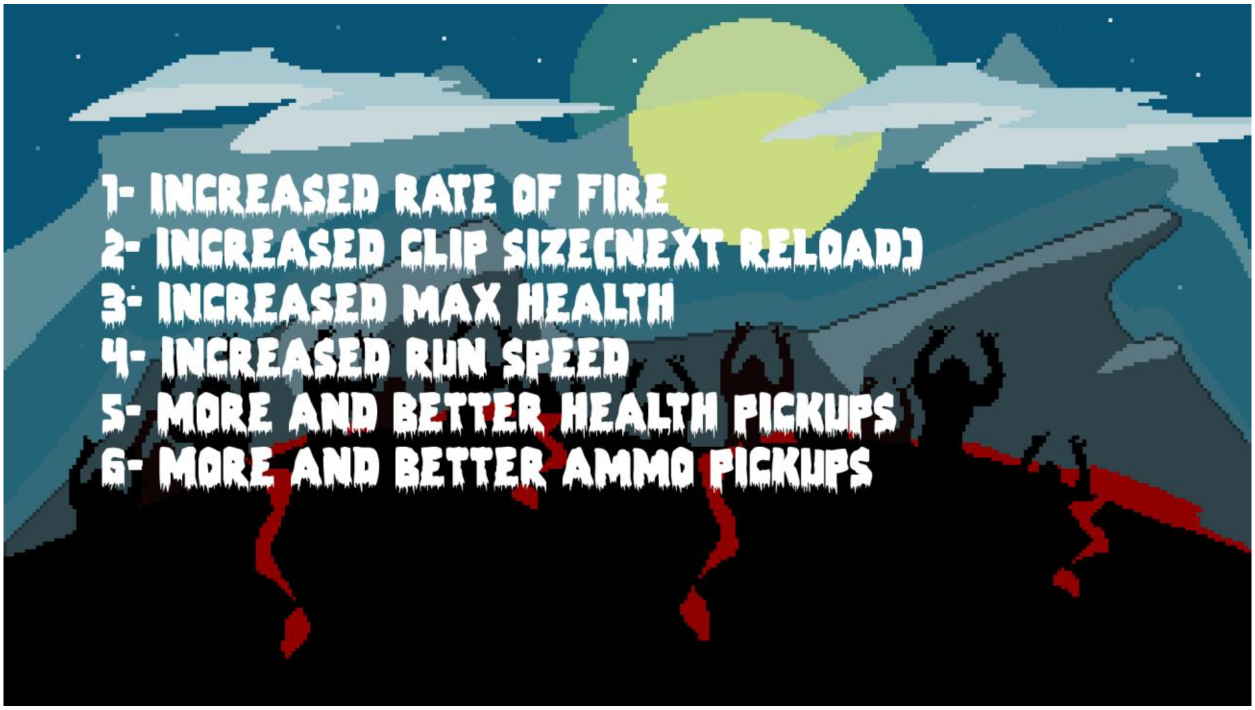




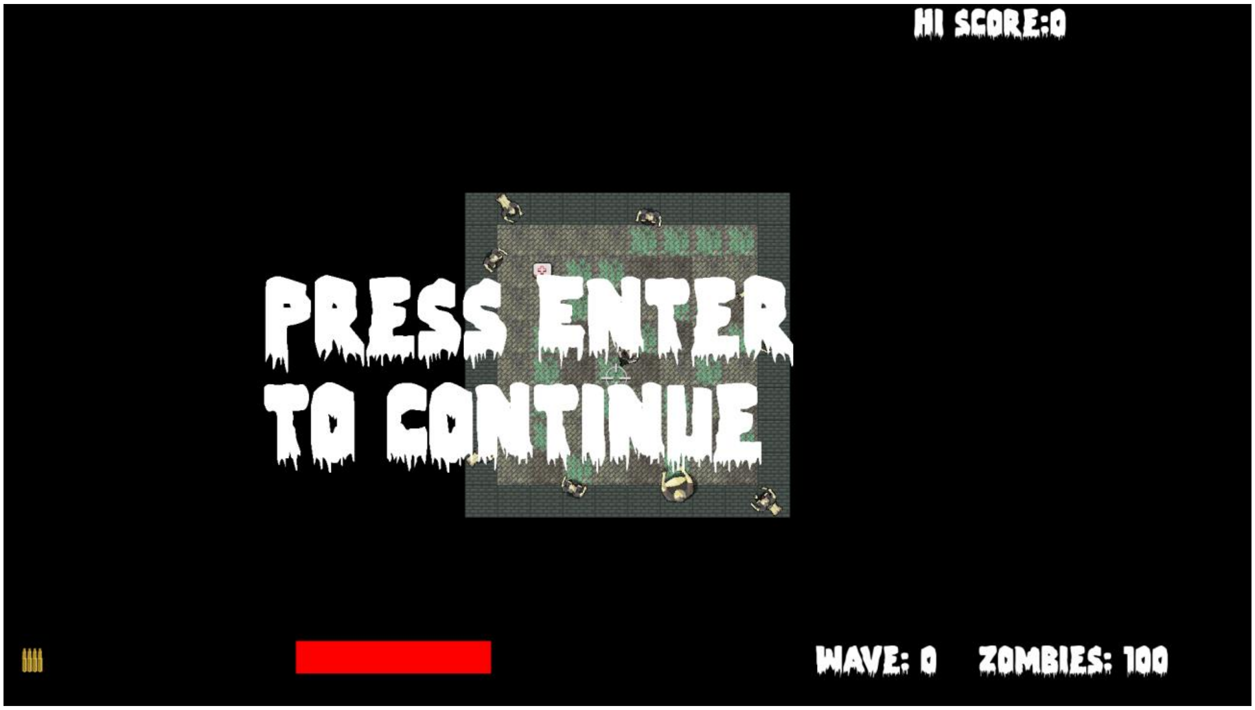
Chapter 10: Layering Views and Implementing the HUD







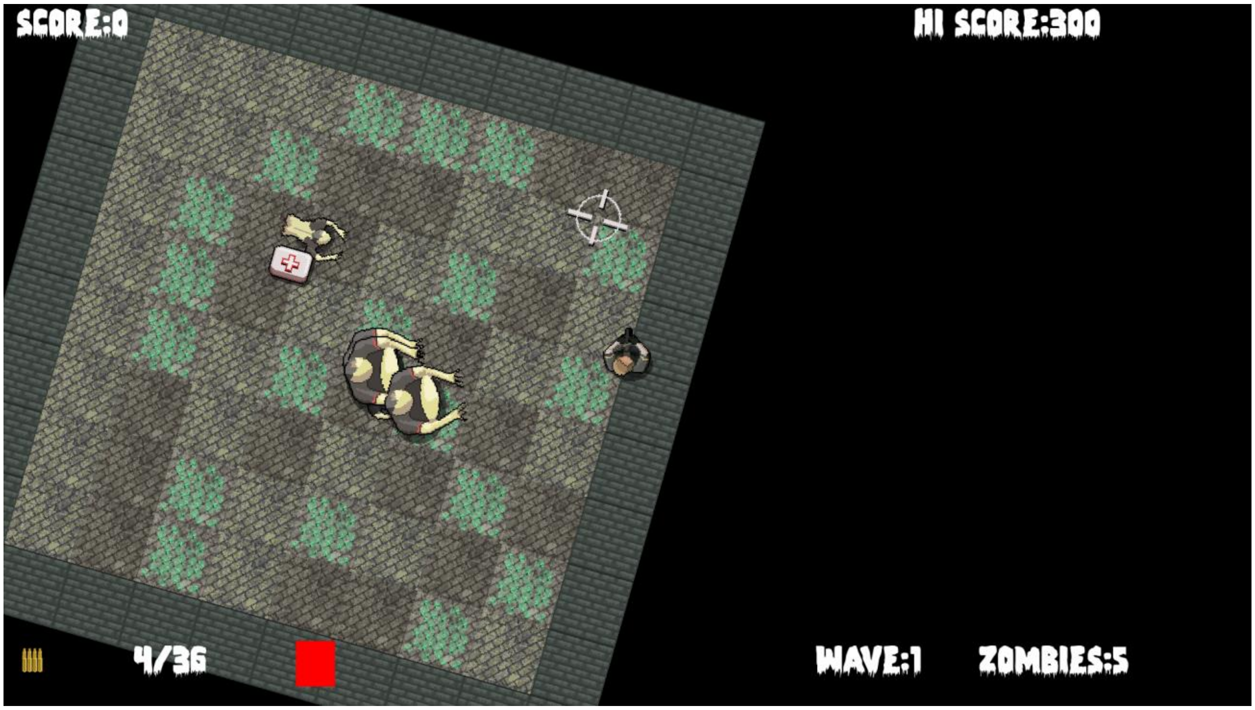
1- INCREASED RATE OF FIRE
2- INCREASED CLIP SIZE(NEXT RELOAD)
3- INCREASED MAX HEALTH
4- INCREASED RUN SPEED
5- MORE AND BETTER HEALTH PICKUPS
6- MORE AND BETTER AMMO PICKUPS



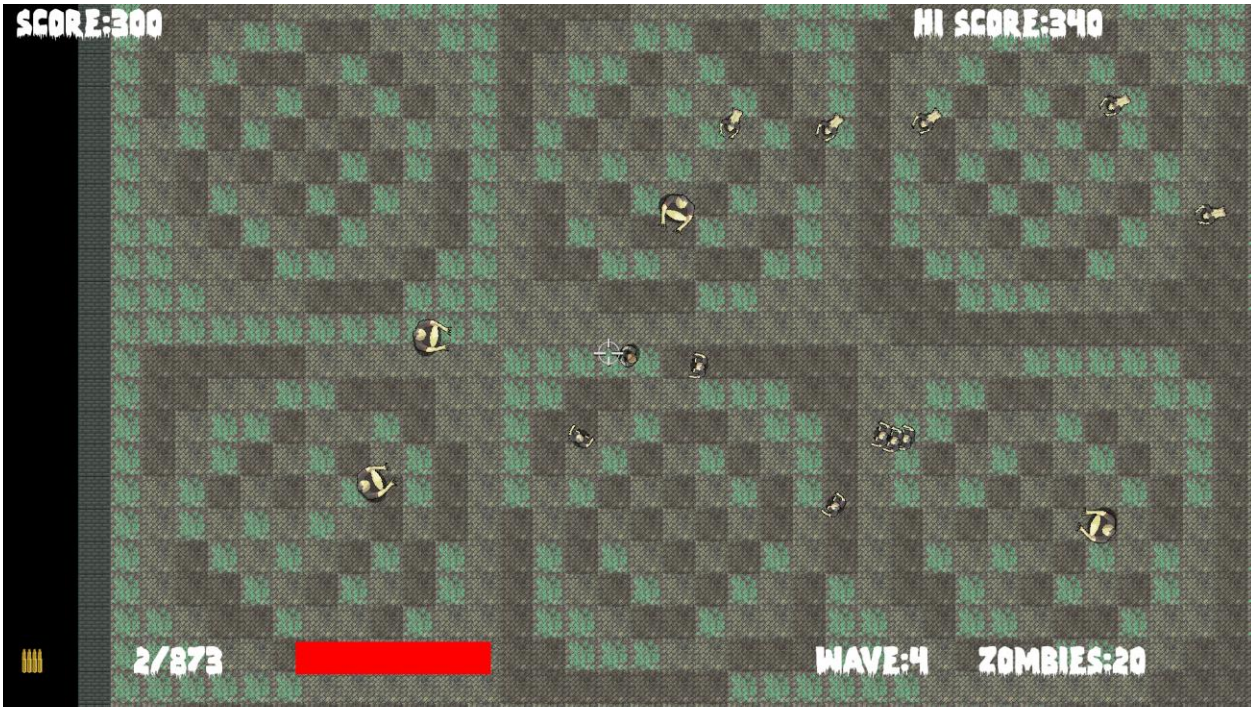
HI SCORE:0

PRESS ENTER
TO CONTINUE

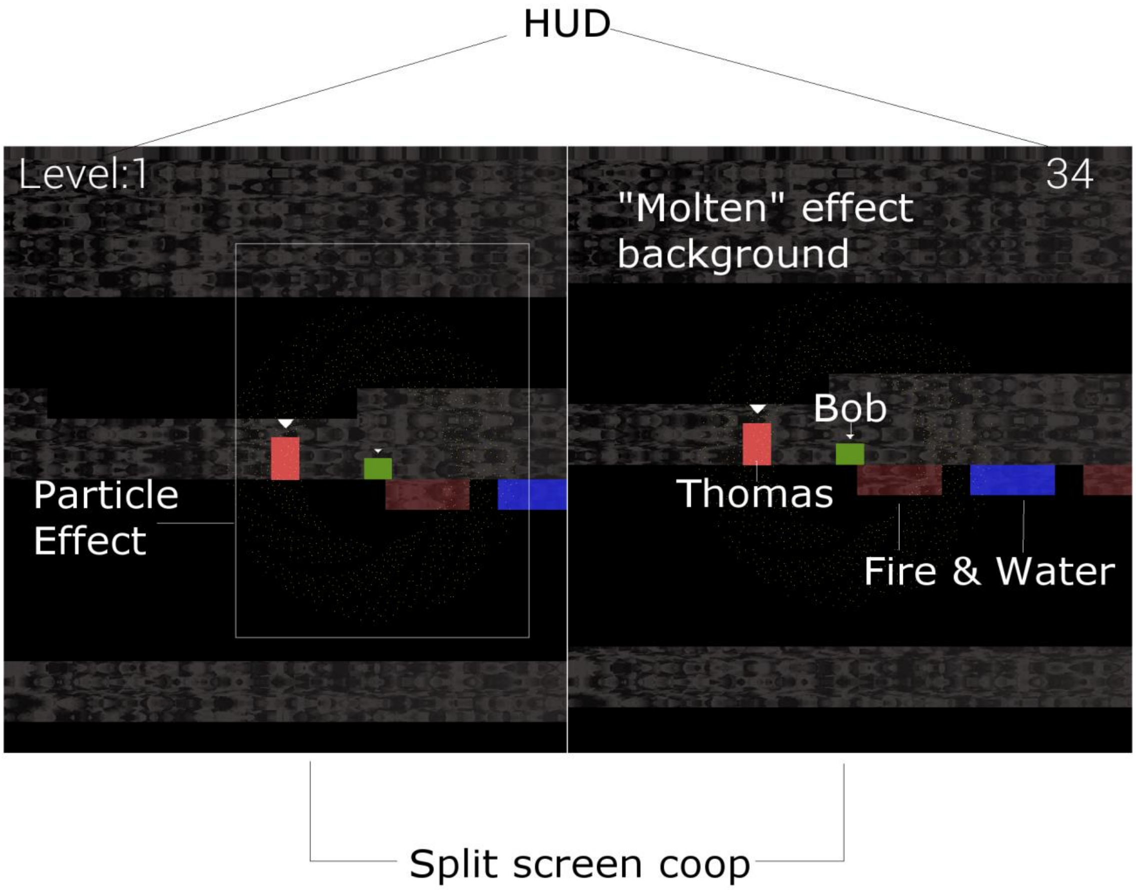
WAVE: 0 ZOMBIES: 100

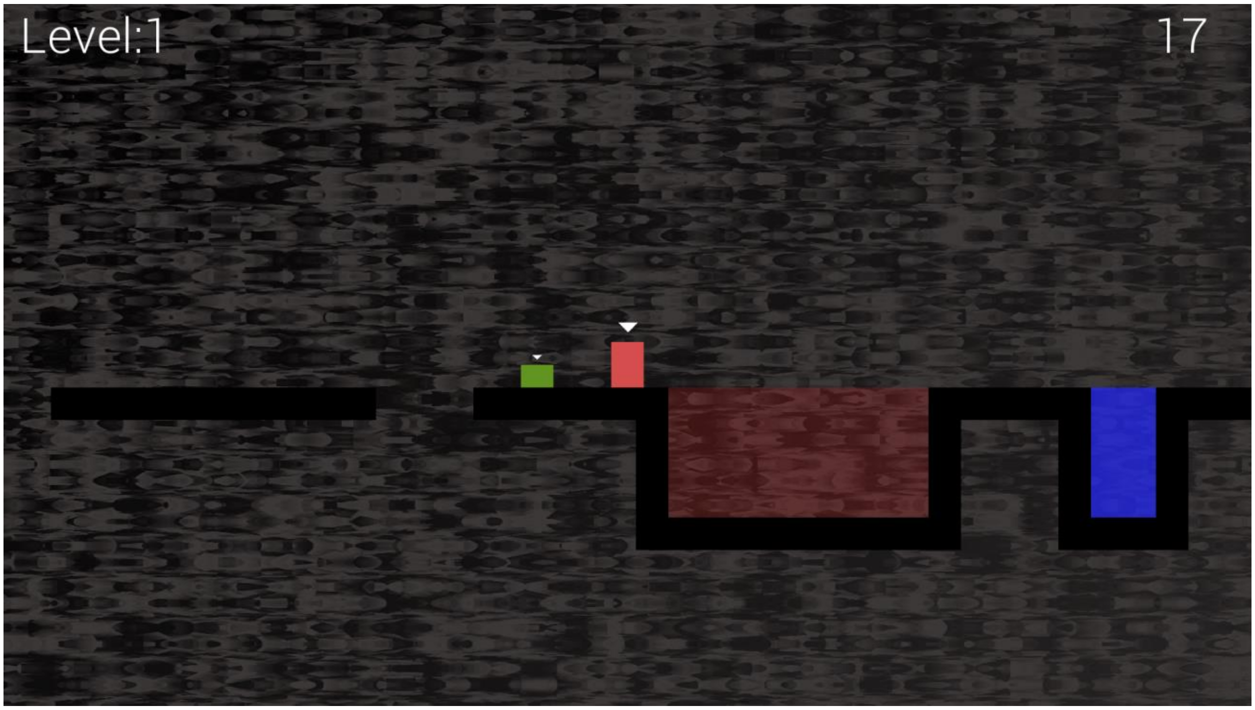

SCORE:0
HI SCORE:300
4/36
WAVE:1
ZOMBIES:5

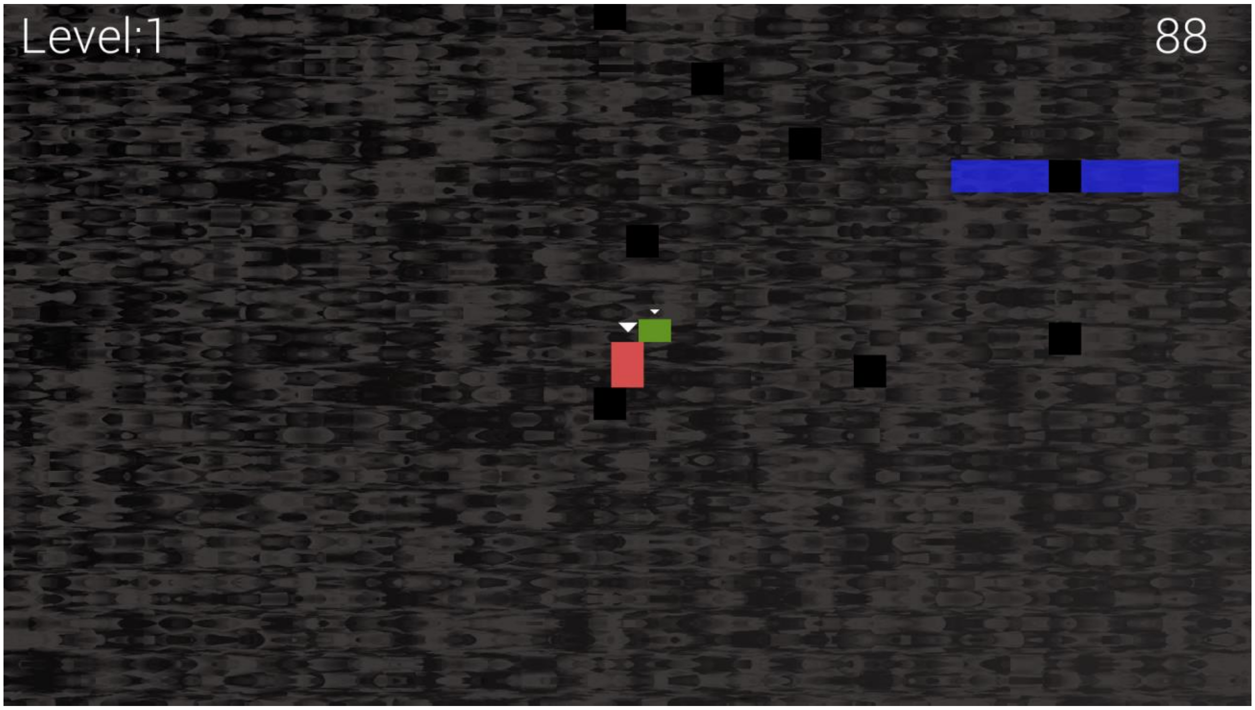
Chapter 11: Sound Effects, File I/O and Finishing the Game

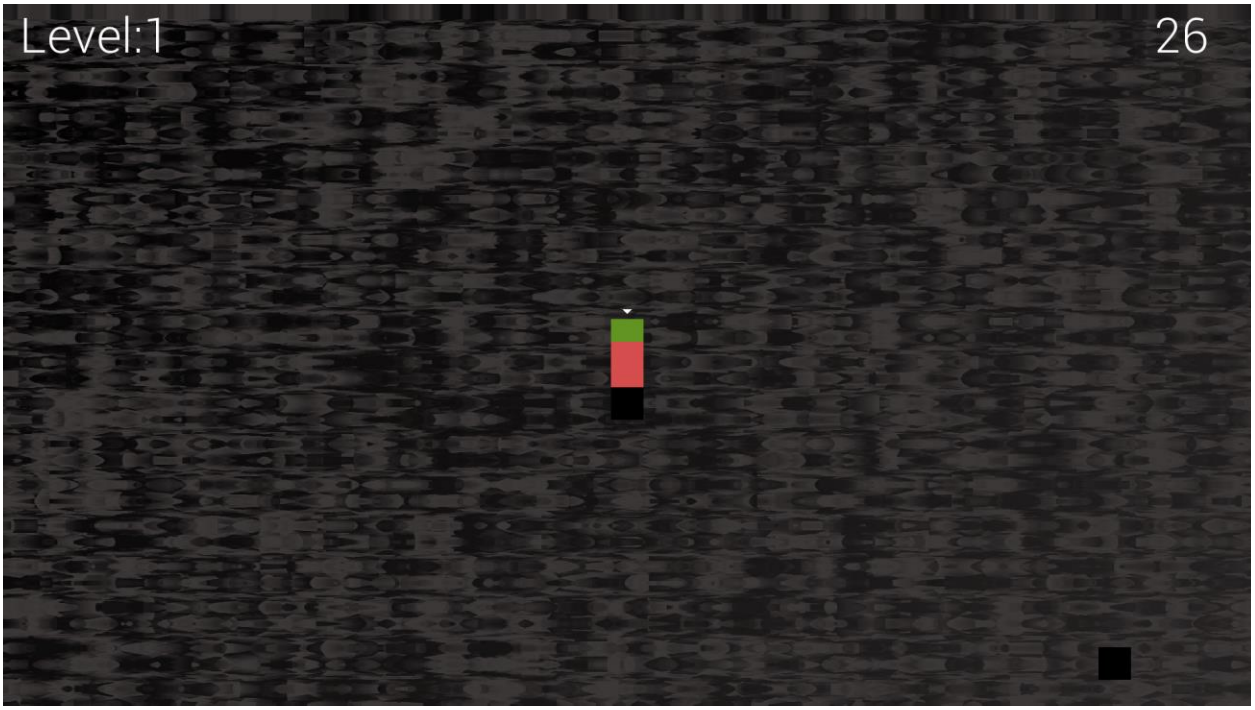


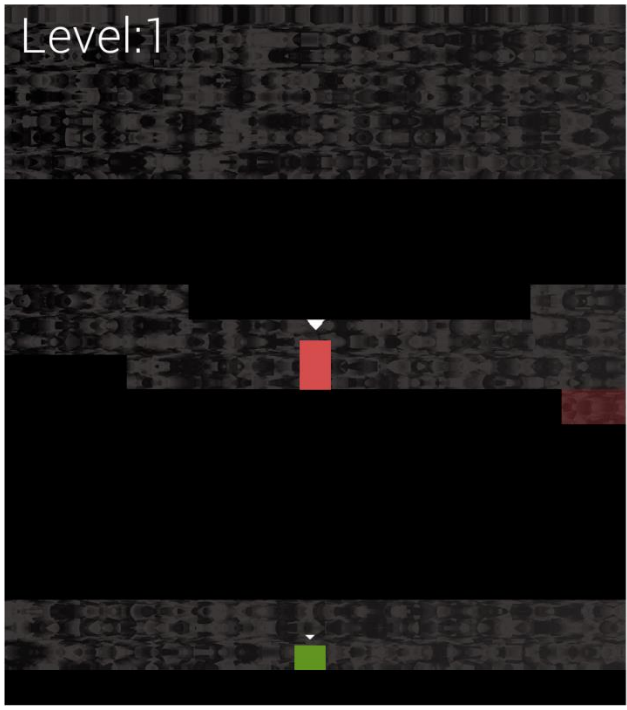
Chapter 12: Abstraction and Code Management – Making Better Use of OOP

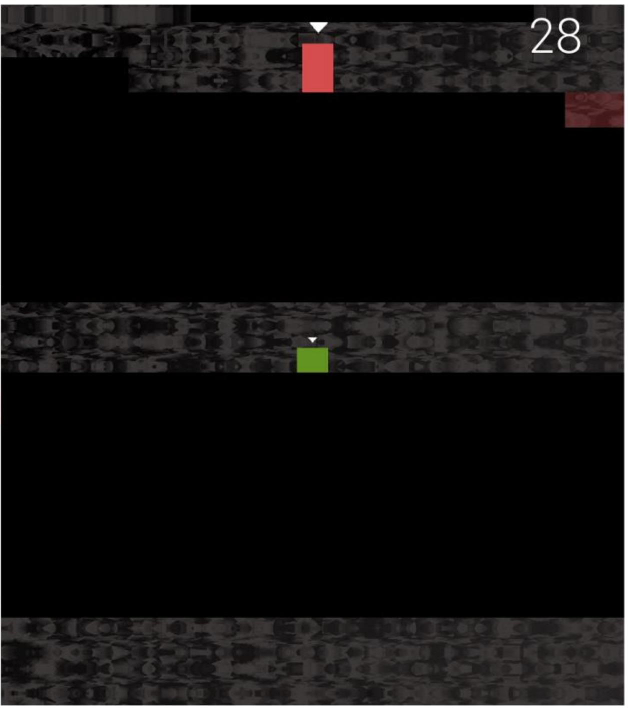


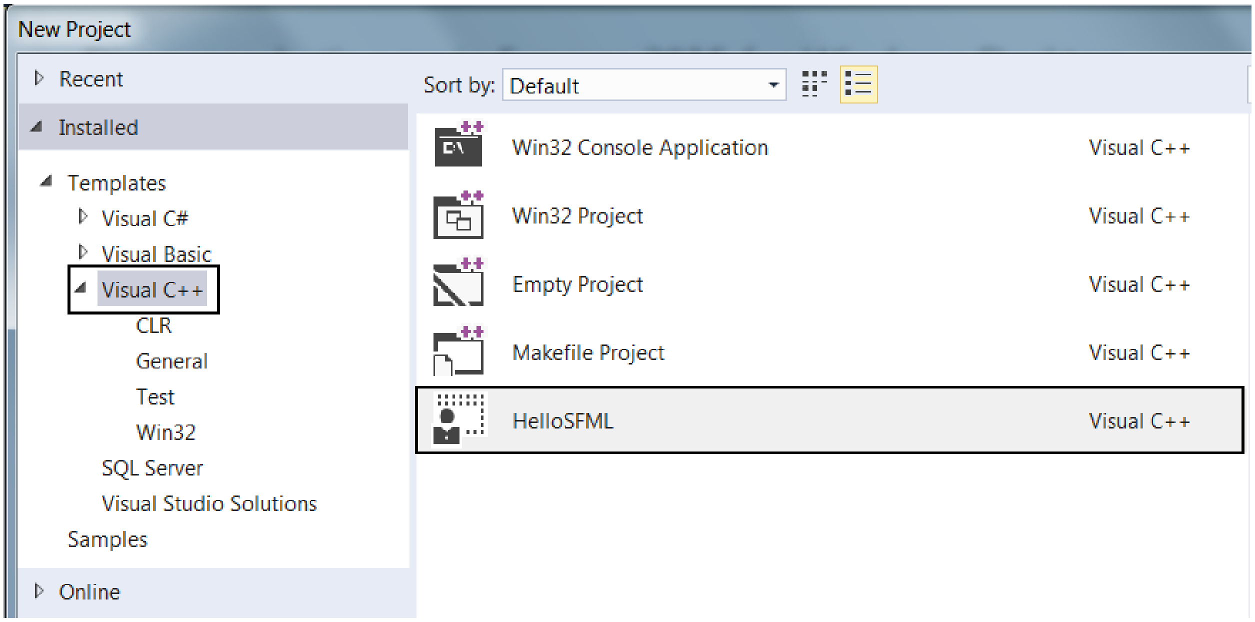

Level:1
17

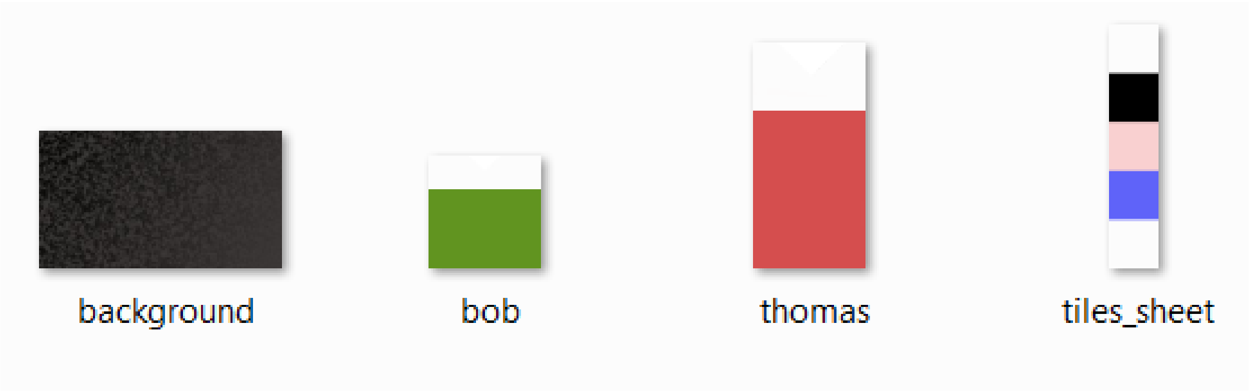

Level:1
88

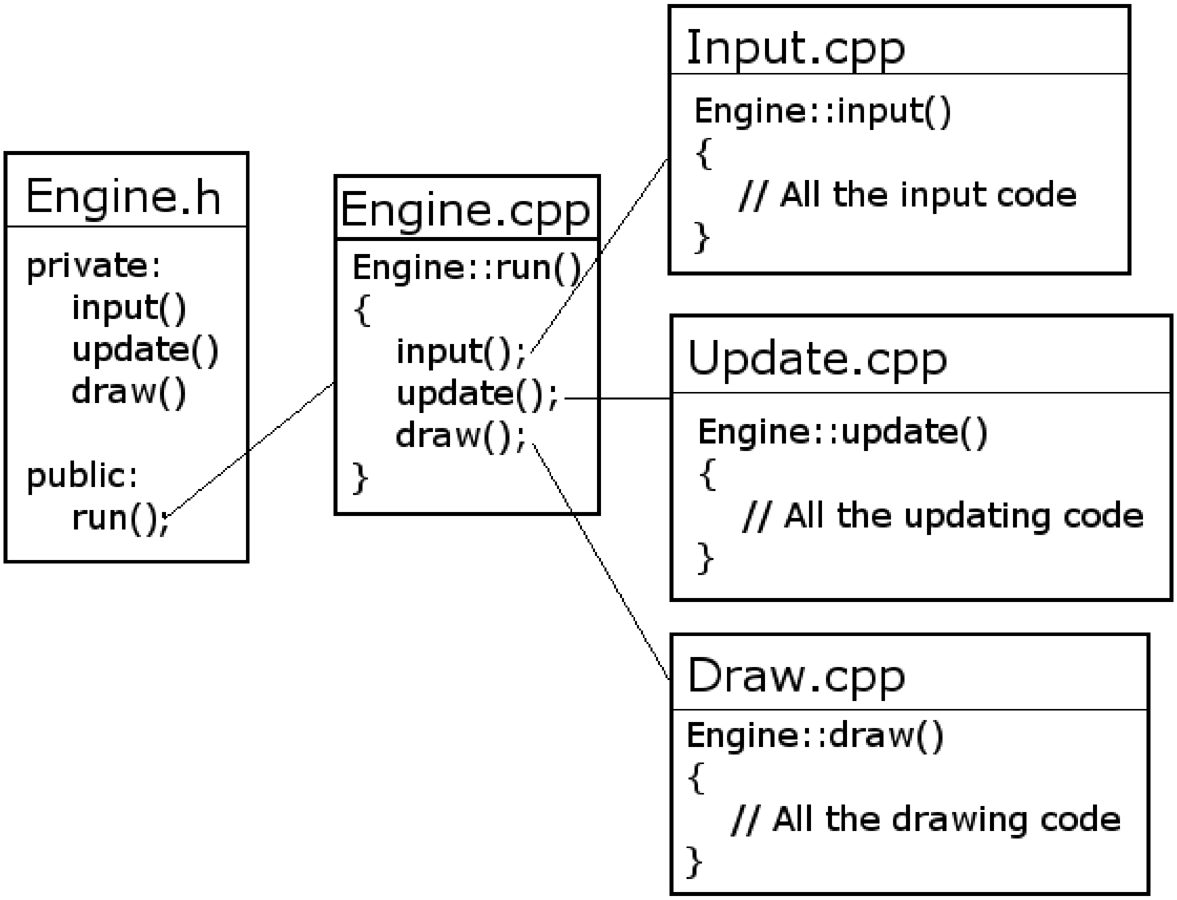

Level:1
26


Level:1


28


New Project
Recent
Installed
Templates
Visual C#
Visual Basic
Visual C++
CLR
General
Test
Win32
SQL Server
Visual Studio Solutions
Samples
Online
Sort by: Default
Win32 Console Application Visual C++
Win32 Project Visual C++
Empty Project Visual C++
Makefile Project Visual C++
HelloSFML Visual C++


background
bob
thomas
tiles_sheet



Engine.h

```
private:
    input()
    update()
    draw()

public:
    run();
```

Engine.cpp

```
Engine::run()
{
    input();
    update();
    draw();
}
```

Input.cpp

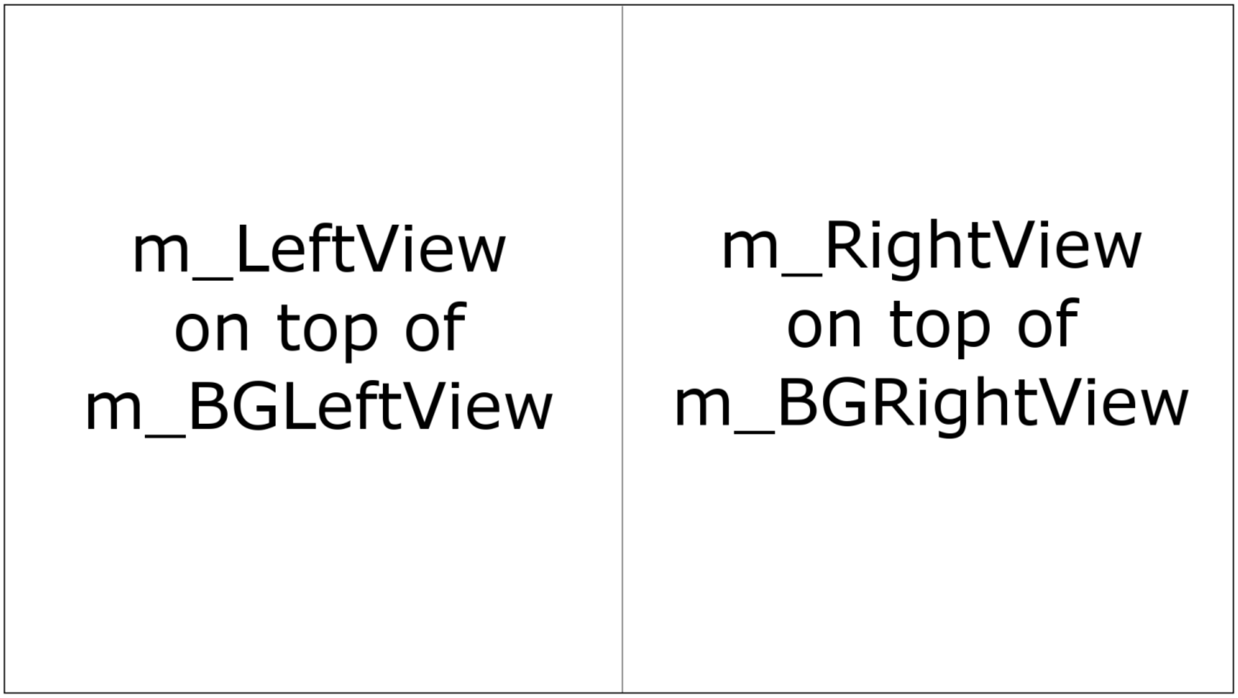
```
Engine::input()
{
    // All the input code
}
```

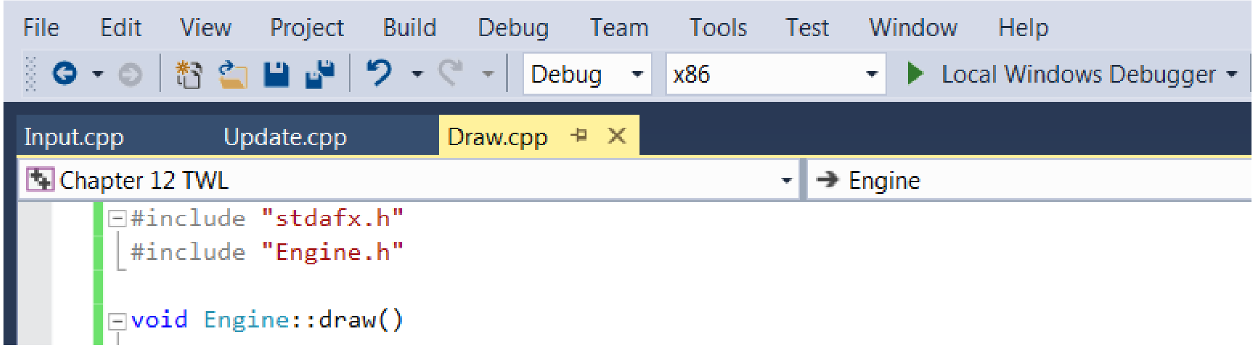
Update.cpp

```
Engine::update()
{
    // All the updating code
}
```

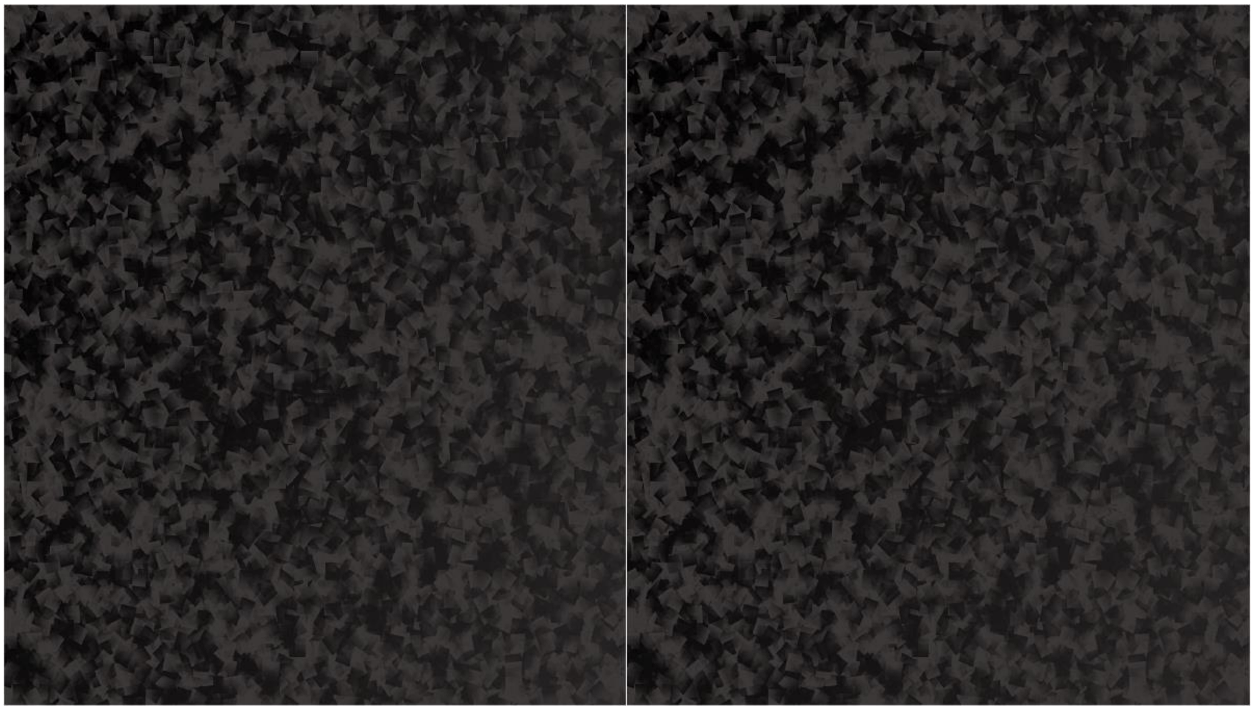
Draw.cpp

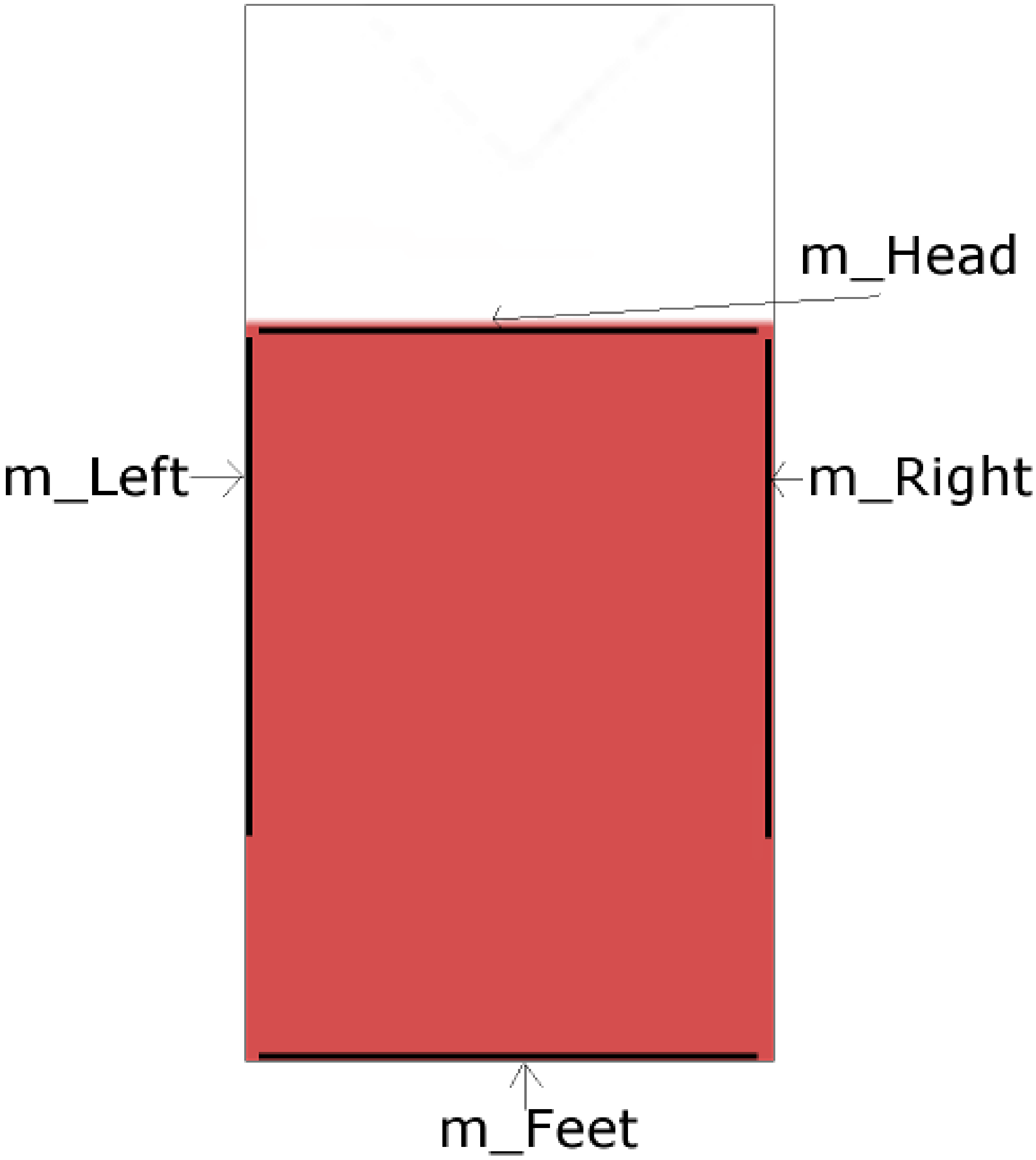
```
Engine::draw()
{
    // All the drawing code
}
```

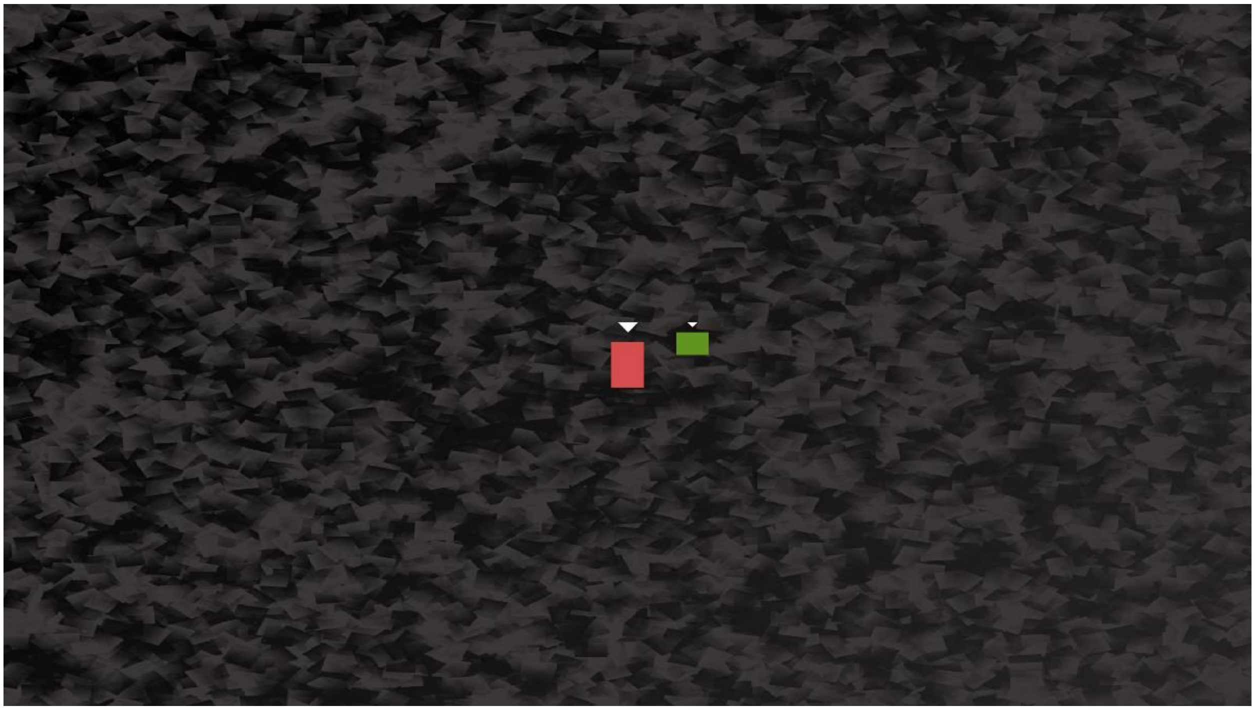


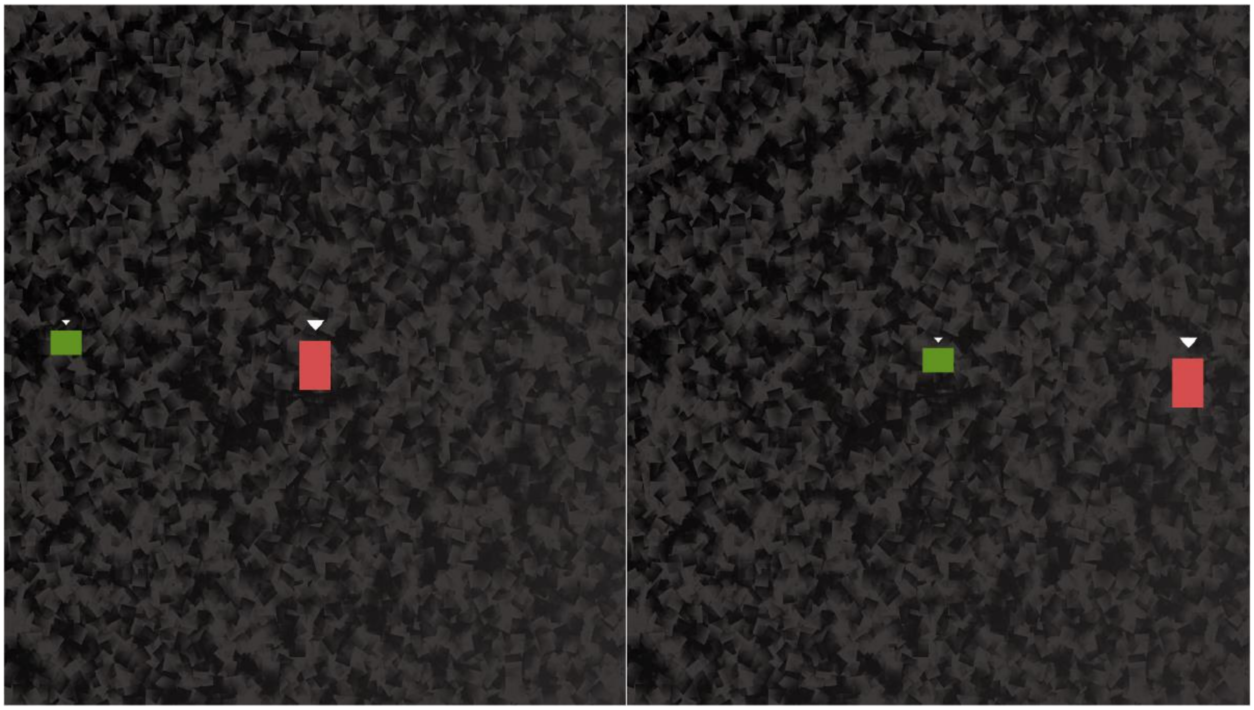


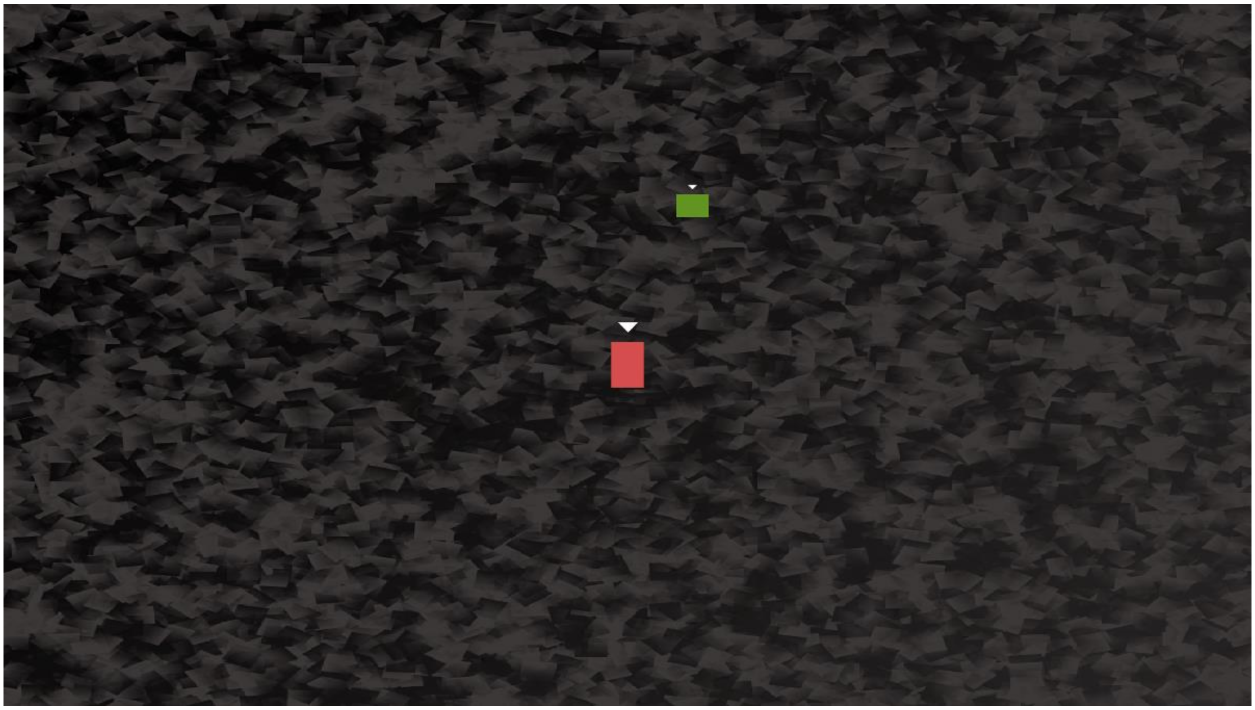
```cpp
#include "stdafx.h"
#include "Engine.h"

void Engine::draw()
```

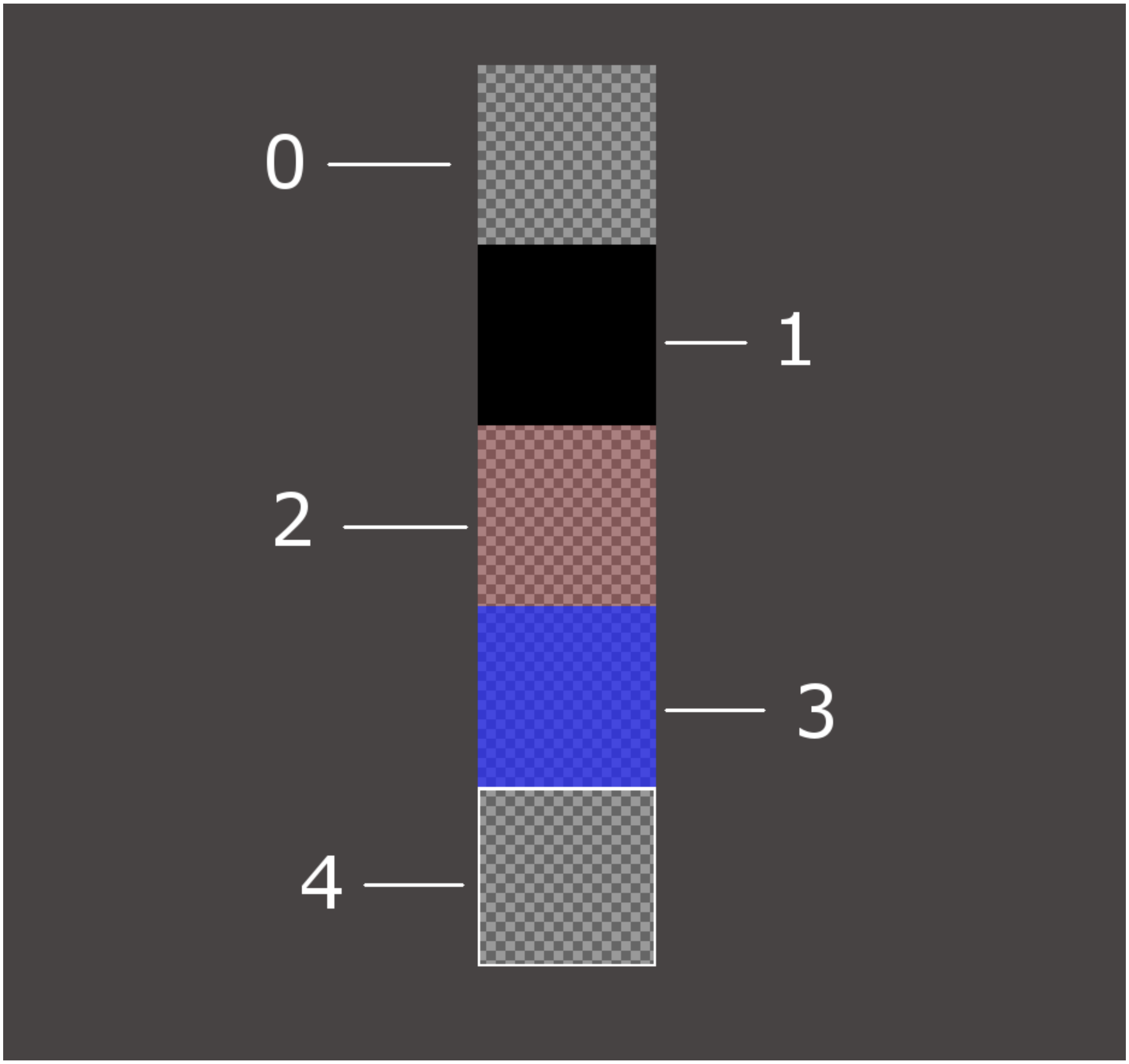


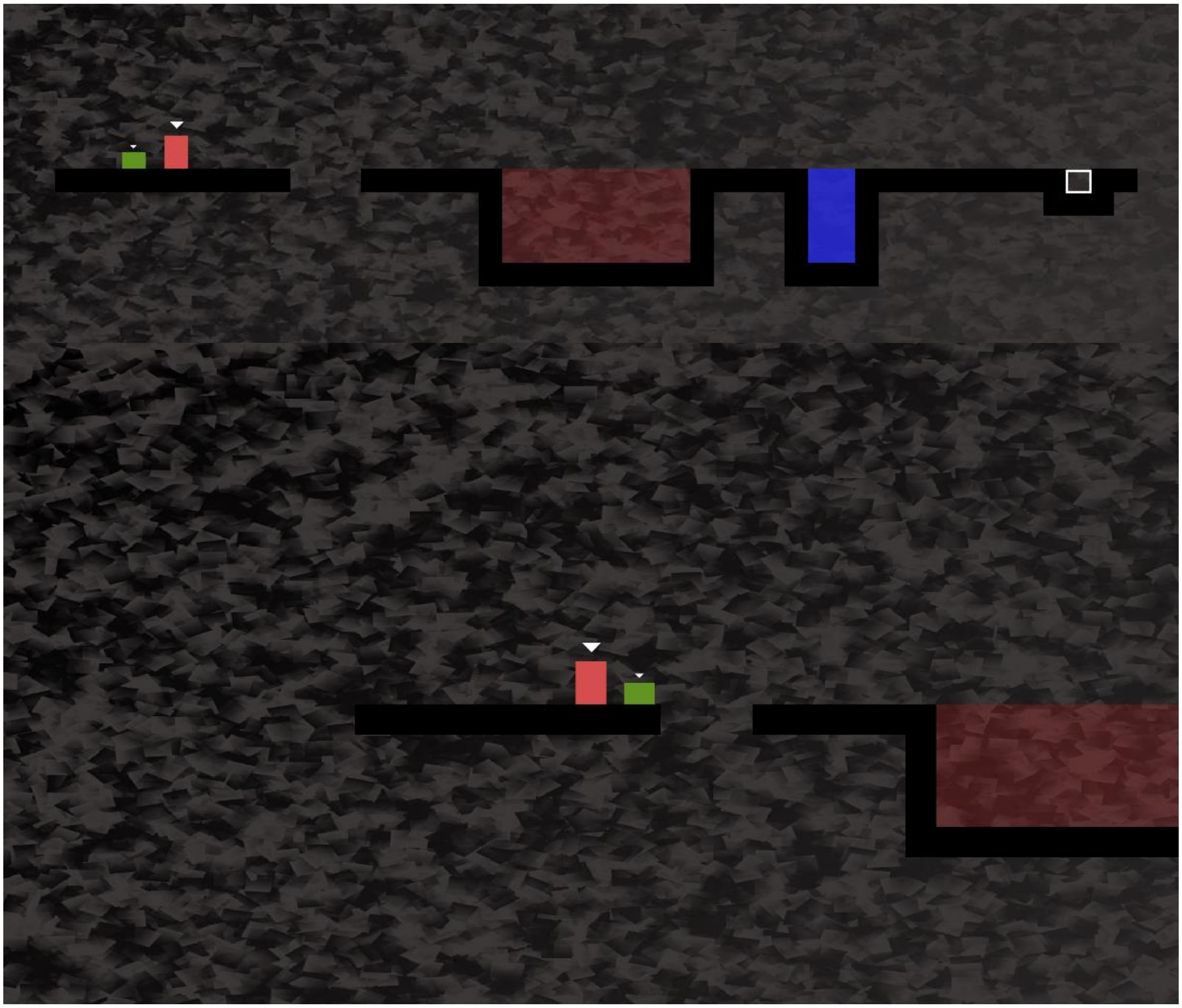

m_Head
m_Left
m_Right
m_Feet

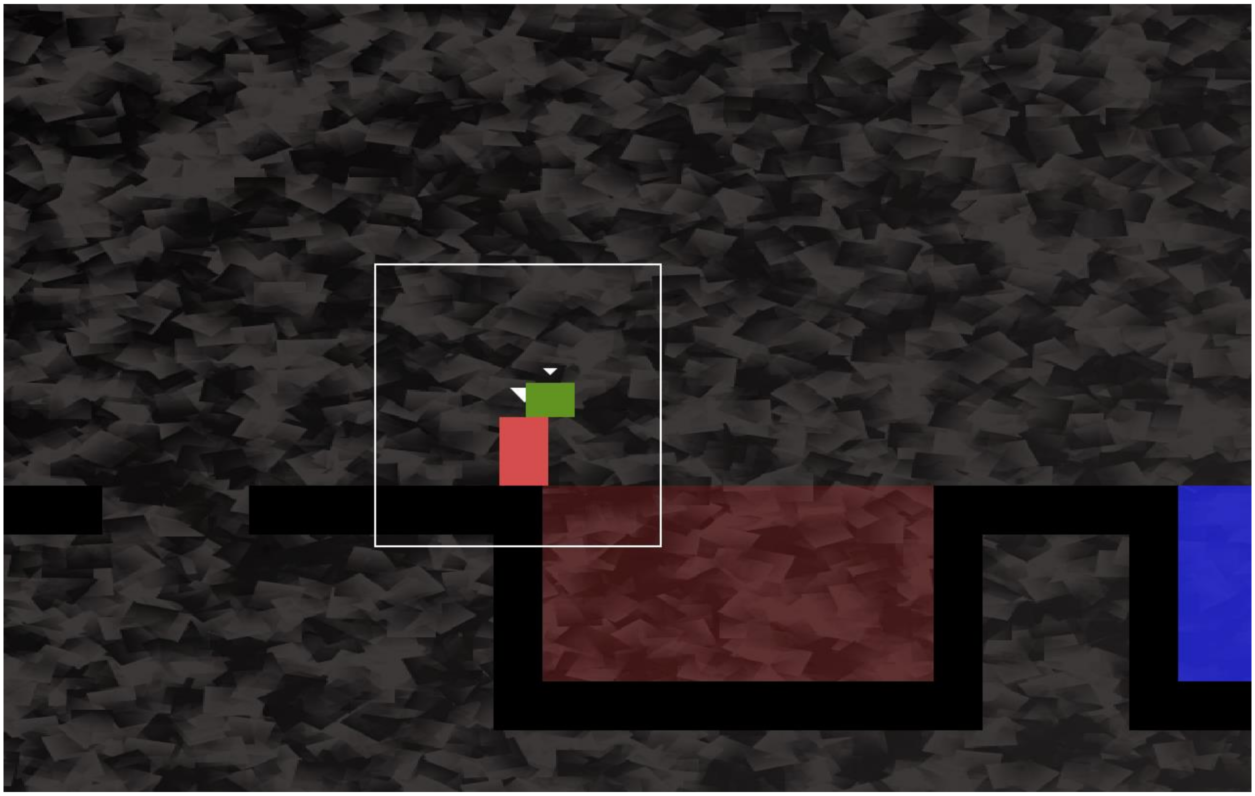


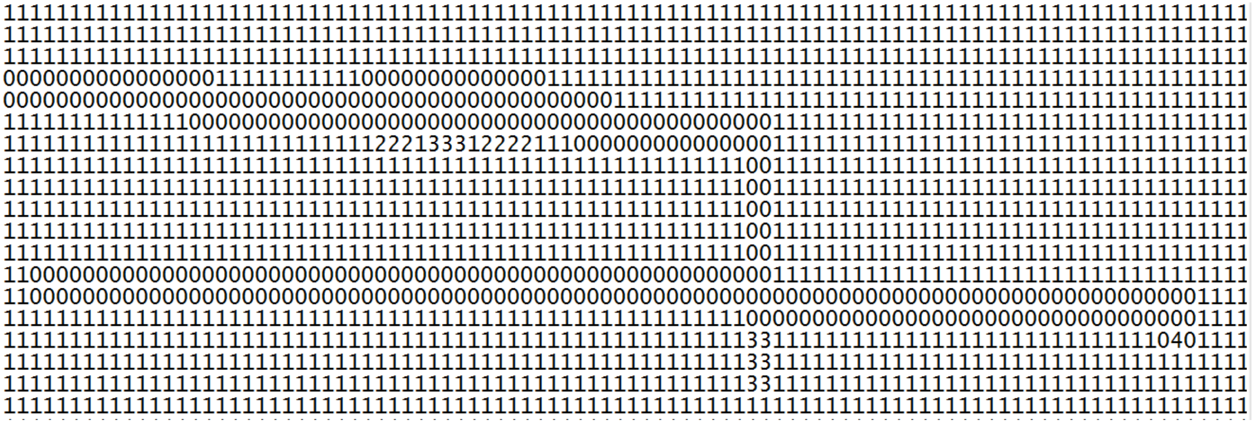




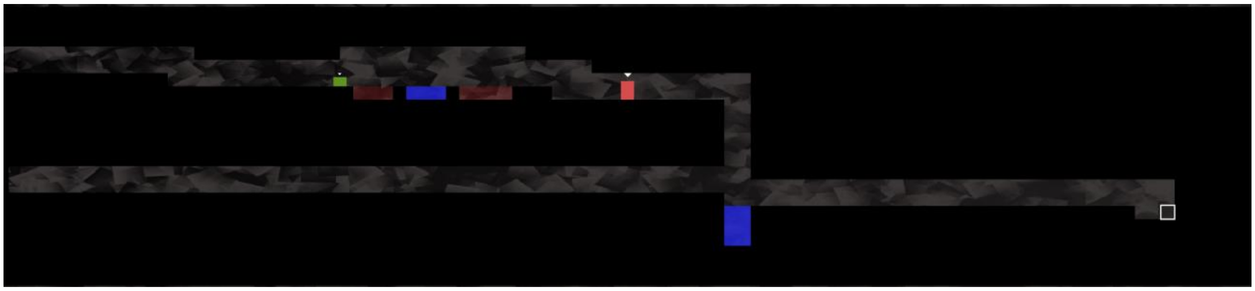

0
1
2
3
4

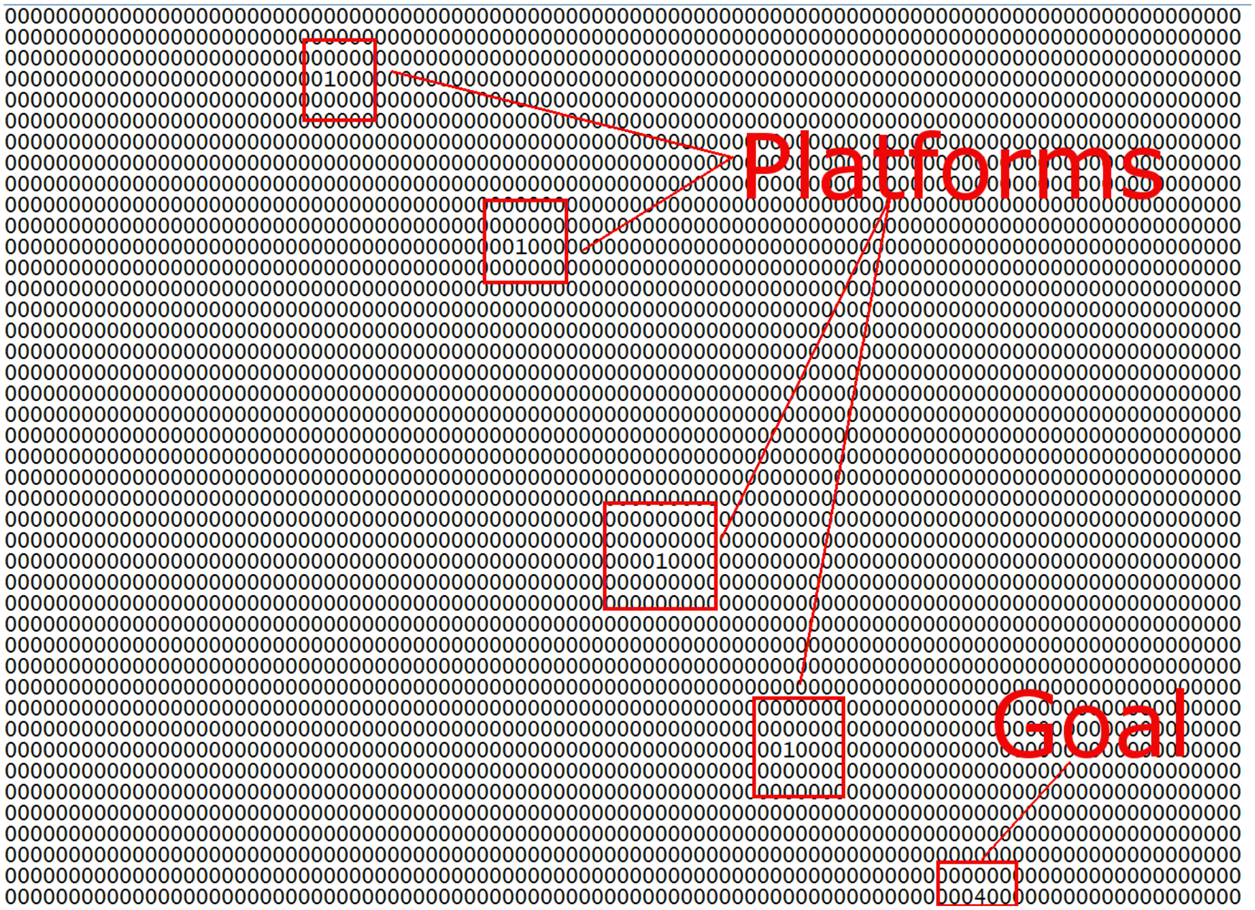




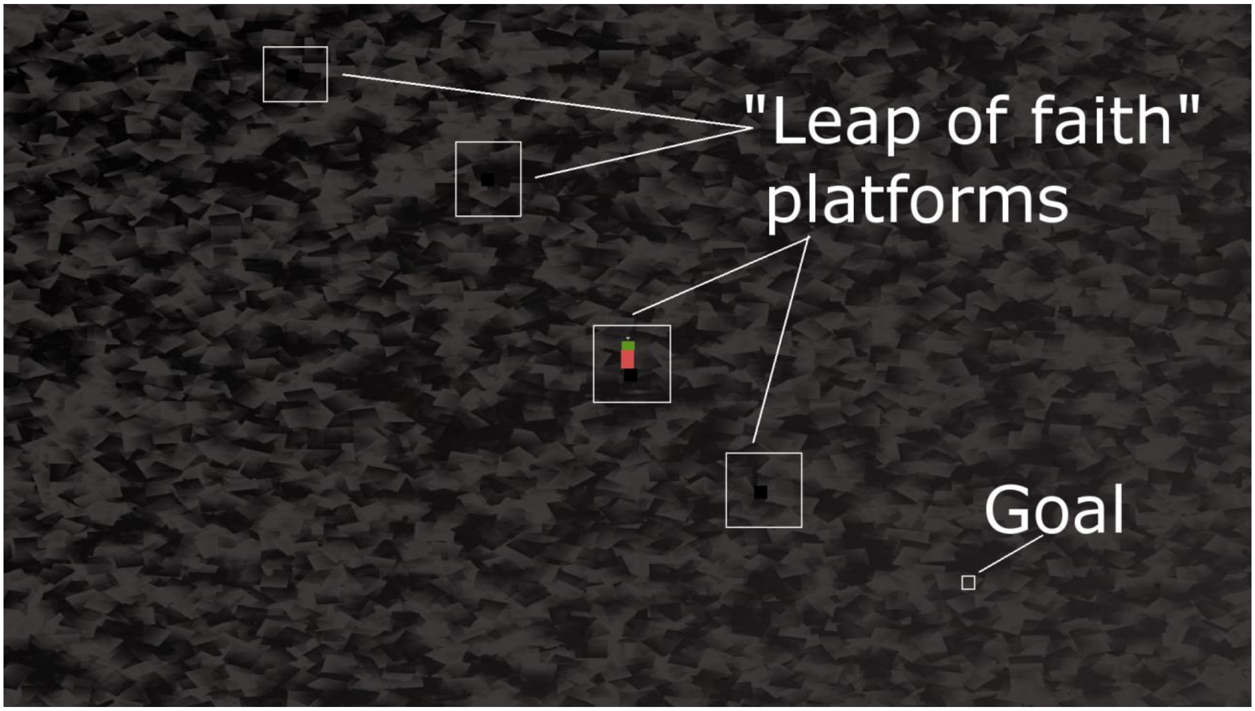


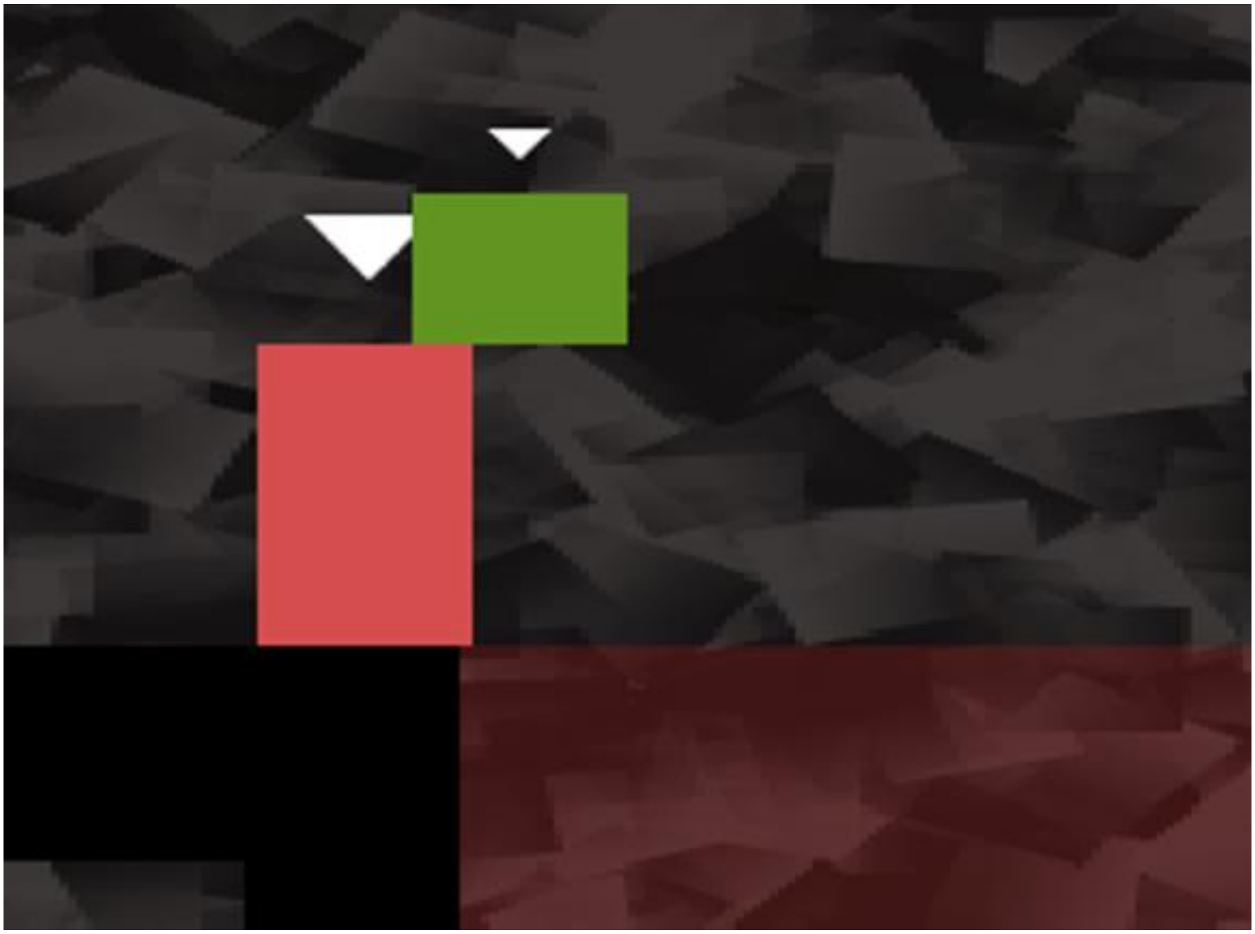
```
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
0000000000000000111111111110000000000000011111111111111111111111111111111111111111111111111111
0000000000000000000000000000000000000000000000111111111111111111111111111111111111111111111111
1111111111111100000000000000000000000000000000000000000000111111111111111111111111111111111111
1111111111111111111111111111222133312222111000000000000000111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111100111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111100111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111100111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111100111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111100111111111111111111111111111111111111
1100000000000000000000000000000000000000000000000000000000111111111111111111111111111111111111
1100000000000000000000000000000000000000000000000000000000000000000000000000000000000000001111
1111111111111111111111111111111111111111111111111111111100000000000000000000000000000000001111
1111111111111111111111111111111111111111111111111111111133111111111111111111111111111040401111
1111111111111111111111111111111111111111111111111111111133111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111133111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
```





```
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000100000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000100000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000001000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000100000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000400000000000000000000
```





Chapter 16: Extending SFML Classes, Particle Systems and Shaders