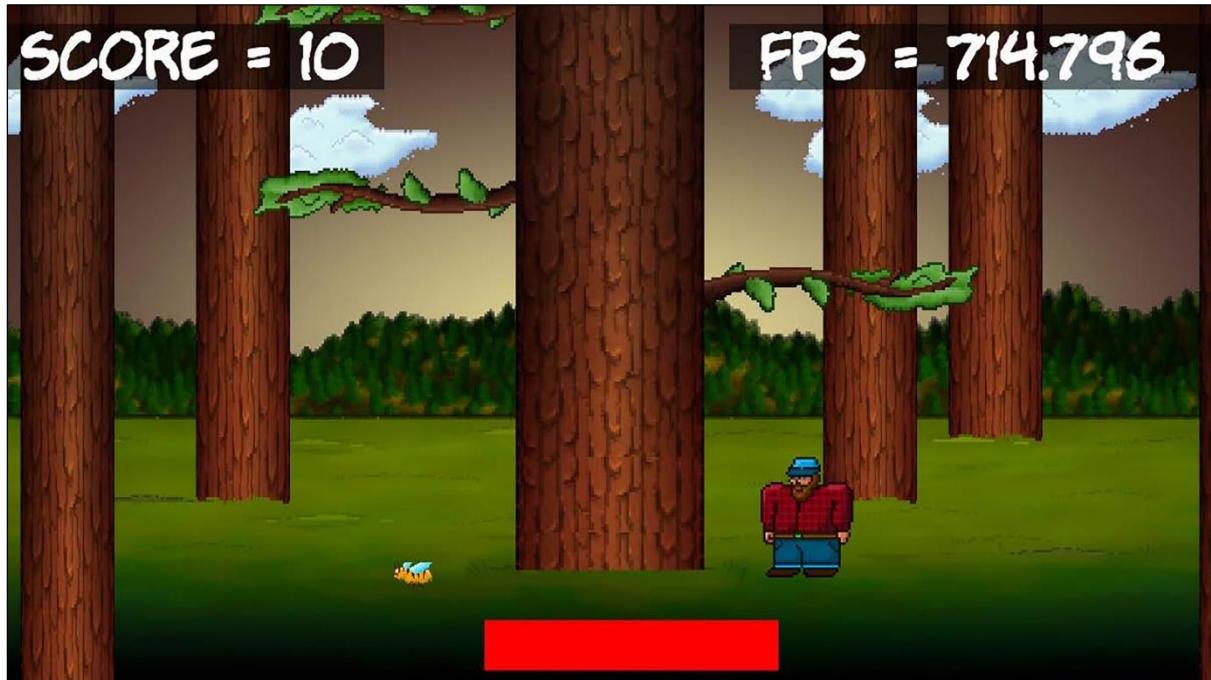
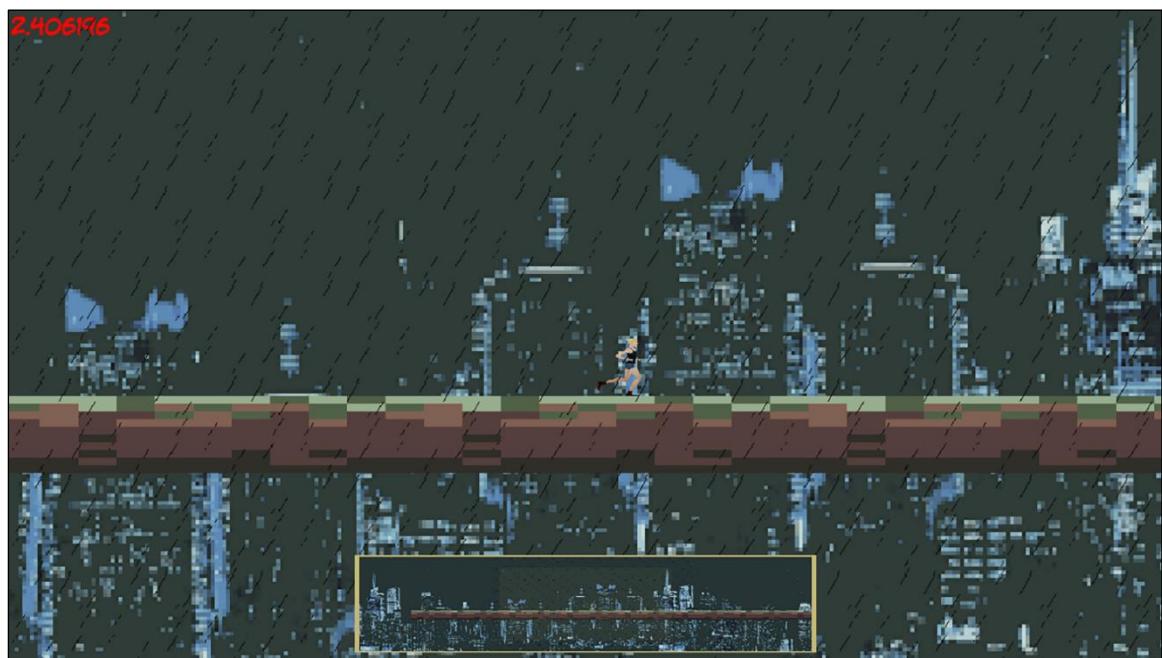
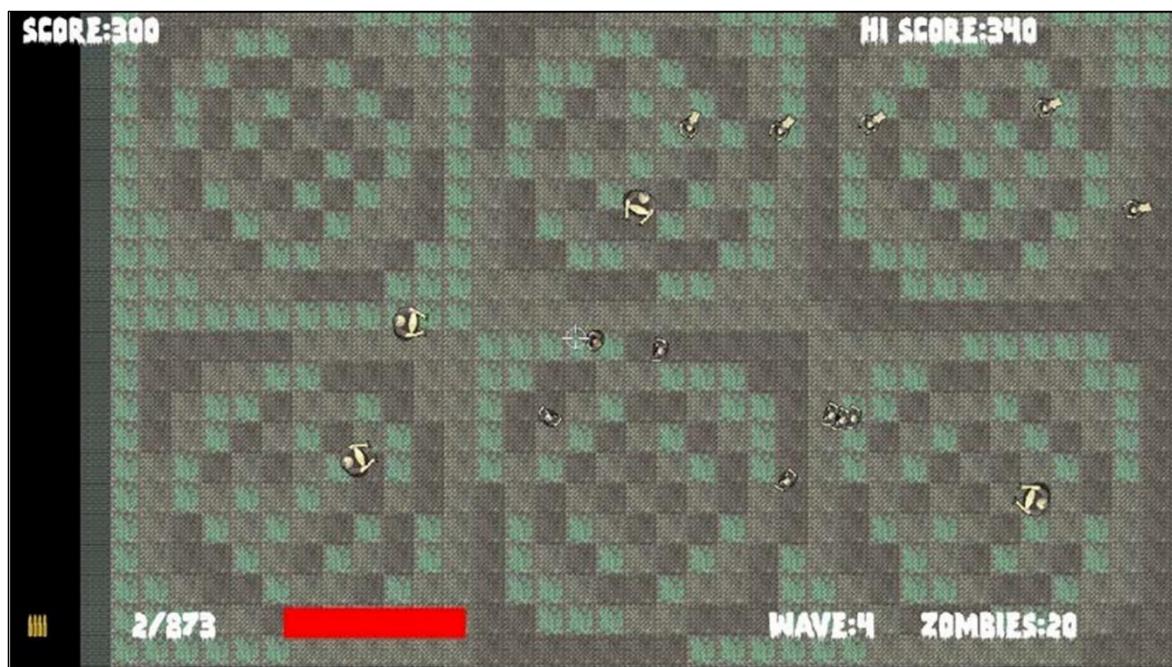


## Chapter 1: Welcome to Beginning C++ Game Programming Third Edition!





## Meet the Visual Studio family



Visual Studio | 

The most comprehensive IDE for .NET and C++ developers on Windows. Fully packed with a sweet array of tools and features to elevate and enhance every stage of software development.

[Learn more →](#)

[Download Visual Studio ▾](#)

Community 2022 Free for individual developers, academic uses, and open source

① V Professional 2022 We can help!

Enterprise 2022



Visual Studio Code |   

A standalone source code editor that runs on Windows, macOS, and Linux. The top pick for JavaScript and web developers, with extensions to support just about any programming language.

[Learn more →](#)

By using Visual Studio Code you agree to its [license & privacy statement](#)

[Download Visual Studio Code ▾](#)

## Download



SFML 2.6.0  
Latest stable version



Snapshots  
In development versions

## Download SFML 2.6.0

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.

**Unless you are using a newer version of Visual Studio, the compiler versions have to match 100%!**

Here are links to the specific MinGW compiler versions used to build the provided packages:

WinLibs MSVCRT 13.1.0 (32-bit), WinLibs MSVCRT 13.1.0 (64-bit)

Visual C++ 17 (2022) - 32-bit

[Download | 20.3 MB](#)

Visual C++ 17 (2022) - 64-bit

[Download | 21.9 MB](#)

Visual C++ 16 (2019) - 32-bit

[Download | 19.3 MB](#)

Visual C++ 16 (2019) - 64-bit

[Download | 20.8 MB](#)

Visual C++ 15 (2017) - 32-bit

[Download | 17.7 MB](#)

Visual C++ 15 (2017) - 64-bit

[Download | 19.4 MB](#)

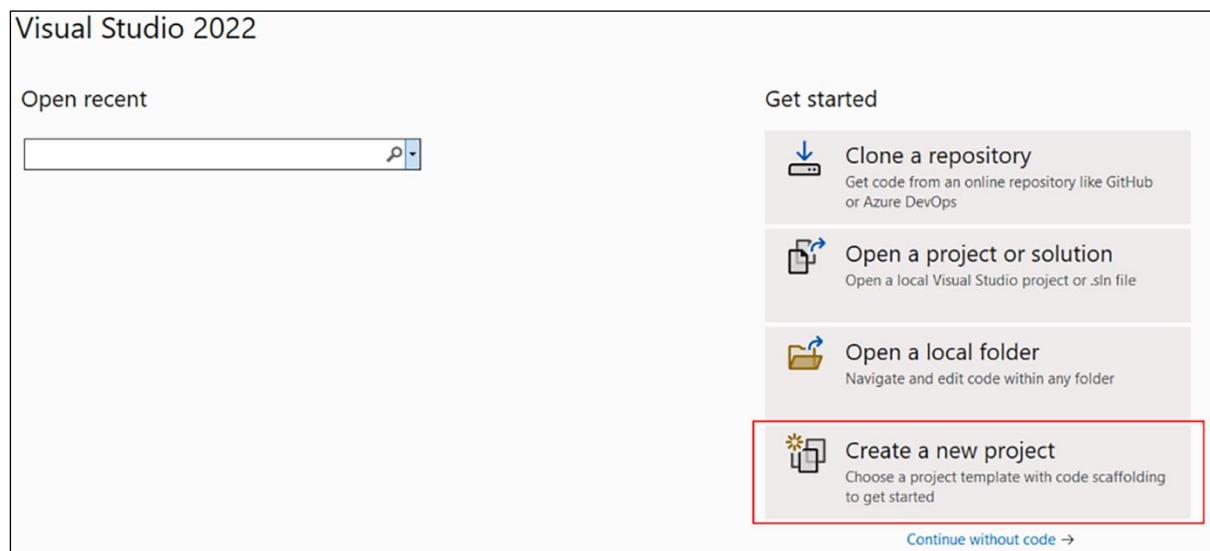
GCC 13.1.0 MinGW (DW2) - 32-bit

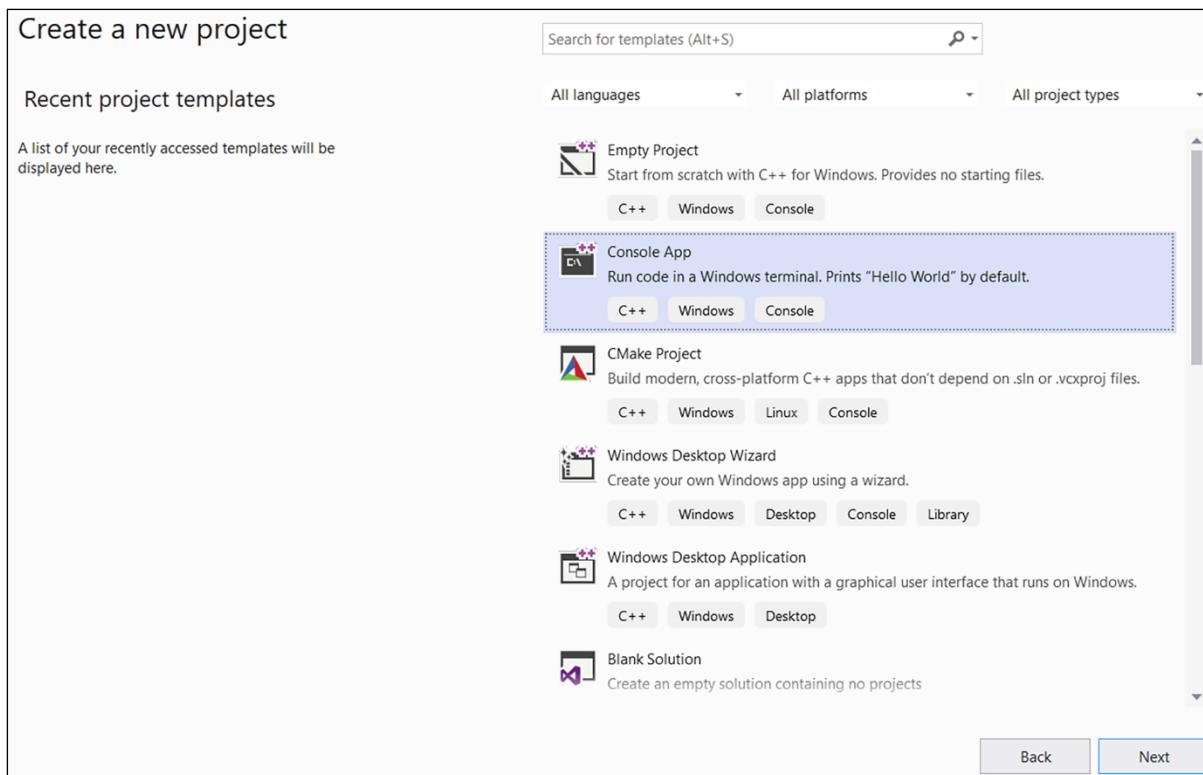
[Download | 17.9 MB](#)

GCC 13.1.0 MinGW (SEH) - 64-bit

[Download | 19.0 MB](#)

Name	Date modified	Type
bin	01/11/2023 12:36	File folder
doc	01/11/2023 12:36	File folder
examples	01/11/2023 12:37	File folder
include	01/11/2023 12:37	File folder
lib	01/11/2023 12:37	File folder
license.md	01/11/2023 12:36	MD File
readme.md	01/11/2023 12:36	MD File





### Configure your new project

Console App

Project name

Location

Solution name

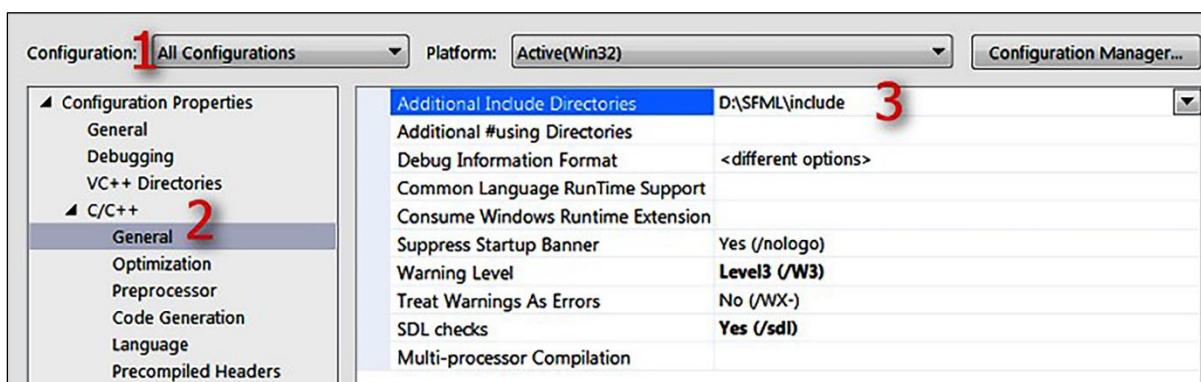
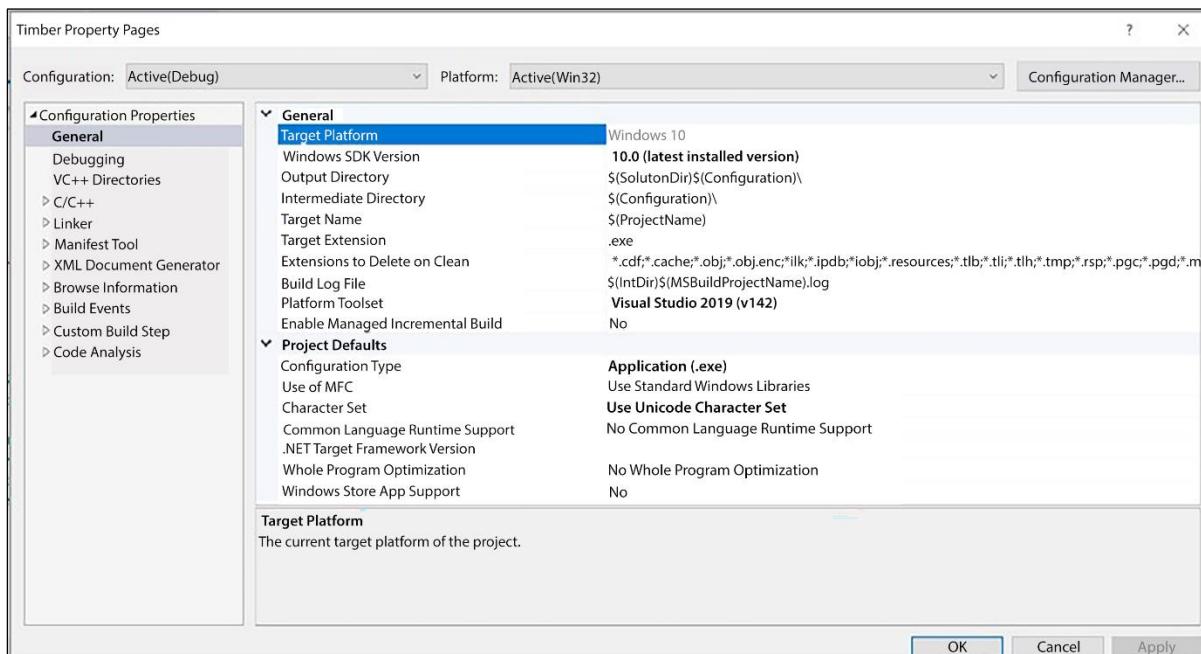
Place solution and project in the same directory

Back Create

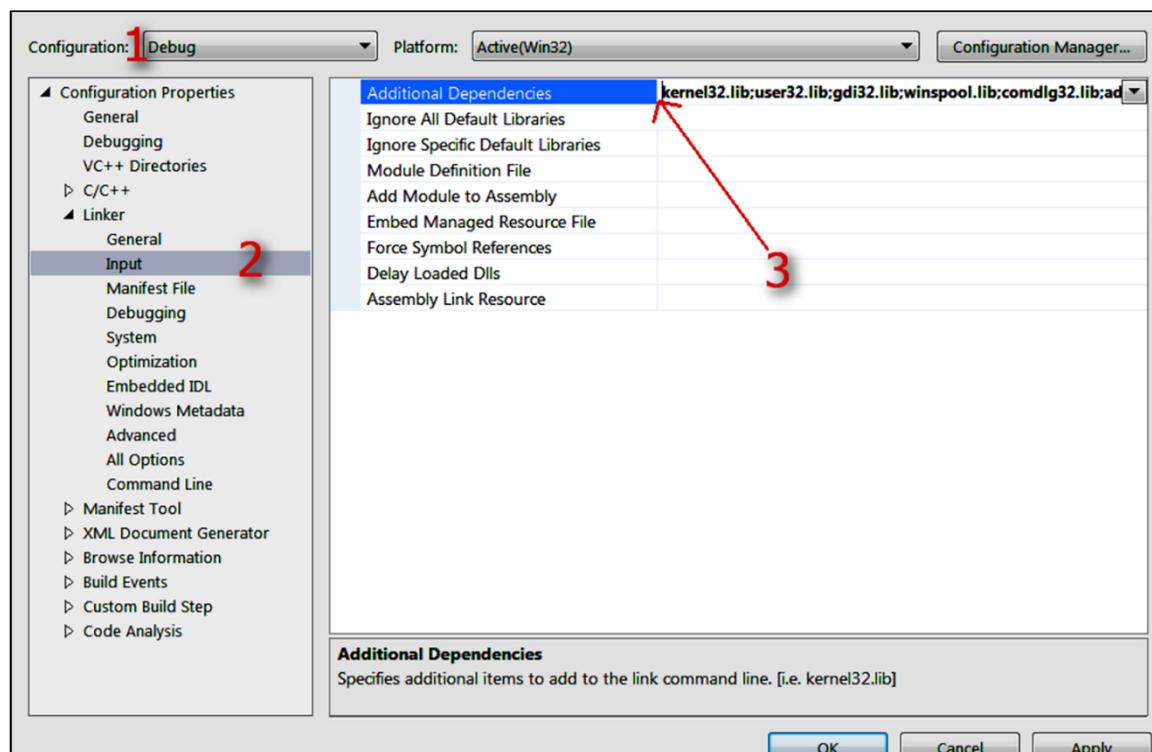
```

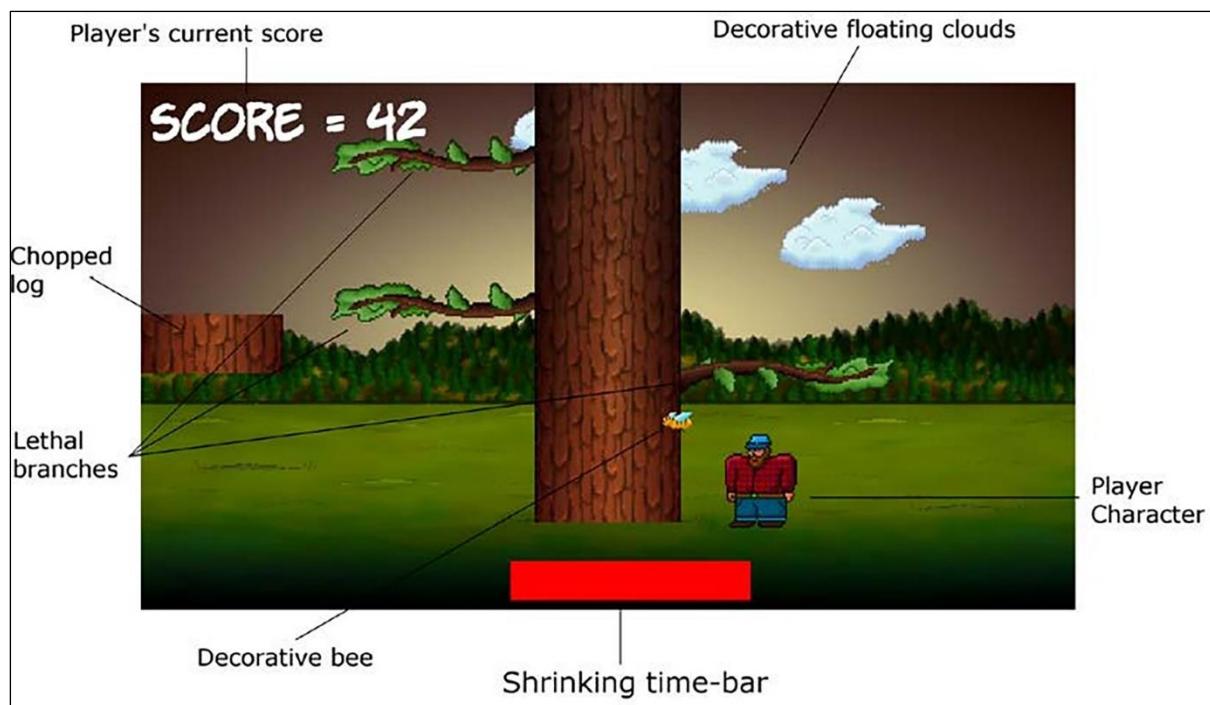
1 // Timber.cpp : This file contains the 'main' function. Program execution begins and ends there.
2 //
3 // include <iostream>
4
5 int main()
6 {
7     std::cout << "Hello World!\n";
8 }
9
10 // Run program: Ctrl + F5 or Debug > Start Without Debugging menu
11 // Debug program: F5 or Debug > Start Debugging menu
12
13 // Tips for Getting Started:
14 // 1. Use the Solution Explorer window to add/manage files
15 // 2. Use the Team Explorer window to connect to source control
16 // 3. Use the Output window to see build output and other messages
17 // 4. Use the Error List window to view errors
18 // 5. Go to Project > Add New Item to create new code files, or Project > Add Existing Item to add existing code files to the project
19 // 6. In the future, to open this project again, go to File > Open > Project and select the .sln file
20

```



▲ Configuration Properties		
General	Output File	\$ (OutDir) \$ (TargetName) \$ (TargetExt)
Debugging	Show Progress	Not Set
VC++ Directories	Version	
▷ C/C++	Enable Incremental Linking	<different options>
▲ Linker	Suppress Startup Banner	Yes (/NOLOGO)
1 General	Ignore Import Library	No
Input	Register Output	No
Manifest File	Per-user Redirection	No
Debugging	Additional Library Directories	2 D:\SFML\lib
System	Link Library Dependencies	Yes
	Use Library Dependency Inputs	No

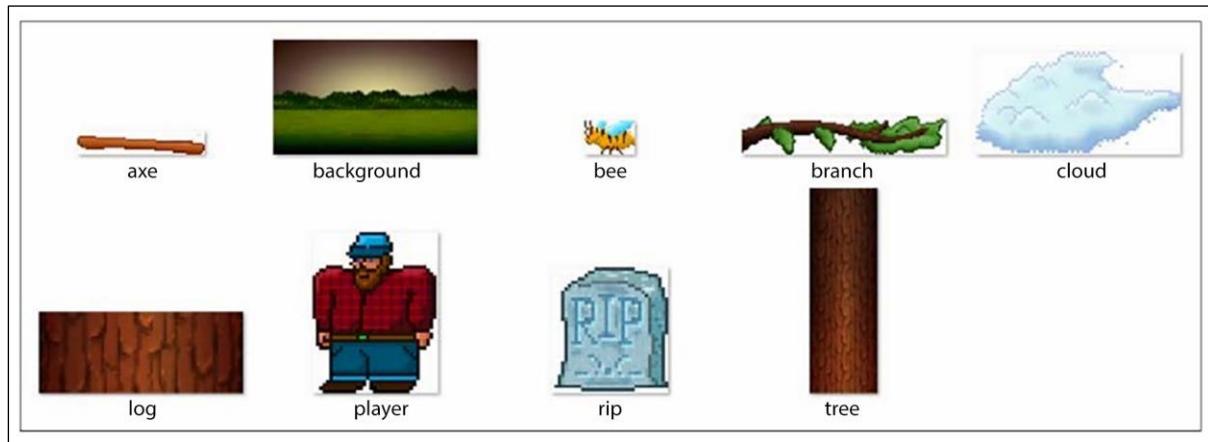


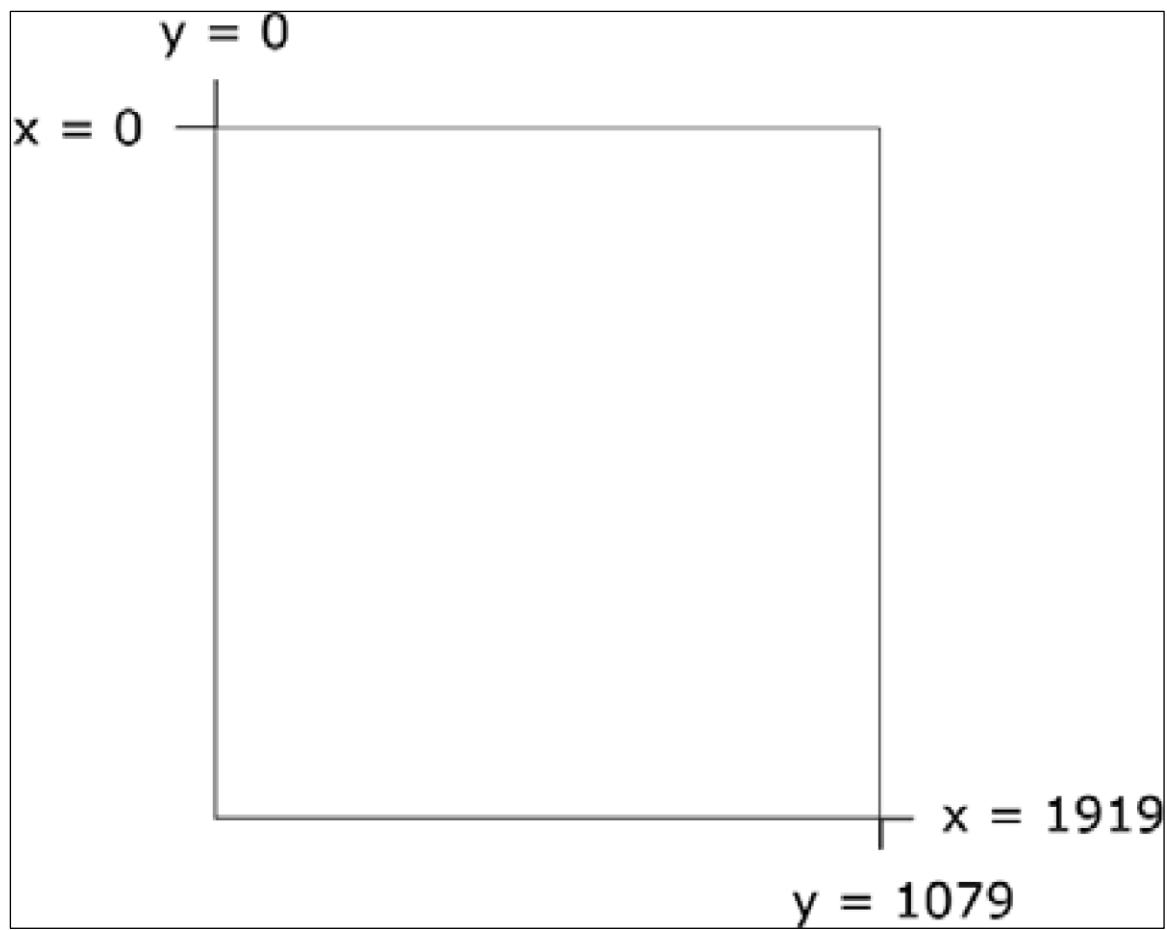


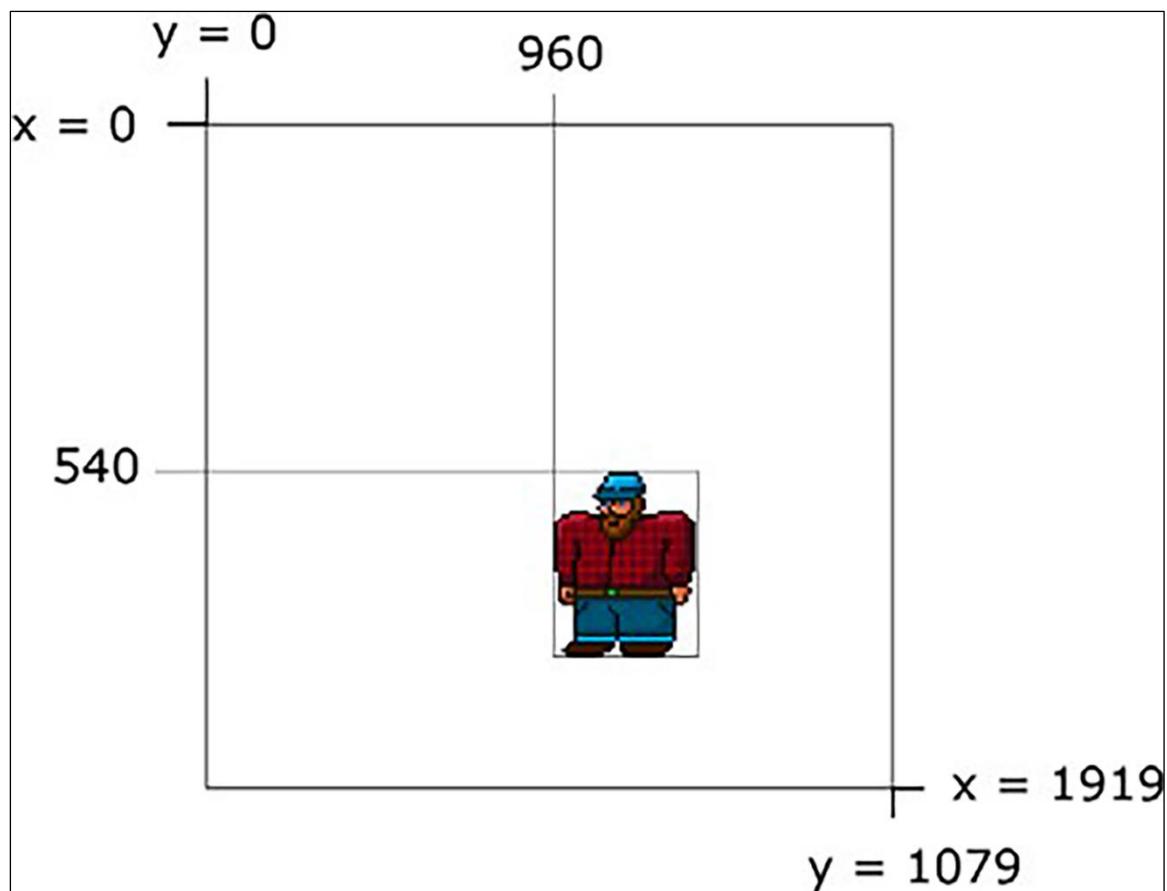
	<a href="#">openal32.dll</a>
	<a href="#">sfml-audio-2.dll</a>
	<a href="#">sfml-audio-d-2.dll</a>
	<a href="#">sfml-graphics-2.dll</a>
	<a href="#">sfml-graphics-d-2.dll</a>
	<a href="#">sfml-network-2.dll</a>
	<a href="#">sfml-network-d-2.dll</a>
	<a href="#">sfml-system-2.dll</a>
	<a href="#">sfml-system-d-2.dll</a>
	<a href="#">sfml-window-2.dll</a>
	<a href="#">sfml-window-d-2.dll</a>

**SOLUTION EXPLORER**

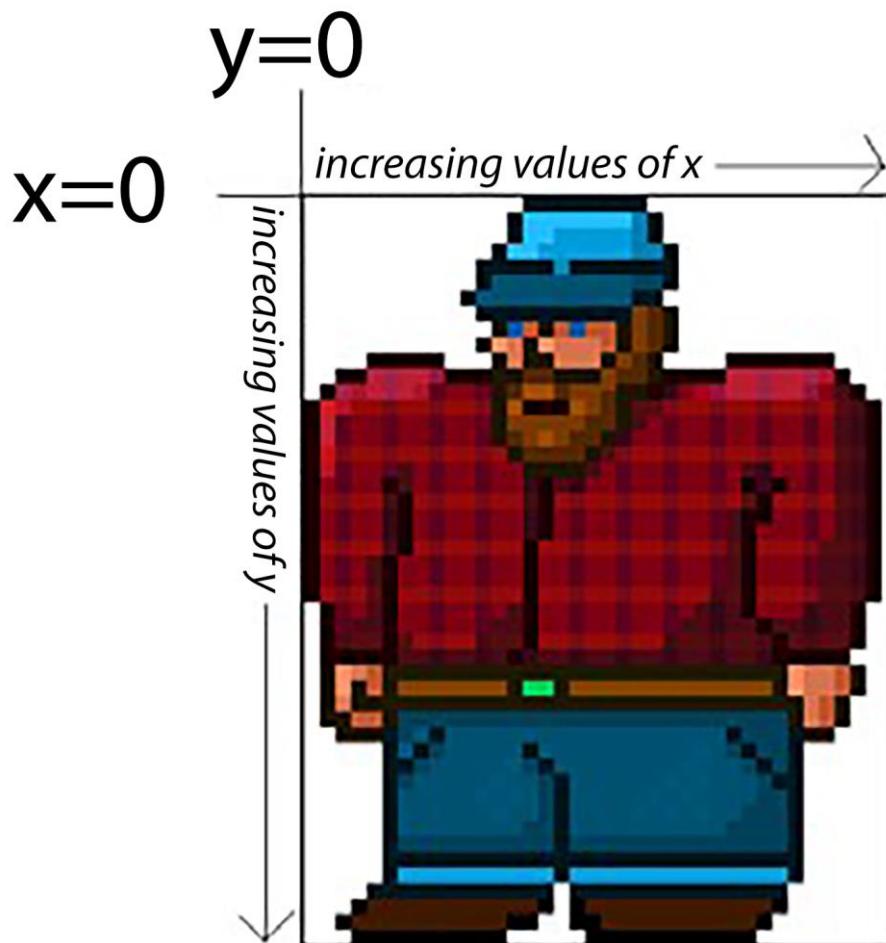
```
1 // Timber.cpp : This file contains the 'main' function. Program execution begins and ends there.
2 //
3
4 #include <iostream>
5
6 int main()
7 {
8     std::cout << "Hello World!" << std::endl;
9 }
10
11 // Run program: Ctrl + F5 or Debug > Start Without Debugging menu
12 // Debug program: F5 or Debug > Start Debugging menu
13
14 // Tips for Getting Started:
15 // 1. Use the Solution Explorer window to add/manage files
16 // 2. Use the Team Explorer window to connect to source control
17 // 3. Use the Output window to see build output and other messages
18 // 4. Use the Error List window to view errors
19 // 5. Go to Project > Add New Item to create new code files, or Project > Add Existing Item to add existing code files to the project
20 // 6. In the future, to open this project again, go to File > Open > Project and select the .sln file
21
```







# *Internal Coordinates (Origin = 0, 0)*



Local Windows Debugger ▾

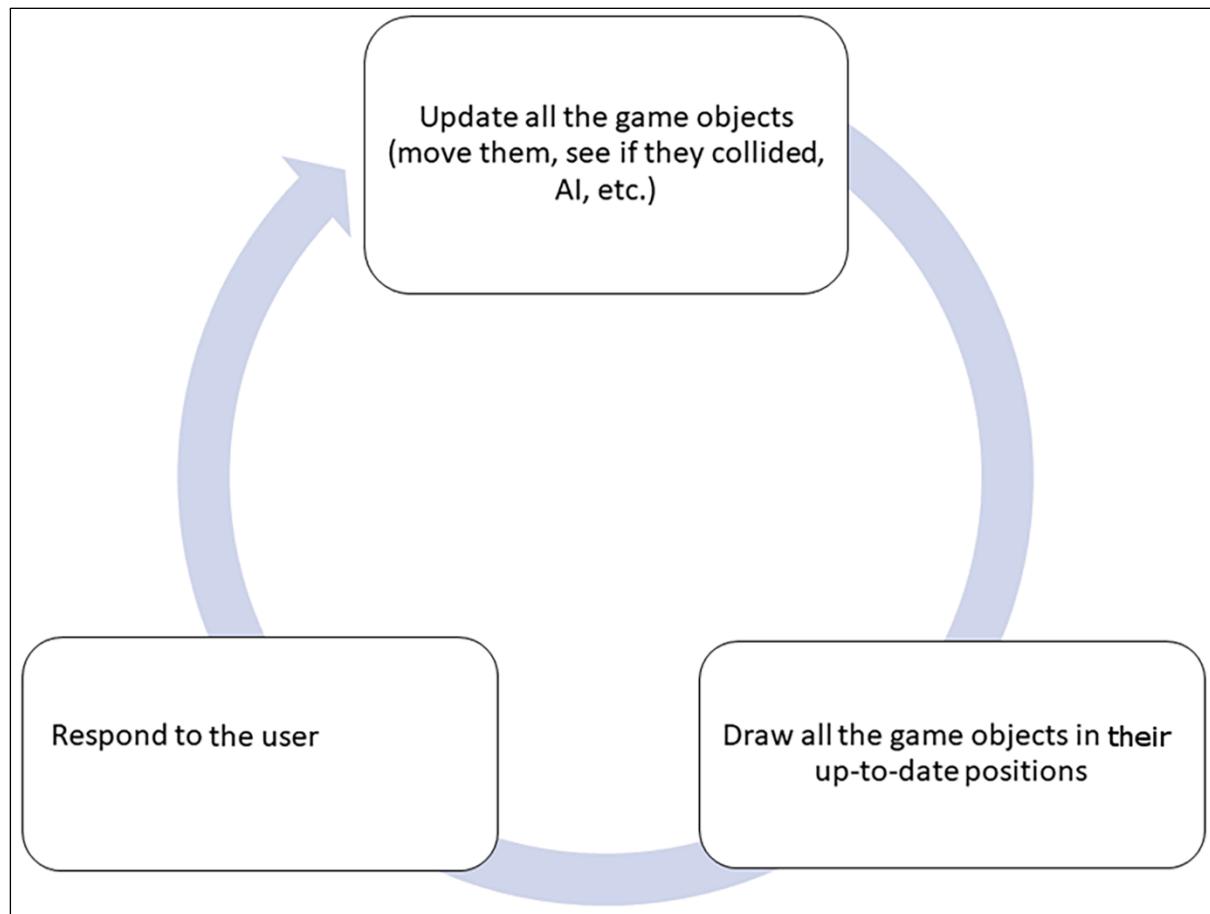
Debug ▾

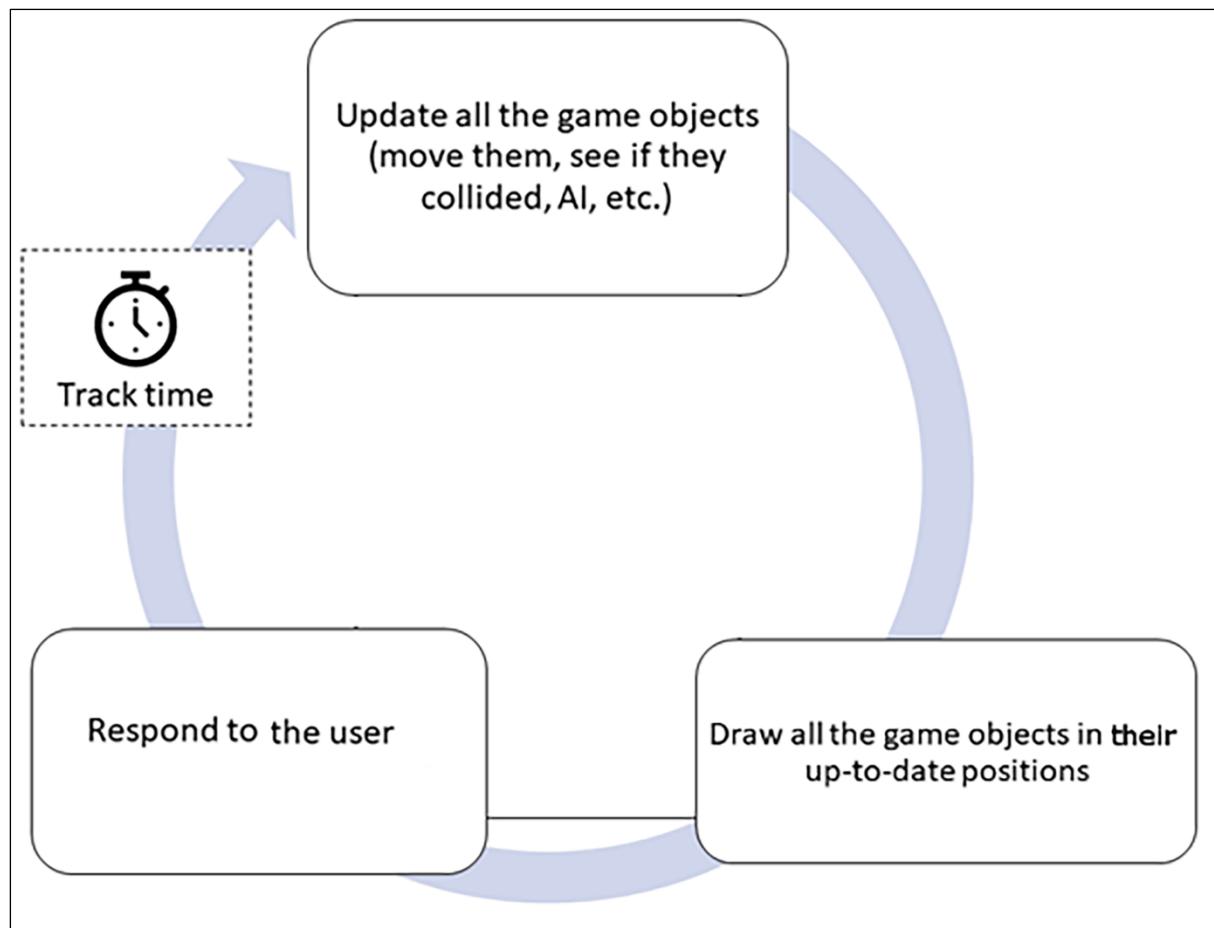
x86

▶ Local Windows Debugger ▾

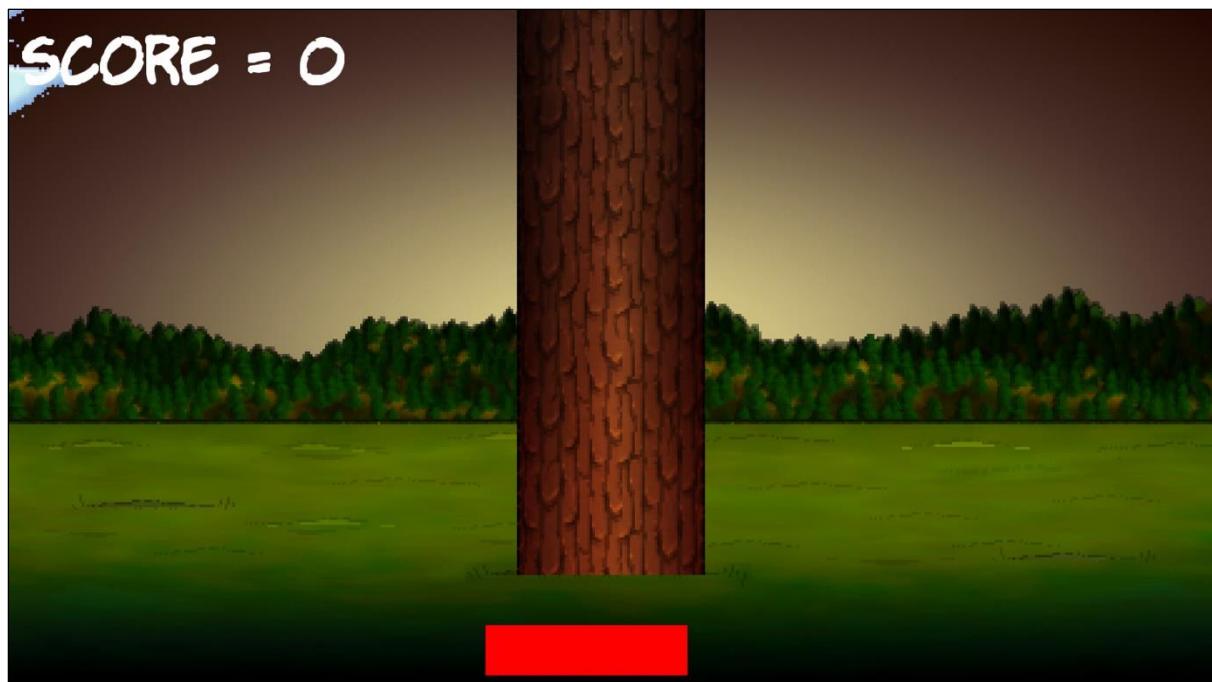
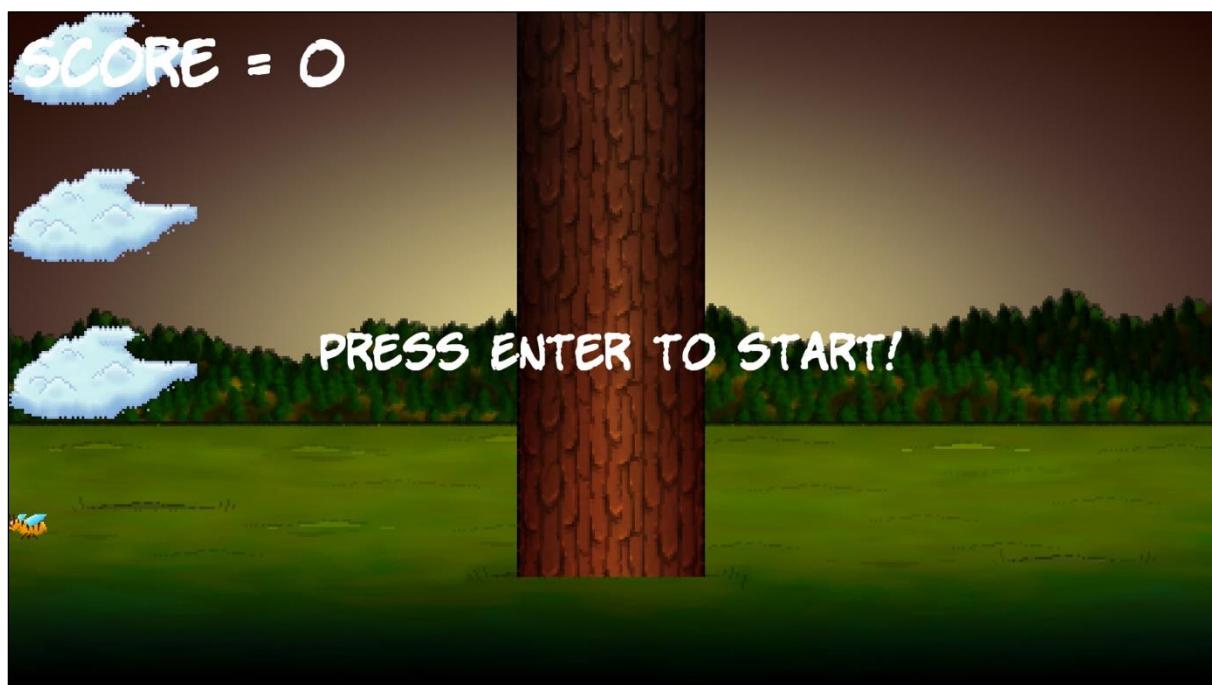


## Chapter 2: Variables, Operators, and Decisions: Animating Sprites





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



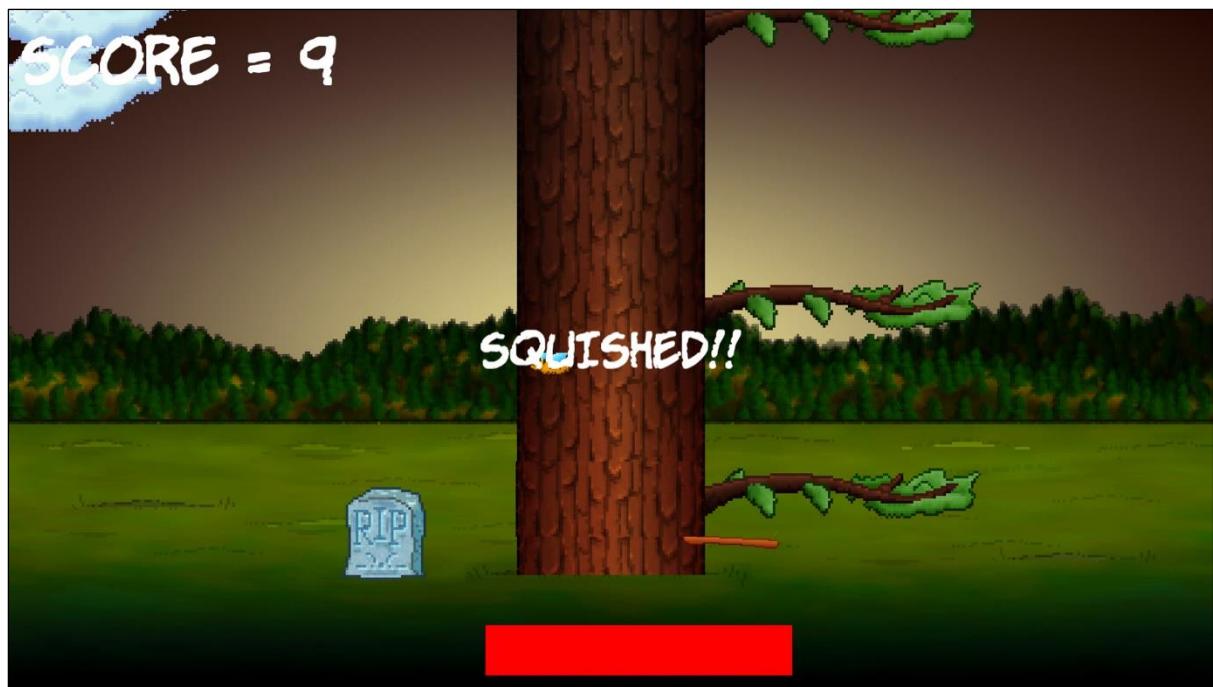


## Chapter 4: Loops, Arrays, Switch Conditions, Enumerations, and Functions: Implementing Game Mechanics



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





## Chapter 6: Object-Oriented Programming - Starting the Pong Game

# Bat

- m\_Length: int
- m\_Height: int
- m\_XPosition: int
- m\_YPosition: int

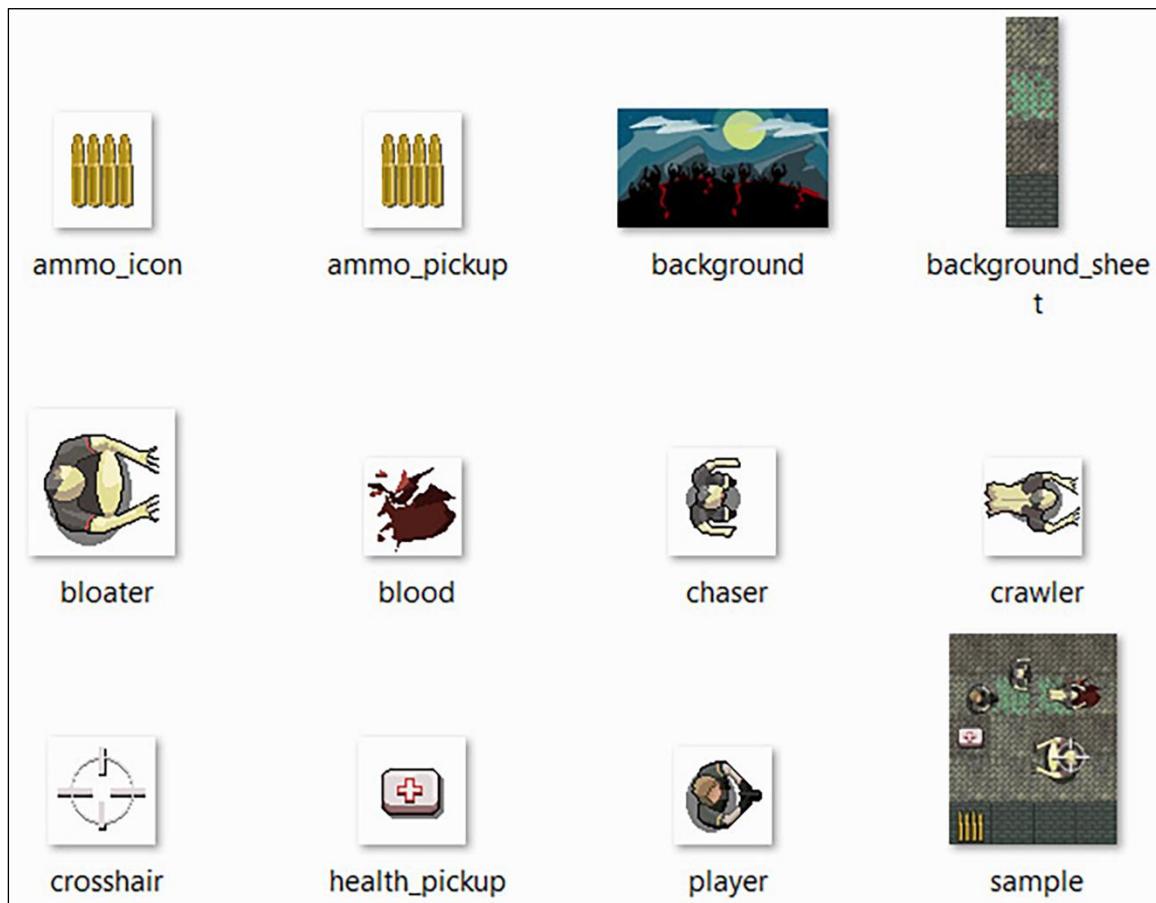
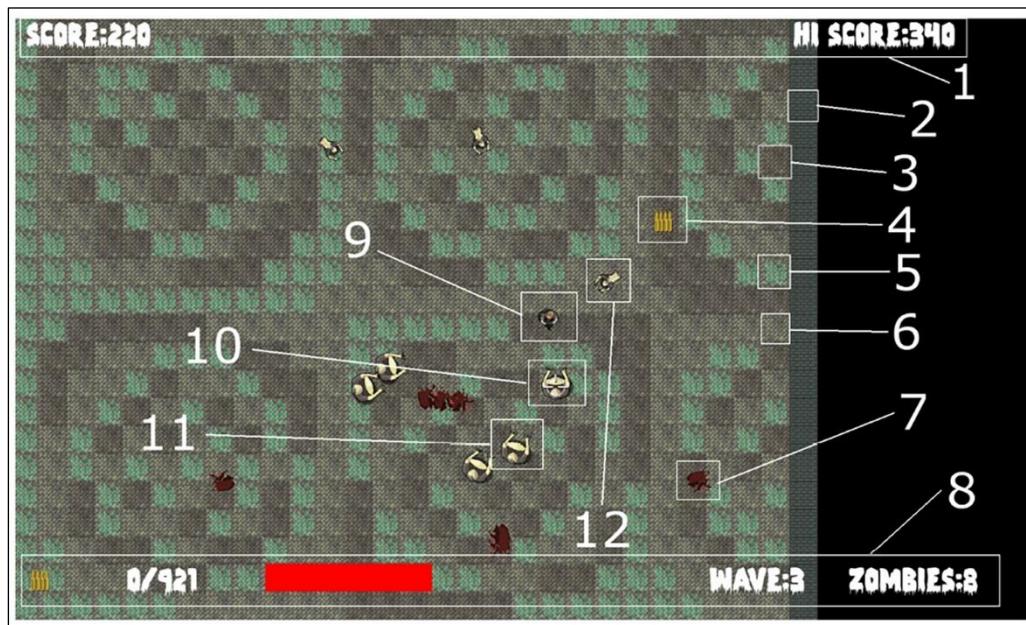
- + moveRight(): void
- + moveLeft(): void

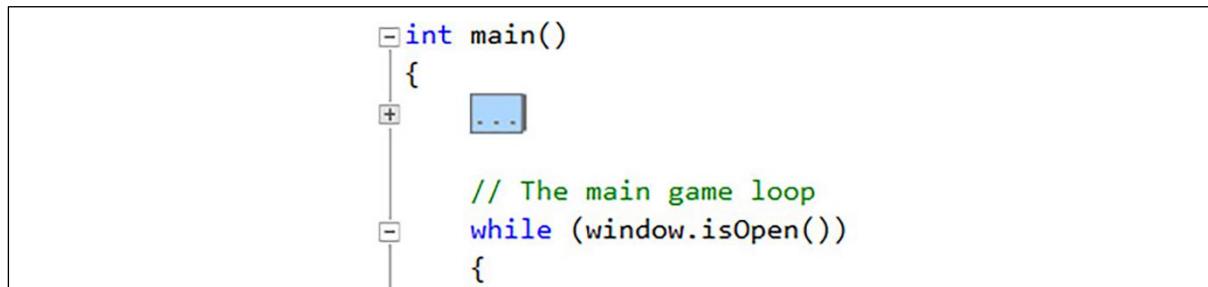
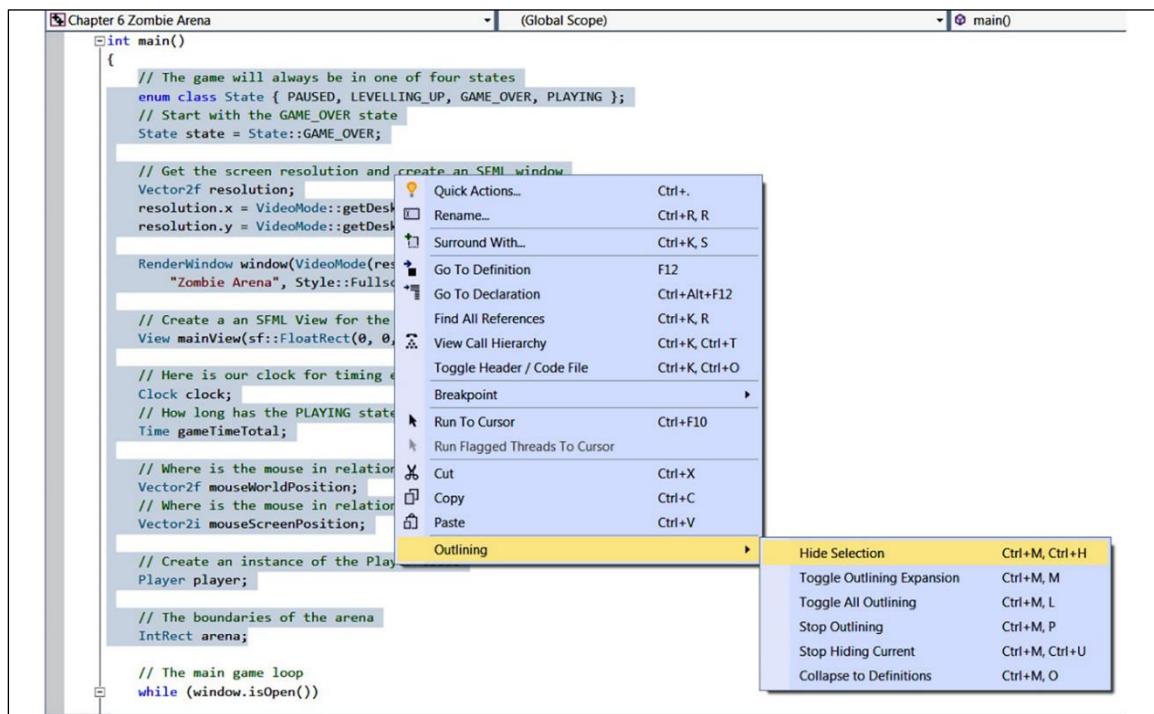
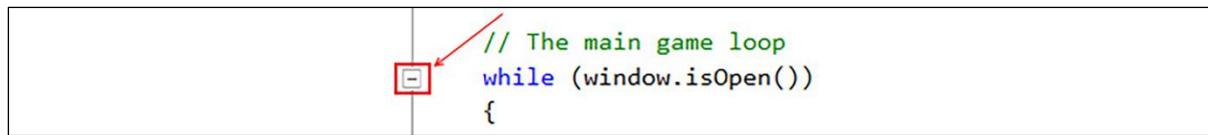
SCORE:0 LIVES:3

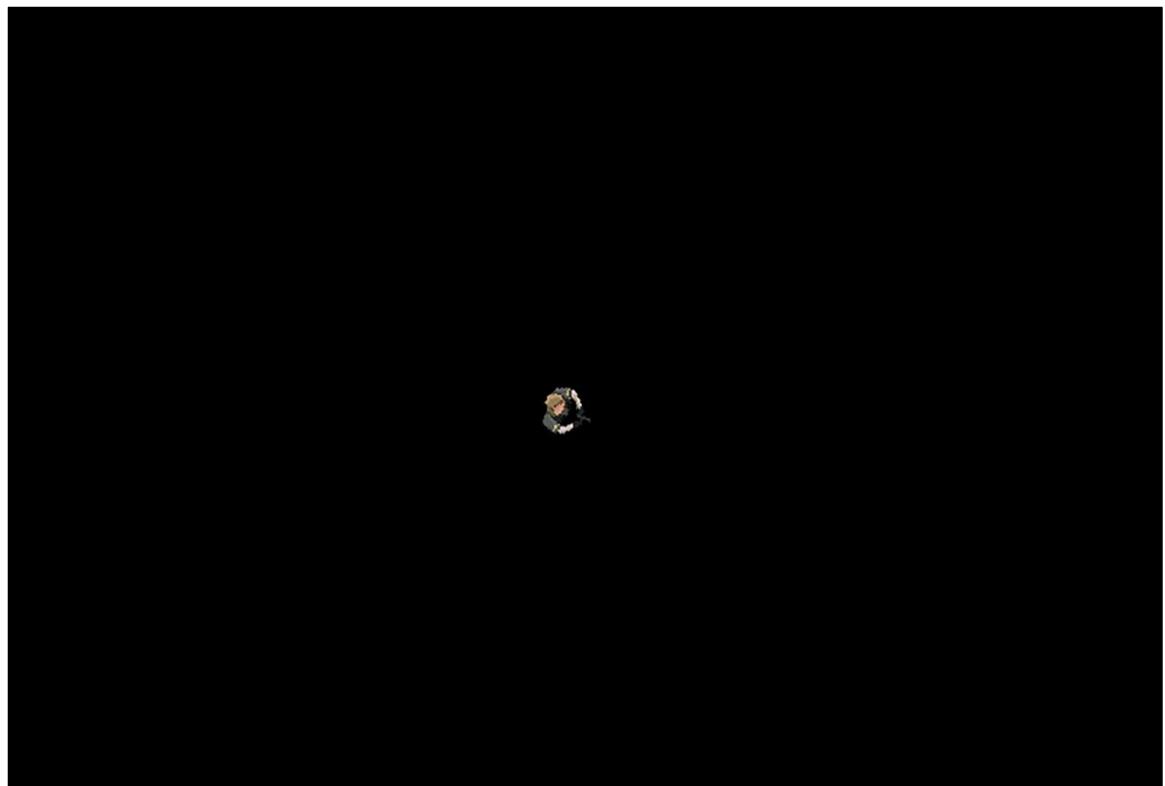
## Chapter 7: AABB Collision Detection and Physics - Finishing the Pong Game

SCORE:0 LIVES:3

## Chapter 8: SFML Views - Starting the Zombie Shooter Game

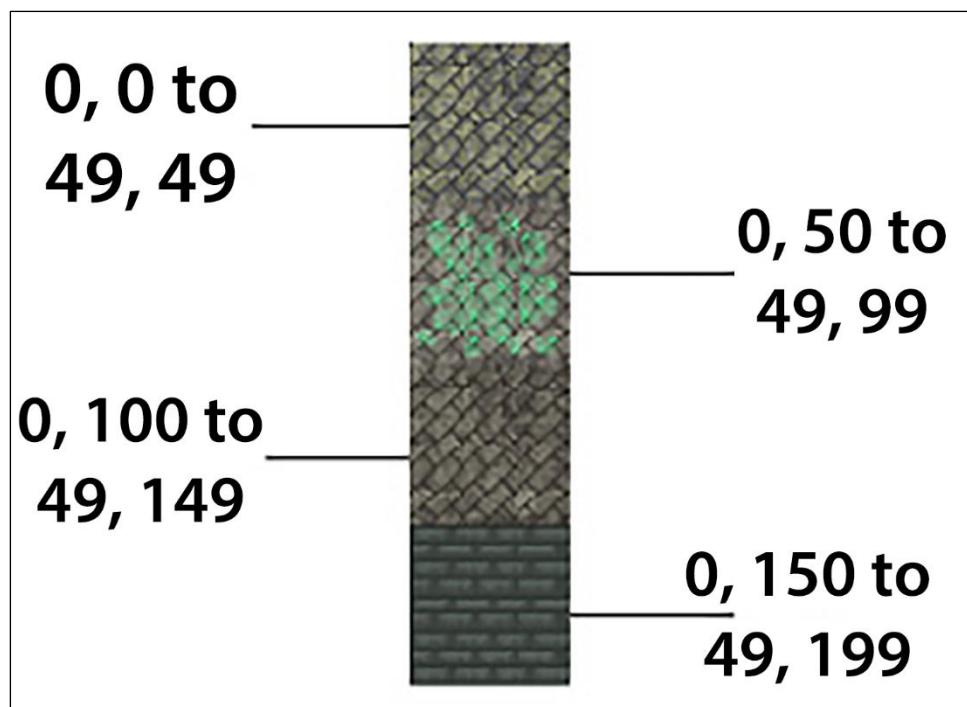


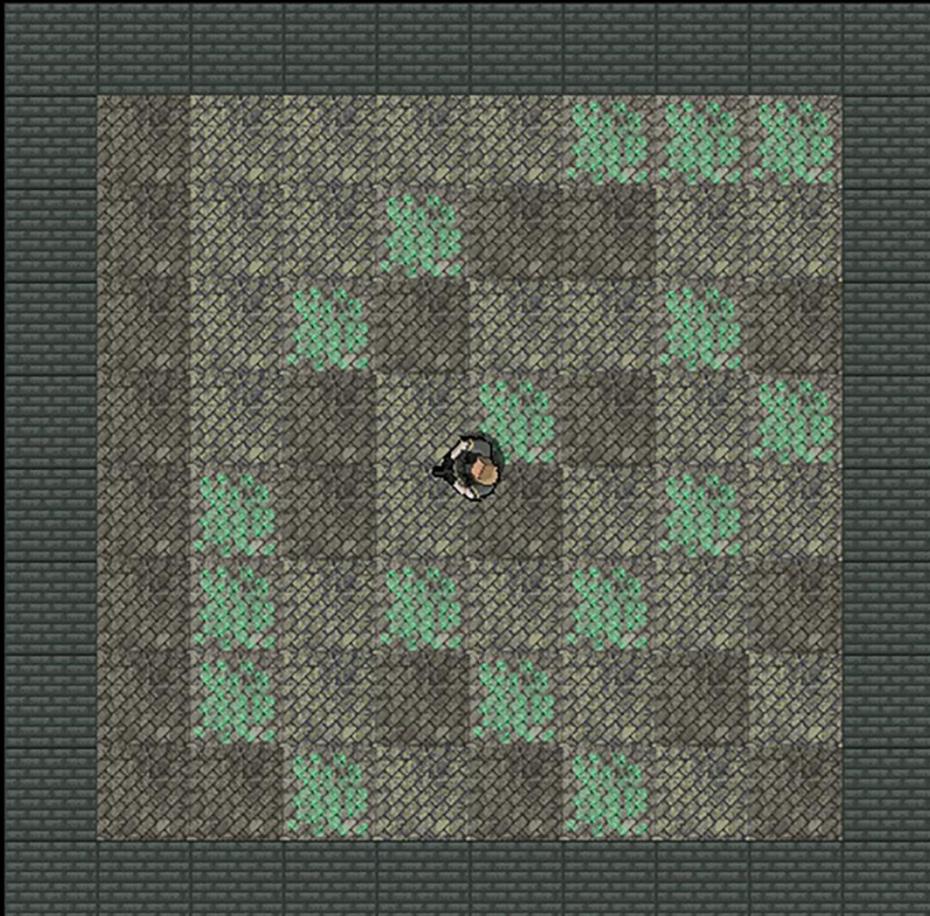




## Chapter 9: C++ References, Sprite Sheets, and Vertex Arrays

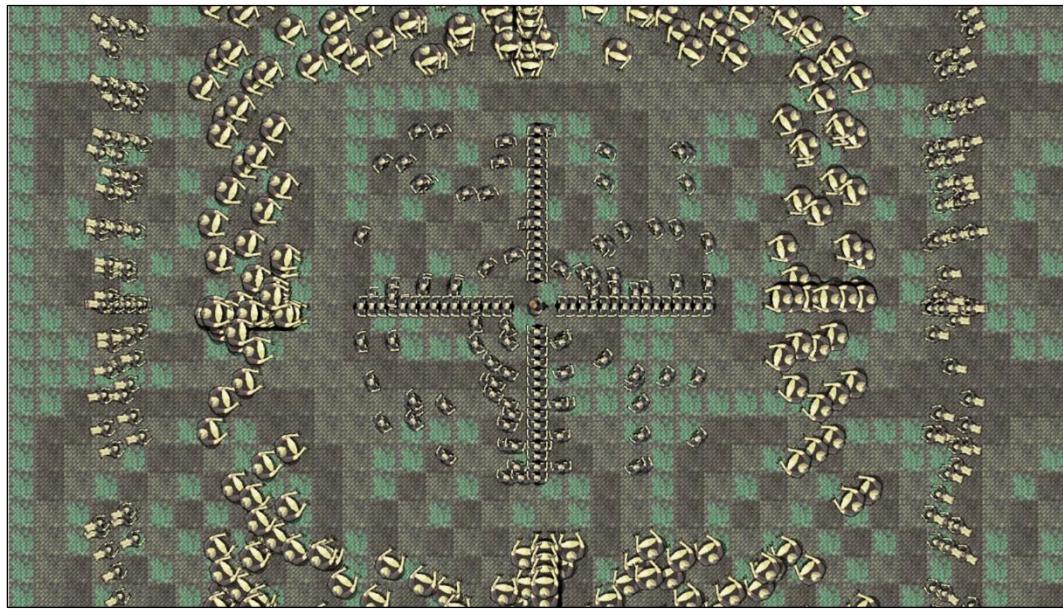




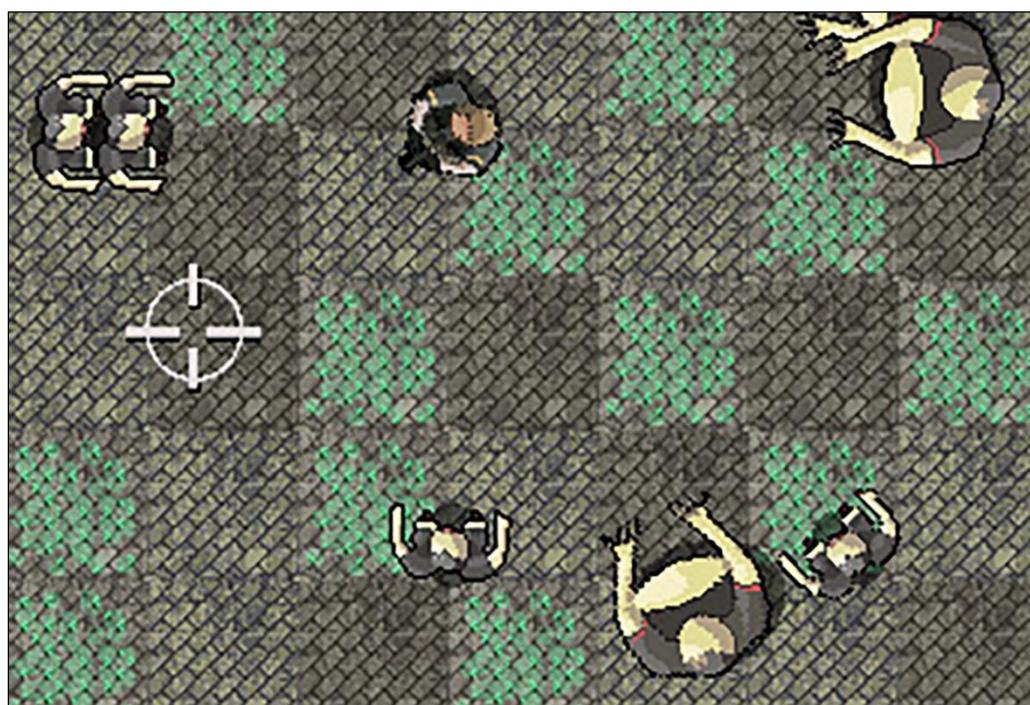


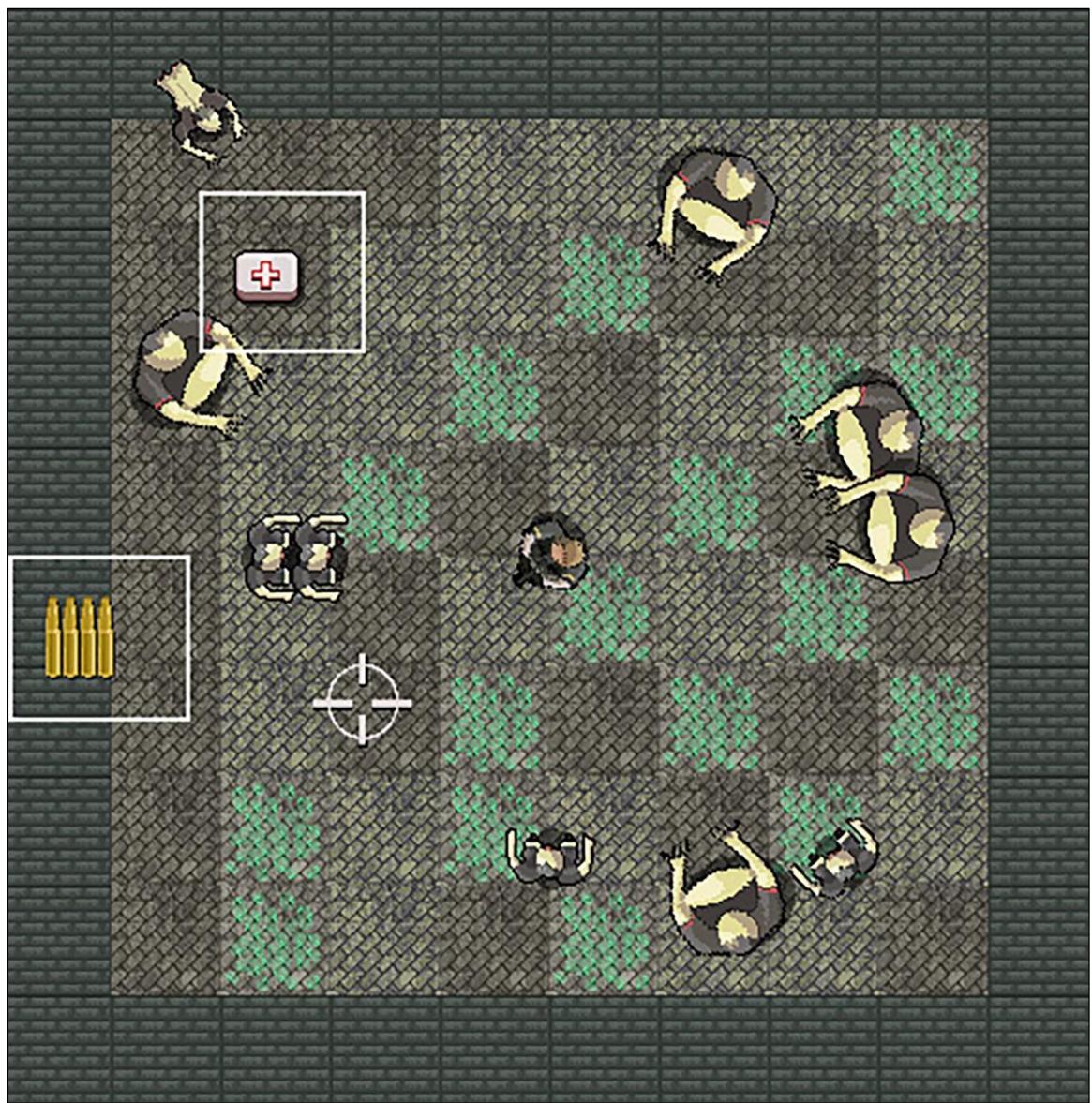
## **Chapter 10: Pointers, the Standard Template Library, and Texture Management**

## Chapter 11: Coding the TextureHolder Class and Building a Horde of Zombies



## Chapter 12: Collision Detection, Pickups, and Bullets







## Chapter 13: 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

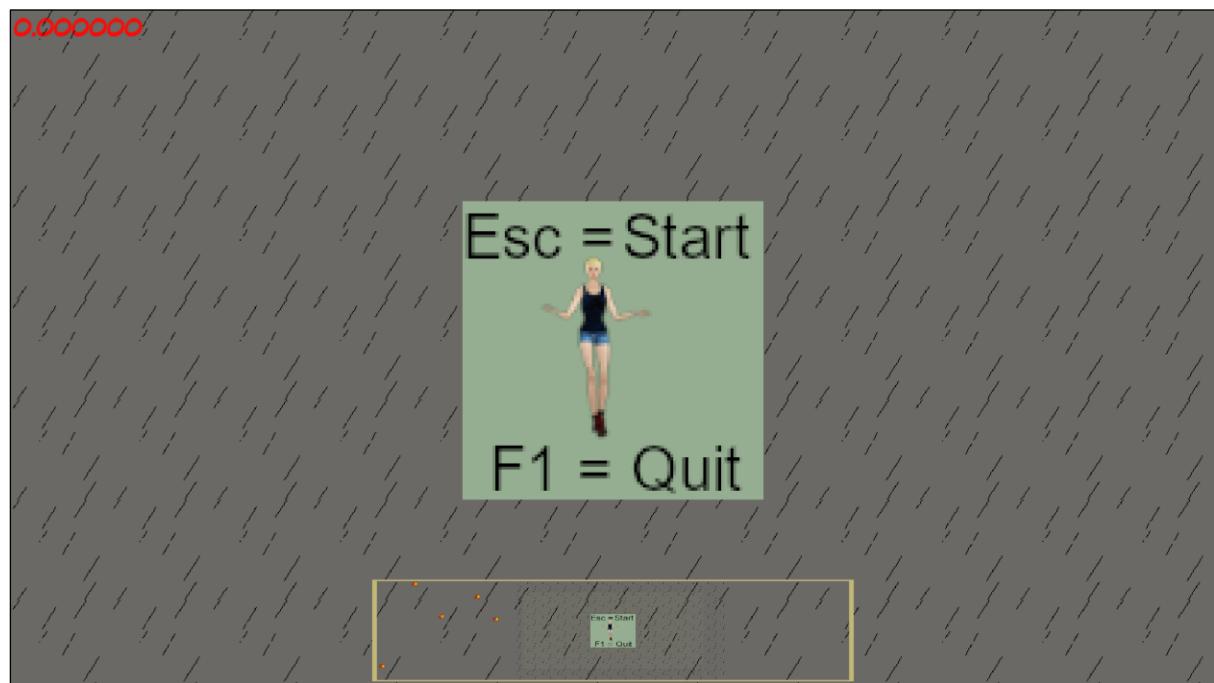
HI SCORE:0

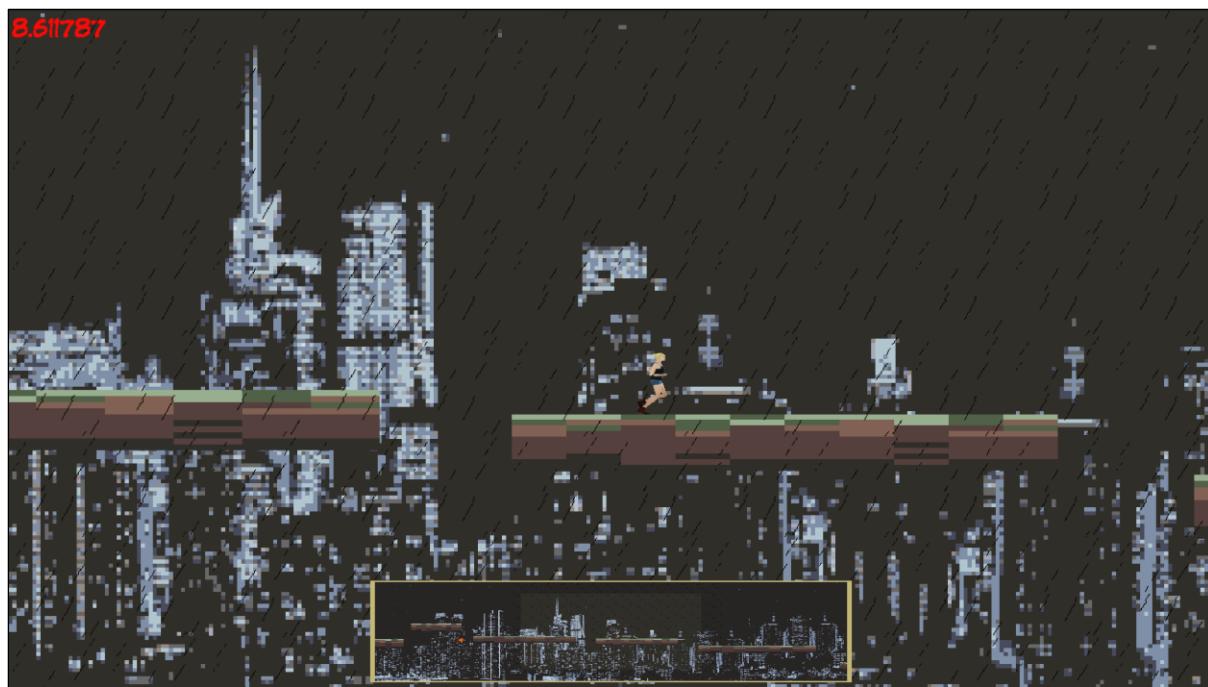
PRESS ENTER  
TO CONTINUE

WAVE: 0 ZOMBIES: 100

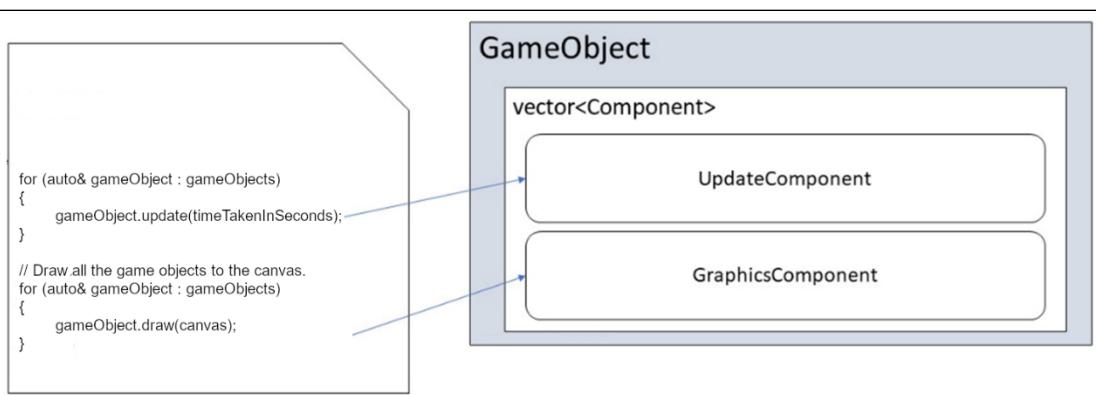
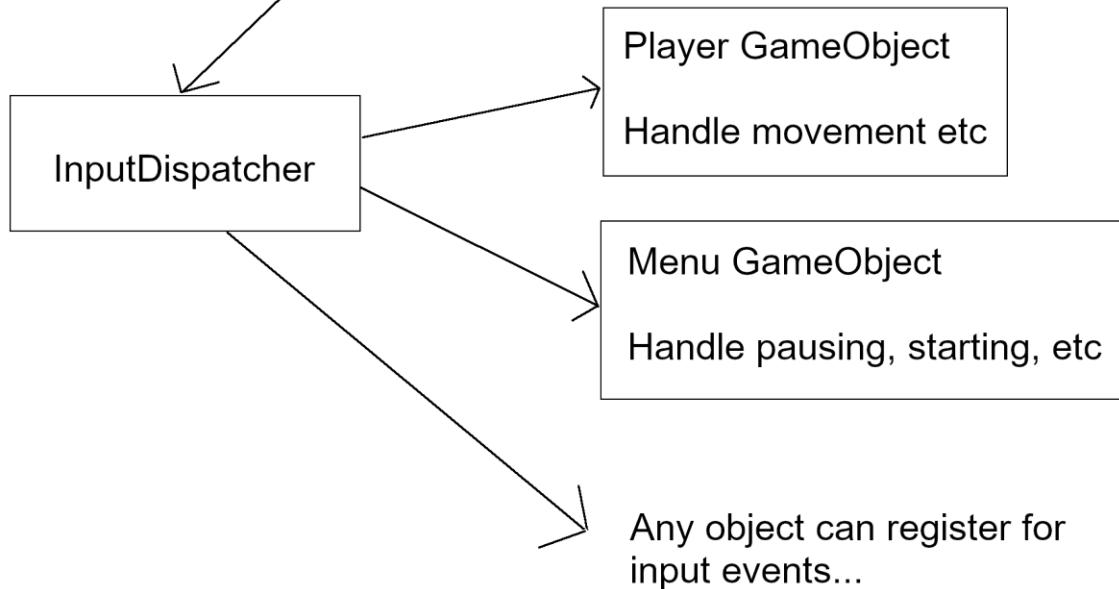
## **Chapter 14: Sound Effects, File I/O, and Finishing the Game**

## Chapter 15: Run!





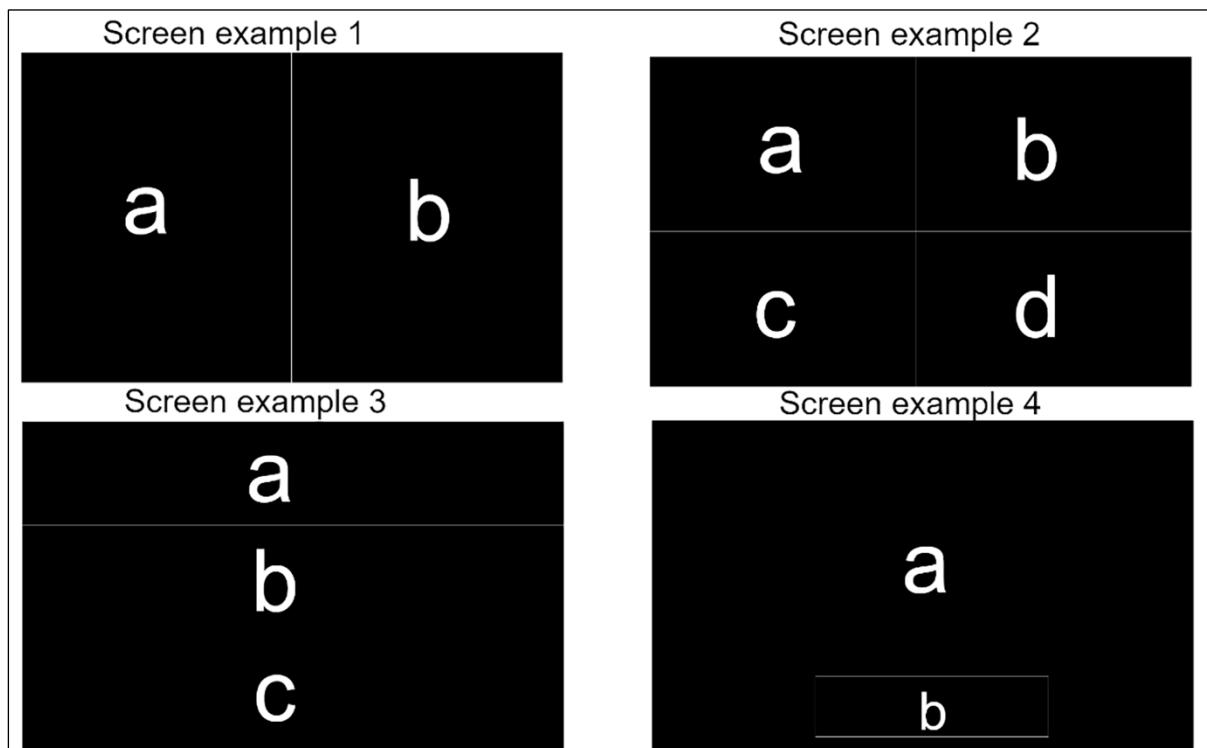
```
main()
{
    ...
    inputDispatcher.dispatchInputEvents();
    ...
}
```



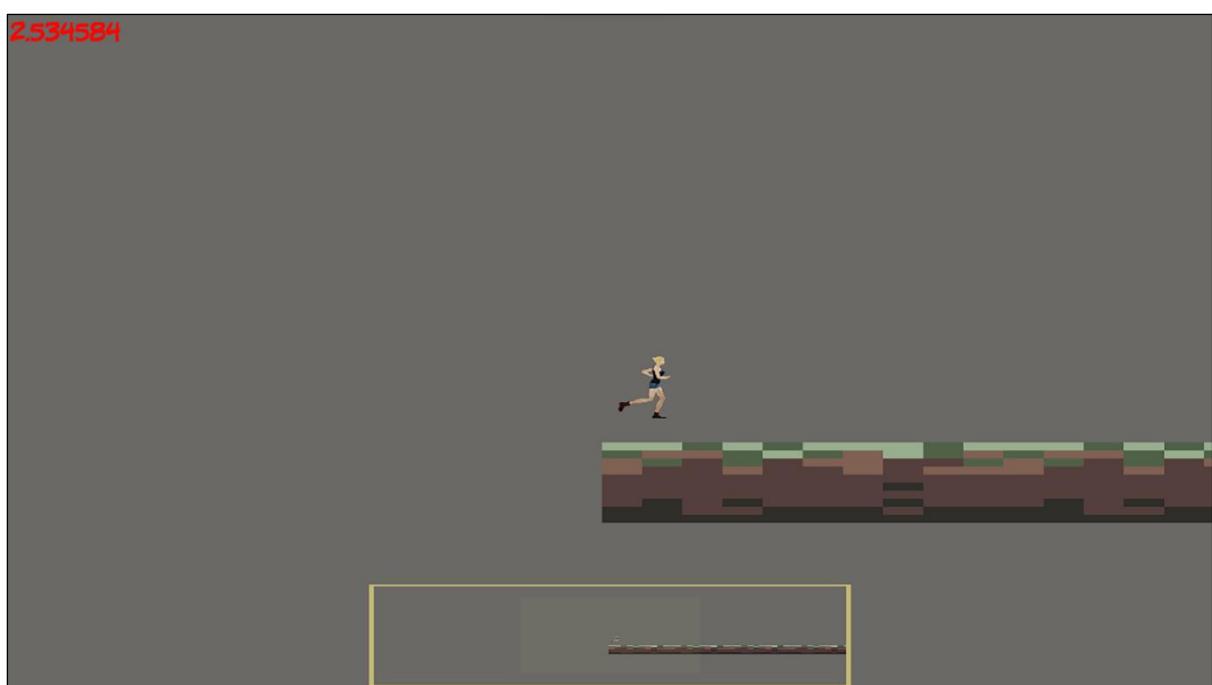
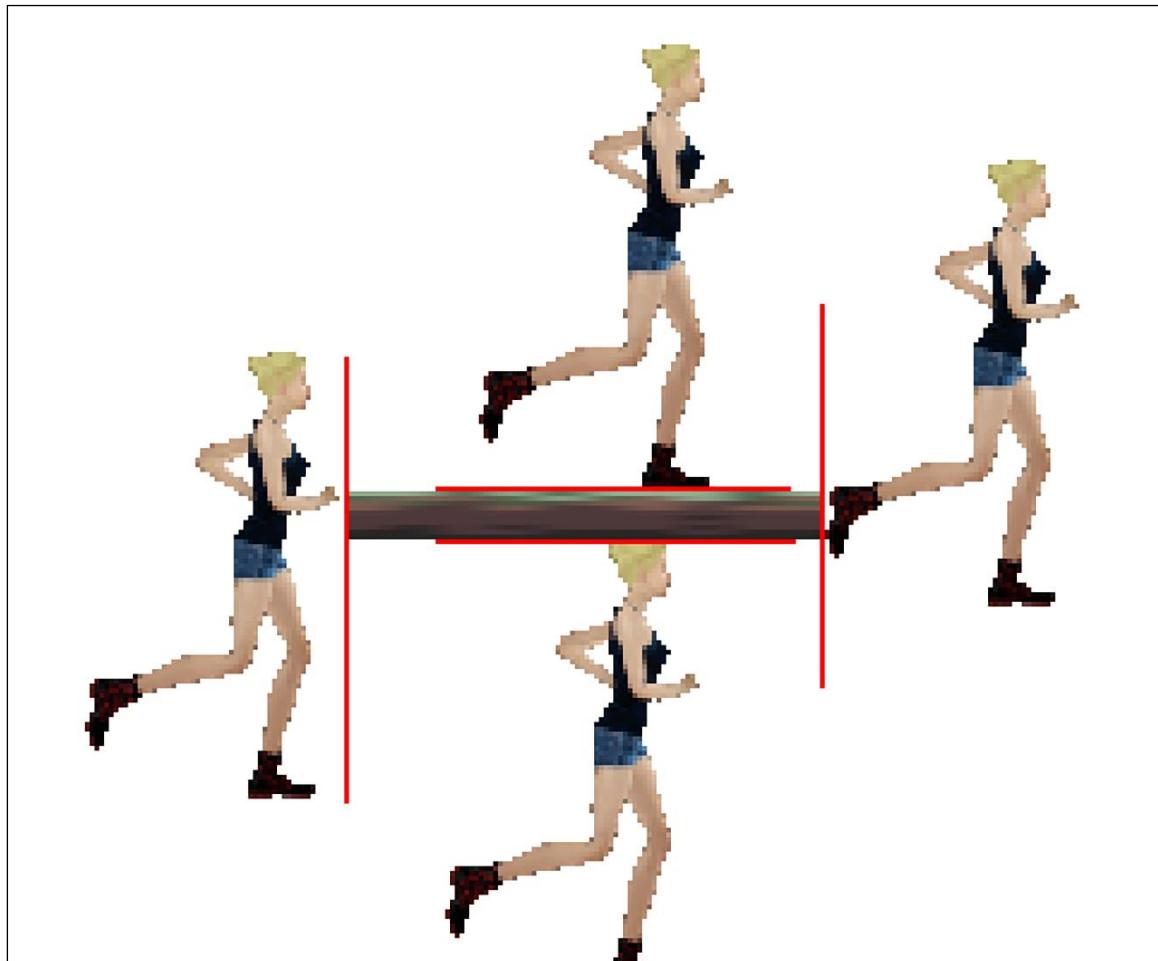
## **Chapter 16: Sound, Game Logic, Inter-Object Communication, and the Player**

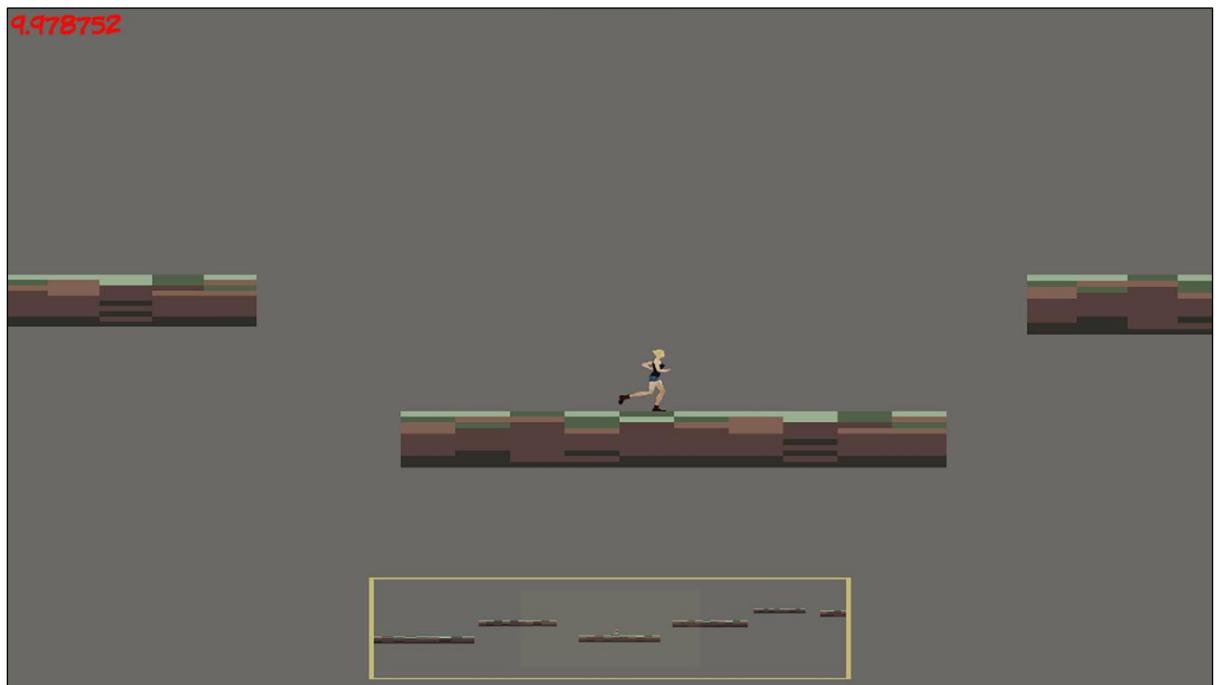


## Chapter 17: Graphics, Cameras, Action



## Chapter 18: Coding the Platforms, Player Animations, and Controls



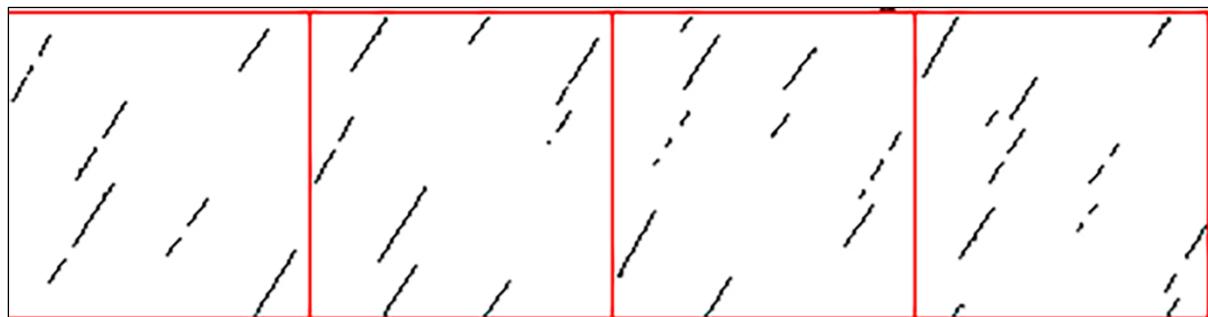
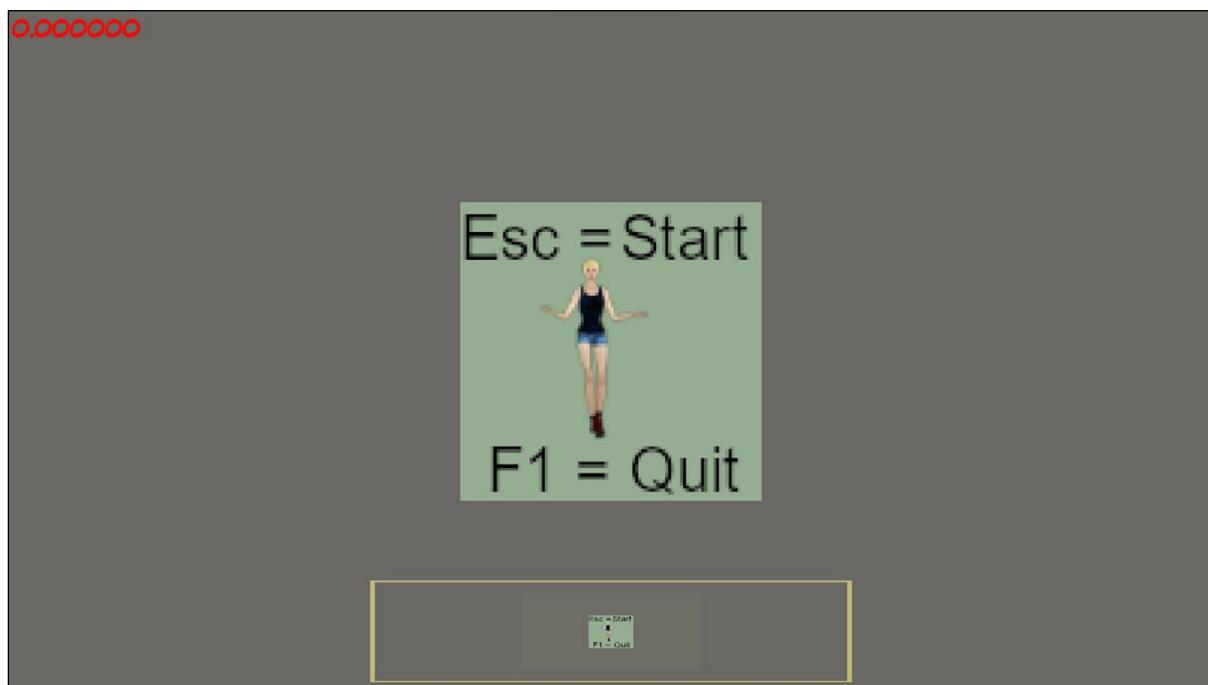
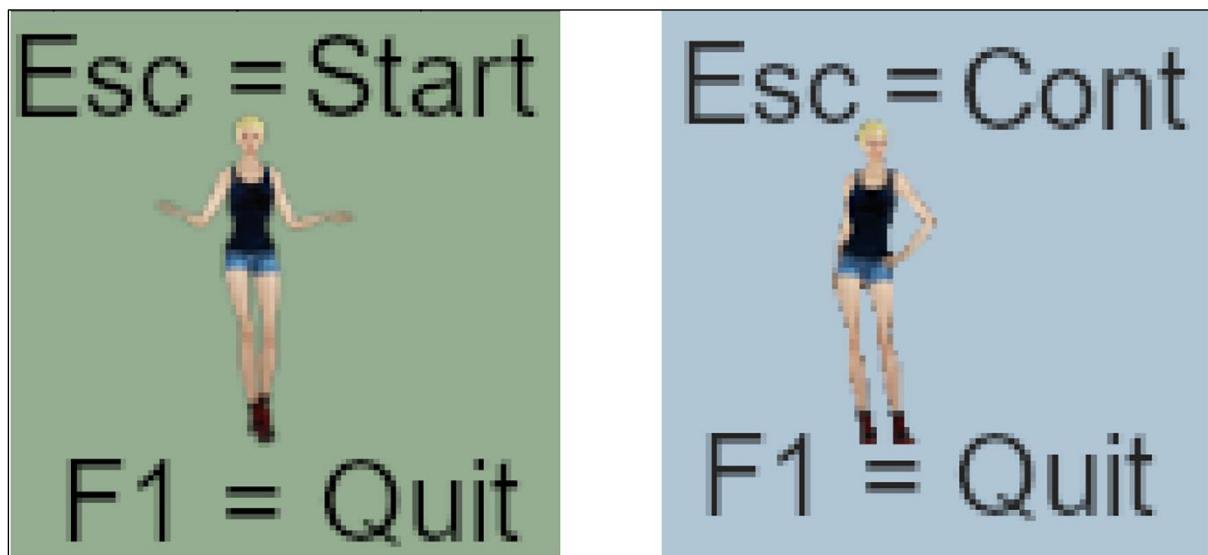


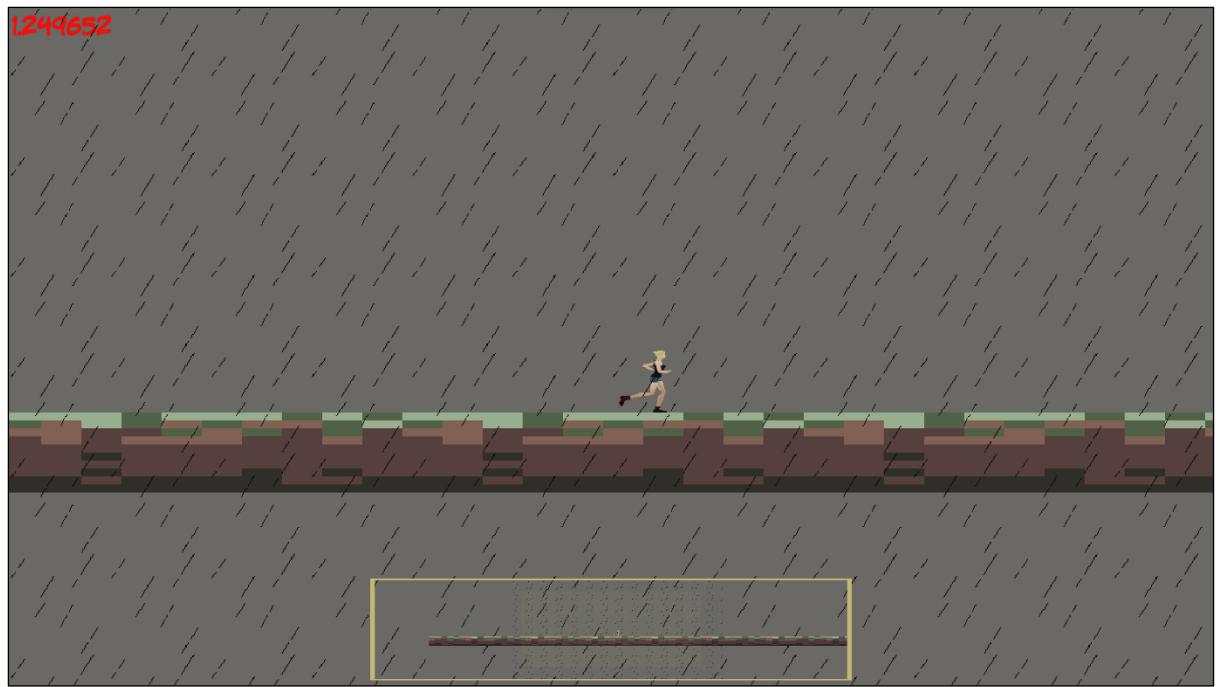




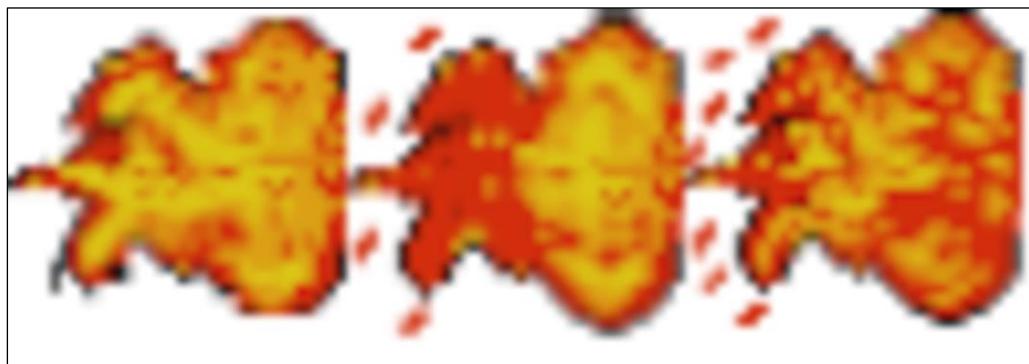


## Chapter 19: Building the Menu and Making It Rain





## Chapter 20: Fireballs and Spatialization



## Chapter 21: Parallax Backgrounds and Shaders

