

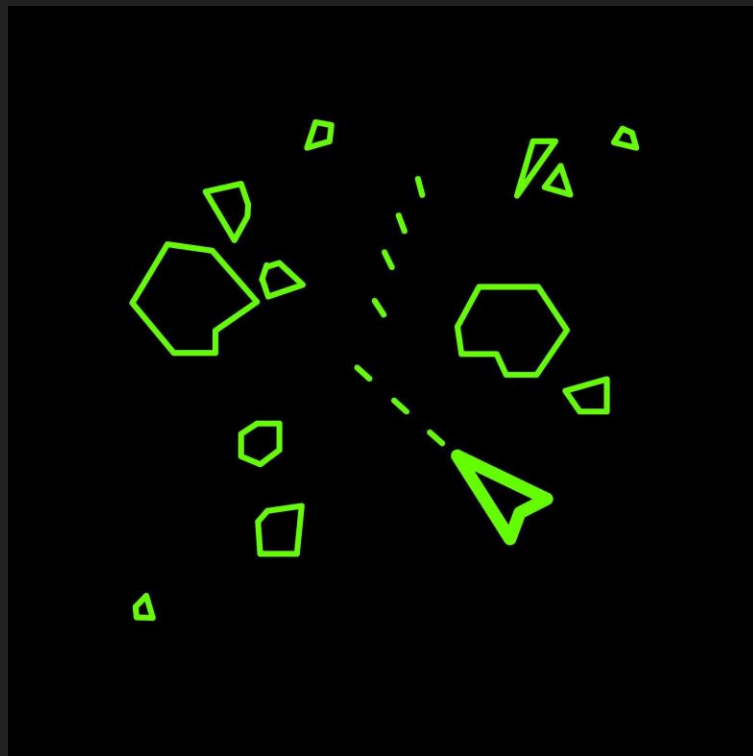
VGP230

Lesson 1

Instructor: Darren Waine

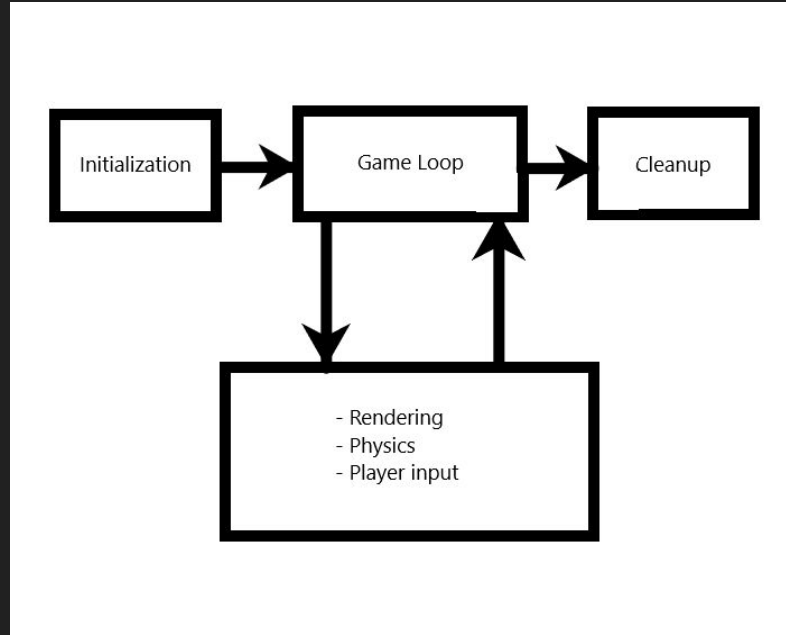
Lesson Overview

- Video game loop basics
- Code pipeline refresh
- Version control refresh
- Creating a Windows App project
- X Engine introduction!
- In-class Project: HelloX



Video Game Loop Basics

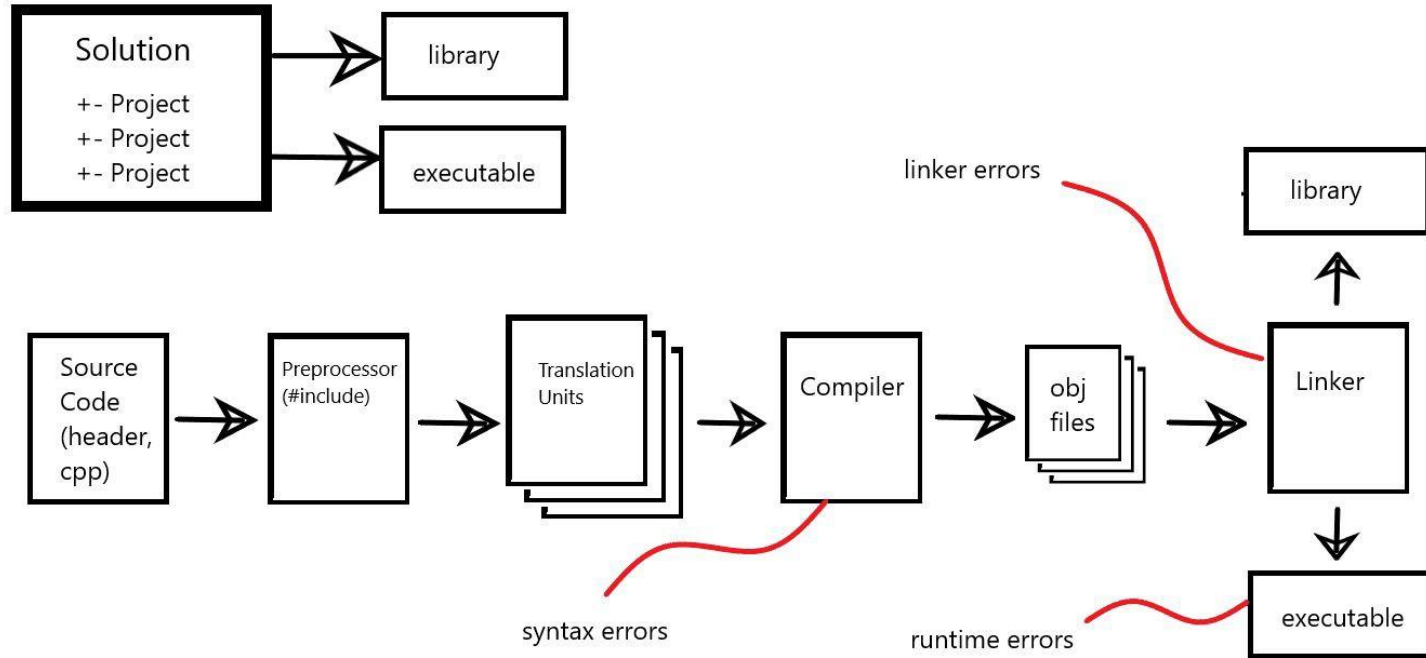
- Initialization
- Game loop
- Cleanup



Code Pipeline Refresh

- All the projects in your solution will either be:
 - Libraries (.lib) or
 - Executables (.exe)
- Source code (headers, cpp files)
- Preprocessor (#include ...)
- Compiler (this is where you get syntax errors)
- Linker (this is where you get linker errors)

Code Pipeline cont.



Version Control refresh

- Also known as source control, revision control, etc
- Why use it?
 - Allows us to keep track of many different iterations/versions of code
 - Branching, merging -> modify changes in isolation!
 - Code checkin ownership (useful for large team projects)
- Examples:
 - Git
 - Perforce
 - Team Foundation Server

Creating a New Project (Windows application)

For the purposes of this class, we want to create a Windows application

- Solution -> Add new project
 - Windows Desktop drop down menu -> Windows Desktop Wizard
 - Windows (also called Desktop) application, empty project
- Add “X.props” property sheet to your new project
 - View -> Property Manager (this is a tab like Solution Explorer)
 - Right click project and add existing property sheet located at ../X/X.props
 - In Solution Explorer, right click project and add reference - X

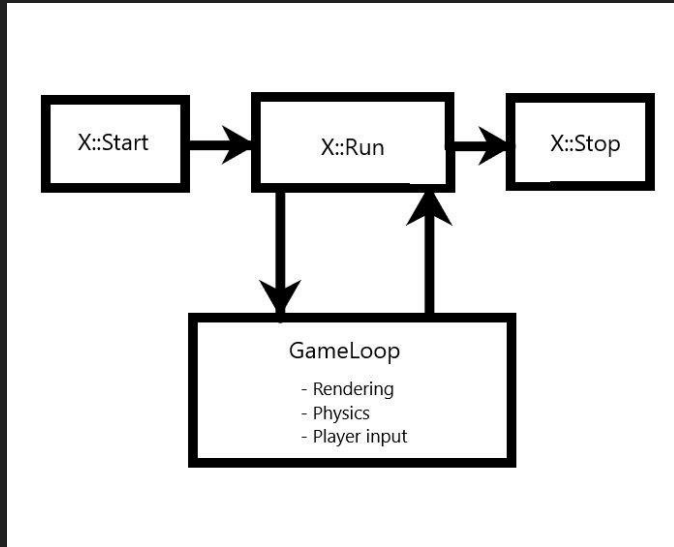
X Engine Introduction

- Engine (Start, Run, Stop)
- Rendering (DrawSprite, DrawScreenCircle, etc)
- Physics/Math (Vec2, Vec3 etc)
- Input (IsKeyDown)
- Coordinate system is X and Y (2D), with positive X going from left to right, and positive Y going from top to bottom.
 - This means the origin (0, 0) is top left!



Game Loop using X

Remember the game loop described earlier? Let's put that into action using X Engine



- `X::Start();`
- `GameInit();`
- `X::Run(GameLoop)`
- `GameCleanUp();`
- `X::Stop();`

Game Loop using X

Code example



All projects within X should usually follow this skeleton structure

```
HelloXEngine (Global Scope)
1  #include <XEngine.h>
2
3  void GameInit()
4  {
5      // Initialize game
6  }
7
8  bool GameLoop(float deltaTime)
9  {
10     // Run game
11
12     return X::IsKeyPressed(X::Keys::ESCAPE);
13 }
14
15 void GameCleanup()
16 {
17     // Cleanup game
18 }
19
20 int WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)
21 {
22     X::Start();
23     GameInit();
24
25     X::Run(GameLoop);
26
27     GameCleanup();
28     X::Stop();
29
30     return 0;
31 }
```

Sources (and useful links!)

- <https://betterexplained.com/articles/a-visual-guide-to-version-control/>