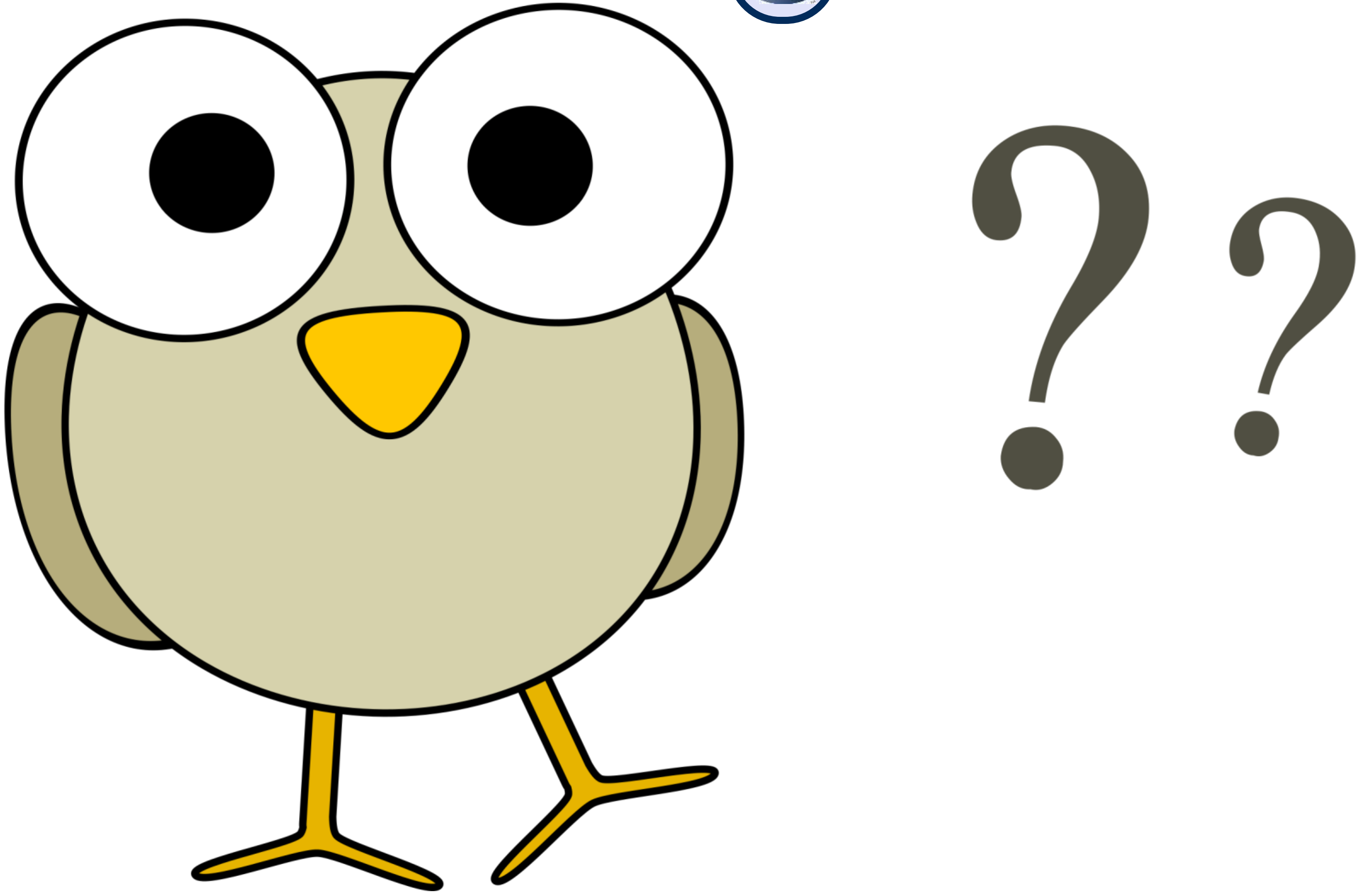
CSCI 200: Foundational Programming Concepts & Design Lecture 27

Making & Using
Libraries

Previously in CSCI 200

- Inheritance
  - Child class inherits members from Parent class

Questions?



Learning Outcomes For Today

- Explain what a library archive is
- Discuss reasons why headers & implementations can be bundled into a library archive and distributed
- Build and install a third-party library (SFML) to use within a C++ program (to display graphics)

On Tap For Today

- Using Libraries
- The SFML Library
- Practice

On Tap For Today

- Using Libraries
- The SFML Library
- Practice

Warehouse

```
#include "Box.h"
#include <vector>

class Warehouse {
public:
   ...
private:
   std::vector<Box*> *_pBoxen;
   ...
};
```

Warehouse

```
#include "Box.h"
#include <vector>

class Warehouse {
public:
   ...
private:
   std::vector<Box*> *_pBoxen;
   ...
};
```

- Where is **`Box.h`** header file located?

Warehouse

```
#include "Box.h"
#include <vector>

class Warehouse {
public:
   ...
private:
   std::vector<Box*> *_pBoxen;
   ...
};
```

- Where is **`Box.h`** header file located?
  - Alongside **`Warehouse.h`**

Warehouse

```
#include "Box.h"
#include <vector>

class Warehouse {
public:
   ...
private:
   std::vector<Box*> *_pBoxen;
   ...
};
```

- Where is **`vector`** header file located?

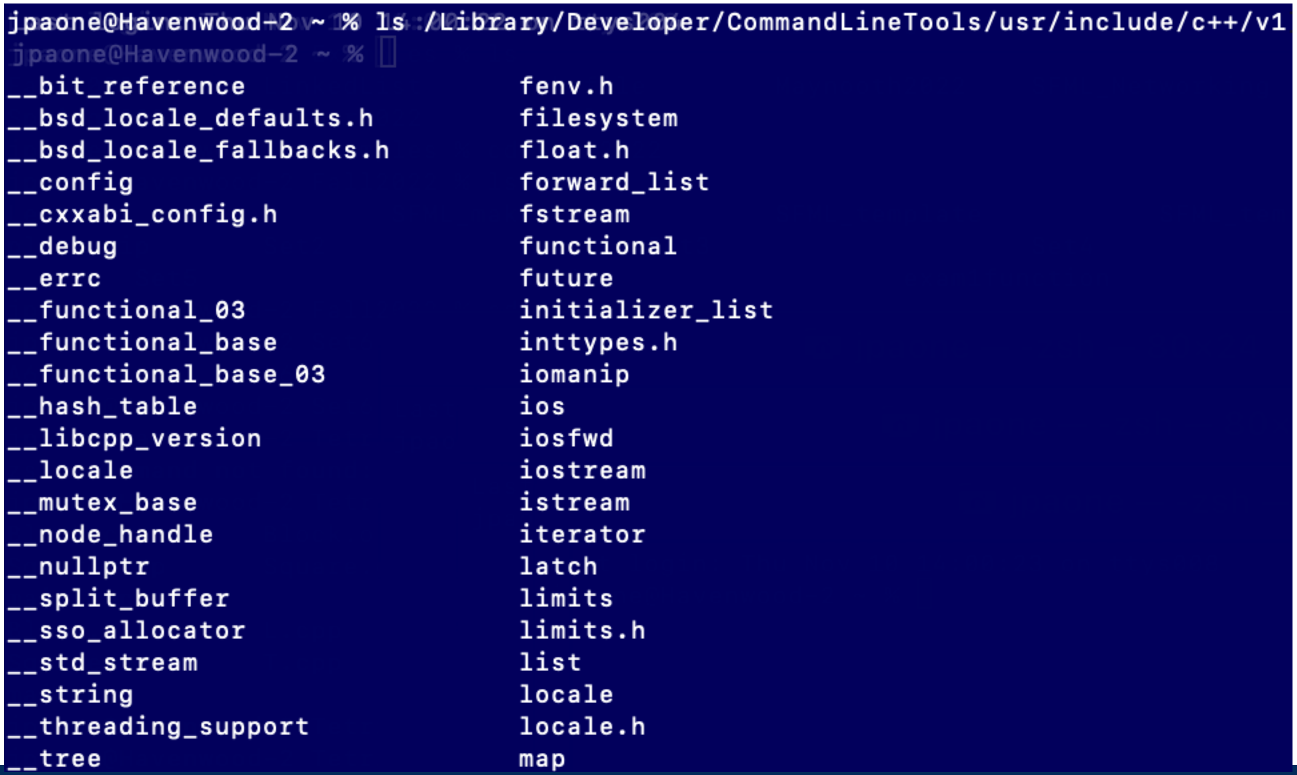
Including Headers

```
#include "LocalHeader.h"


#include <SystemHeader.h>
```

Finding System Headers

- Windows: Inside MinGW hierarchy
- OS X: Inside XCode hierarchy



```
jpaone@Havenwood-2 ~ % ls /Library/Developer/CommandLineTools/usr/include/c++/v1
jpaone@Havenwood-2 ~ %
__bit_reference                fenv.h
__bsd_locale_defaults.h        filesystem
__bsd_locale_fallbacks.h       float.h
__config                       forward_list
__cxxabi_config.h              fstream
__debug                        functional
__errc                         future
__functional_03                initializer_list
__functional_base              inttypes.h
__functional_base_03           iomanip
__hash_table                   ios
__libcpp_version               iosfwd
__locale                       iostream
__mutex_base                   istream
__node_handle                  iterator
__nullptr                      latch
__split_buffer                 limits
__sso_allocator                limits.h
__std_stream                   list
__string                       locale
__threading_support            locale.h
__tree                         map
```

Warehouse

```
#include "Box.h"
#include <vector>

class Warehouse {
public:
   ...
private:
   std::vector<Box*> *_pBoxen;
   ...
};
```

- Where is **`Box`** implementation located?

Warehouse

```
#include "Box.h"
#include <vector>

class Warehouse {
public:
   ...
private:
   std::vector<Box*> *_pBoxen;
   ...
};
```

- Where is **`Box`** implementation located?
  - Alongside in **`Box.cpp`**

Warehouse

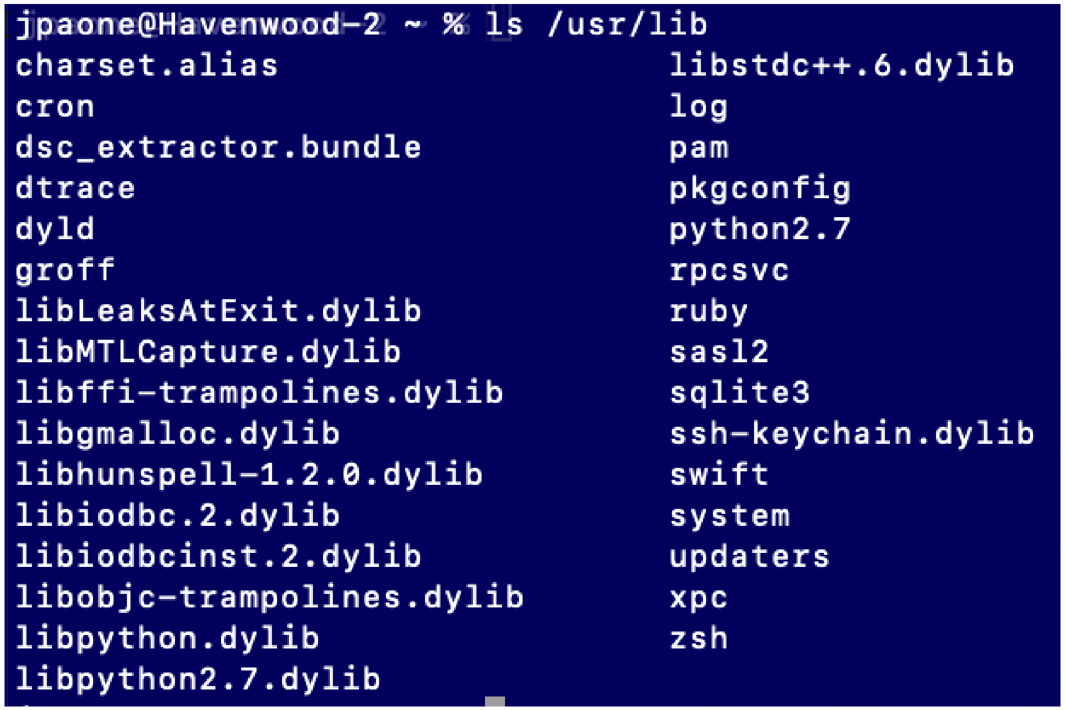
```
#include "Box.h"
#include <vector>

class Warehouse {
public:
   ...
private:
   std::vector<Box*> *_pBoxen;
   ...
};
```

- Where is **`vector`** implementation located?

Finding System Libraries

- Again, from MinGW / XCode
- `libstdc++` contains the C++ Standard Libraries



```
jpaone@Havenwood-2 ~ % ls /usr/lib
charset.alias                     libstdc++.6.dylib
cron                              log
dsc_extractor.bundle              pam
dtrace                            pkgconfig
dyld                              python2.7
groff                             rpcsvc
libLeaksAtExit.dylib              ruby
libMTLCapture.dylib               sasl2
libffi-trampolines.dylib          sqlite3
libgmalloc.dylib                  ssh-keychain.dylib
libhunspell-1.2.0.dylib           swift
libiodbc.2.dylib                  system
libiodbcinst.2.dylib              updaters
libobjc-trampolines.dylib         xpc
libpython.dylib                   zsh
libpython2.7.dylib
```

Archive File

- What's a library archive file?
  - Collection of object files

- What's an object file?
  - Compiled binary representation of C++ source code

Why Use Archive Files As Libraries?

- Single source of truth / Reuse
  - Headers & Implementations live in one place for ALL programs to access & use
- Distribution
  - Share a stable version of classes/functions for others to use

Distribution

1. Share header files
   - Serves as form of documentation for interface that is available
   - Necessary for other programs to include to be able to compile with your declared components

2. Share precompiled object files
   - Implementation doesn’t change (compile once, use many)
   - Abstract (& hide) details of implementation (may contain proprietary algorithm)

Header Best Practices

- Headers should be minimal, only include what is absolutely necessary for interface
- Don’t include any headers in your header that aren’t required in that file
  - Anything that can go into source file should
- Don’t use `using namespace` which forces others to as well

Informing the Compiler

- Tell the compiler where to look for header files

- Compiler command template

```
$(CXX) $(CFLAGS) -o $(OBJ_NAME) -c $(SRC_NAME) -I$(INC_PATH)
```

- For instance

```
g++ -Wall -g -std=c++17 -o main.o -c main.cpp -IZ:\CSCI200\include
```

- Specify location in Makefile

Informing the Linker

- Tell the linker where to look for library files
- Linker command template

```
$(CXX) -o $(TARGET_NAME) $(OBJECT_NAMES) -L$(LIB_PATH) $(LIBS)
```

- For instance

```
g++ -o SFMLExample.exe main.o -LZ:\CSCI200\lib -lsfml-system
```

- Specify location and libraries in Makefile

Makefile Updates

```
# customize your build
TARGET = TestLibrary
SRC_FILES = main.cpp
CXX = g++
CXXFLAGS = -Wall -Wextra -Wpedantic -g -std=c++17
INC_PATH = Z:\CSCI200\include # location where headers are
LIB_PATH = Z:\CSCI200\lib     # location where archive files are
LIBS = -lTestLib              # name of archive file to load
# do not edit below here
OBJECTS = ${SRC_FILES:.cpp=.o}
DEL = ...
all: ${TARGET}
${TARGET}: ${OBJECTS}
    ${CXX} -o $@ $^ -L${LIB_PATH} ${LIBS}
.cpp.o:
    ${CXX} ${CXXFLAGS} -o $@ -c $< -I${INC_PATH}
clean:
    ${DEL} ${TARGET} ${OBJECTS}
```

On Tap For Today

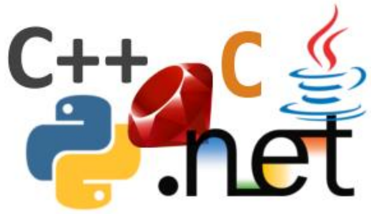
- Using Libraries
- The SFML Library
- Practice

SFML

- Simple & Fast Multimedia Library

Multimedia

Multiplatform



Multilanguage

What is SFML?

- http://www.sfml-dev.org
- Multimedia = Graphics & Audio
- Used for:
  - Games
  - Data Visualization
  - Networking
  - And much much more!

SFML Libraries

- SFML source code download consists of:
  - `SFML-3.0.2/`
    - `include/`
    - `lib/`
    - *`Bunch of other stuff`*
- Need to copy the
  - header files from `include/`
  - library files from `lib/`
- This is the start of Lab4C

Important Version Note

- Using v3.0.2 for this semester
  - Make sure any tutorials, references, resources are using v3.0 and not v2.6 or earlier

SFML Header

- Include like any other library file

```
#include <SFML/Graphics.hpp>
```

- Provides functions and complex data types that are used to display graphics

- (Other SFML headers exist as needed)

Sample SFML Program

```cpp
#include <SFML/Graphics.hpp>

int main() {
  // add File I/O commands here so they occur once!
  sf::Vector2u windowSize(640, 640);
  sf::RenderWindow window( sf::VideoMode( windowSize ),
                           "Window Title" ); // create & open a window
  while( window.isOpen() ) {
    window.clear();  // clear the existing contents of the window
    // add drawing commands here so they draw every frame
    window.display();// display the window on screen
    // check for user interaction
    while( const std::optional event = window.pollEvent() ) {
      if( event->is<sf::Event::Closed>() ) { // user press window X
        window.close(); // close the window
      }
    }
  }
  return 0;
}
```

Drawing Shapes

- Circle

```
sf::CircleShape circ;
circ.setRadius( 20.f );
circ.setPosition( sf::Vector2f(45.f, 90.f) );
circ.setFillColor( sf::Color::Yellow );
window.draw( circ );
```

- Rectangle

```
sf::RectangleShape rect;
rect.setSize( sf::Vector2f(150.f, 75.f) );
rect.setPosition( sf::Vector2f(115.f, 120.f) );
rect.setFillColor( sf::Color::Blue );
window.draw( rect );
```

Adding Color

- Use the `setFillColor()` function
- Use the `setOutlineColor()` and `setOutlineThickness()` functions

- Several options:
  - Built-in values: `sf::Color::`***Red***, `sf::Color::`***Green***, etc.
  - Custom value: `sf::Color(` *red*, *green*, *blue* `)`

Drawing Text: Part I

- First, we need to load a font to use
- And make sure it loaded properly

```
sf::Font myFont;
if( !myFont.openFromFile( "data/arial.ttf" ) ) {
  cerr << "Could not load font" << endl;
  return -1;
}
```

Drawing Text: Part II

- Now set up our Text object using the font

```
Text myLabel( myFont );
```

- Make use of text functions to set properties

```
myLabel.setString( "Hello World!" );

myLabel.setFillColor( sf::Color::Green );

myLabel.setPosition( sf::Vector2f(400.f, 150.f) );
```

- And tell window to draw it

```
window.draw( myLabel );
```

Displaying a Picture: Part I

- First, load the image into memory
  - Like text, we needed to load the font into memory

```
sf::Texture myTexture;
if( !myTexture.loadFromFile( "data/bubble.png" ) ) {
    cerr << "Could not load image" << endl;
    return -2;
}
```

Displaying a Picture: Part II

- Now add it to a Sprite and draw!
  - Sprite is like RectangleShape with a picture
  - Ta Da!

```
sf::Sprite mySprite( myTexture );
mySprite.setPosition( sf::Vector2f(200.f, 250.f) );
mySprite.setScale( sf::Vector2f(0.3f, 0.3f) );
mySprite.setColor( sf::Color::Green );
window.draw( mySprite );
```

Animation: Part I

- Need to store position as a variable

```
sf::Vector2f spritePos(0.f, 0.f);

while( window.isOpen() ) {
  // in draw loop
  mySprite.setPosition( spritePos );
  window.draw( mySprite );
}
```

Animation: Part II

- Each iteration of draw loop = one frame

```
#include <SFML/System/Clock.hpp>

// before draw loop

sf::Vector2f spritePos(0.f, 0.f);

sf::Clock programClock;
sf::Time lastTime = programClock.getElapsedTime();

while( window.isOpen() ) {

  // in draw loop
  mySprite.setPosition( spritePos );
  window.draw( mySprite );

  sf::Time currTime = programClock.getElapsedTime();
  if( (currTime - lastTime).asMilliseconds() > THRESHOLD ) {
    spritePos.x += dx;
    spritePos.y += dy;
    lastTime = currTime;
  }

}
```

Interaction = Event Handling

- Ways user can interact
  - Via keyboard
    - Key press
    - Key release
  - Via mouse
    - Button press
    - Button release
    - Mouse movement
  - And others (window minimize, lose focus, etc.)

Keyboard Interaction

- Can have user close window via keyboard

```
// in draw loop
while( std::optional event = window.pollEvent() ) {
  if( const sf::Event::KeyPressed * keyPressed =
                event->getIf<sf::Event::KeyPressed>() ) {
    if( keyPressed->scancode == sf::Keyboard::Scan::Q ) {
      window.close();
    }
  }
}
```

Mouse Clicks

- Have sprite jump to mouse click location

```
// in draw loop
while( std::optional event = window.pollEvent() ) {
  if( const sf::Event::MouseButtonPressed * mouseButtonPressed =
                    event->getIf<sf::Event::MouseButtonPressed>() ) {
    // mouseButtonPressed->button   - which button was pressed
    // mouseButtonPressed->position - mouse location when press occurred
  }
}
```

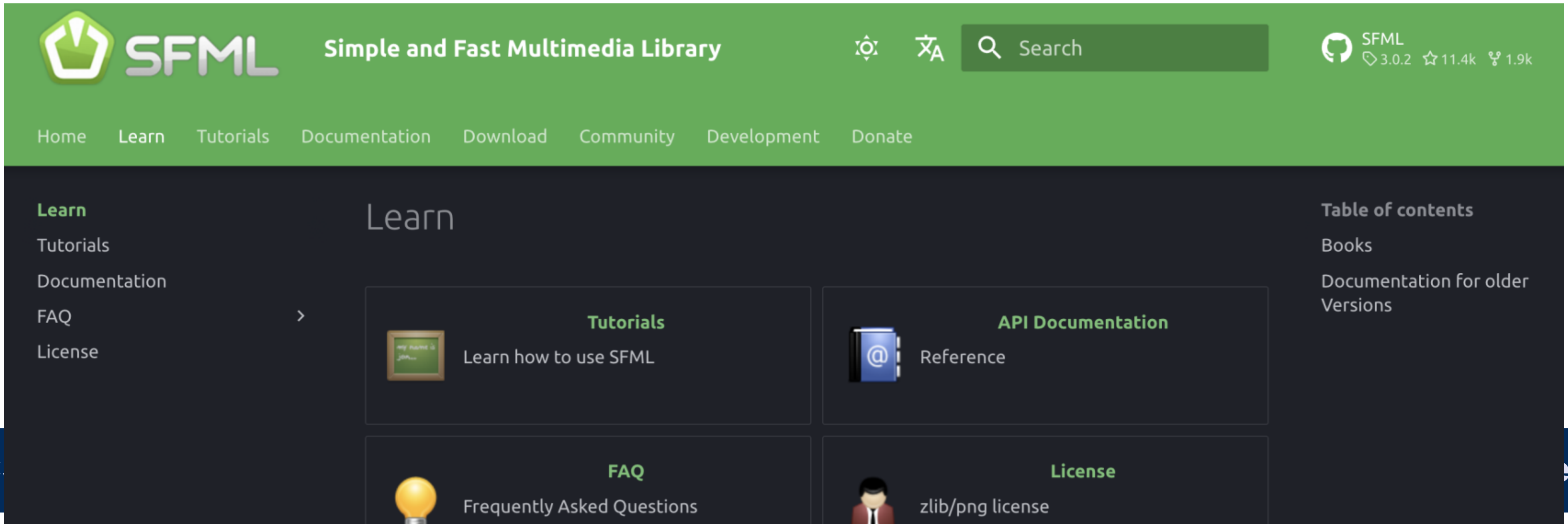
Event Handling

- Check all event types

```
while( std::optional event = window.pollEvent() ) {
  if( event->is<sf::Event::Closed>() ) {
    // close window
  } else if( const sf::Event::KeyPressed * keyPressed =
                 event->getIf<sf::Event::KeyPressed>() ) {
    // do something based on which key is pressed
  } else if( const sf::Event::MouseButtonPressed * mouseButtonPressed =
                 event->getIf<sf::Event::MouseButtonPressed>() ) {
    // do something based on mouse click location/button
  }
  // check other events that could occur
}
```

SFML Documentation

- Tutorials: https://www.sfml-dev.org/tutorials/3.0/
- API Documentation: https://www.sfml-dev.org/documentation/3.0.2/
- FAQ: https://www.sfml-dev.org/faq.php


SFML
Simple and Fast Multimedia Library
Search
SFML 3.0.2 11.4k 1.9k
Home Learn Tutorials Documentation Download Community Development Donate

Learn
Tutorials
Documentation
FAQ
License

Learn

Tutorials Learn how to use SFML

API Documentation Reference

FAQ Frequently Asked Questions

License zlib/png license

Table of contents
Books
Documentation for older Versions

Building Against SFML – Makefile

```
# if not placed in system folders
# location where SFML headers are
INC_PATH = Z:\CSCI200\include
# location where SFML archive files are
LIB_PATH = Z:\CSCI200\lib

# name of archive files to load
LIBS = -lsfml-window -lsfml-graphics -lsfml-system -lsfml-audio -lsfml-network
```

On Tap For Today

- Using Libraries
- The SFML Library
- Practice

To Do For Next Time

- L4C
  - Build and install SFML on your machine
  - OS dependent but cross-platform
  - YMMV – ask for help on Ed!
- A4
  - Draw balls bouncing around the screen
  - Add/remove balls
- Ask for help on Ed about setting up SFML!