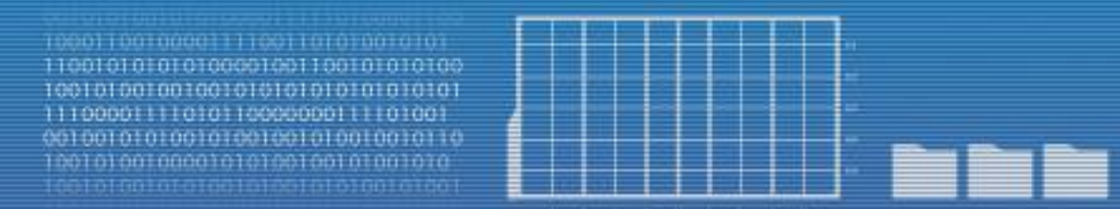
DEPARTMENT OF INFORMATION SYSTEMS AND COMPUTER SCIENCE

Timing

Time for SFML

Lecture Time!

- Getting Started: Guided Tutorial
- Timing is Everything: Using the Clock in SFML

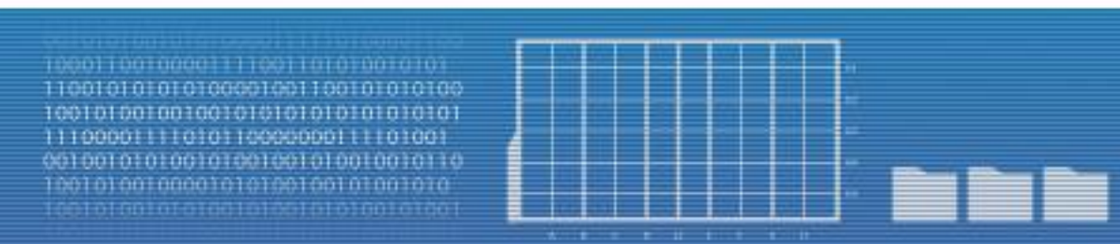


Tutorials

- You could just follow the tutorials in http://www.sfml-dev.org/learn.php
  - I recommend doing so if you're a fast learner and/or you're super excited
- Or we could accelerate the process through these slides

Header File

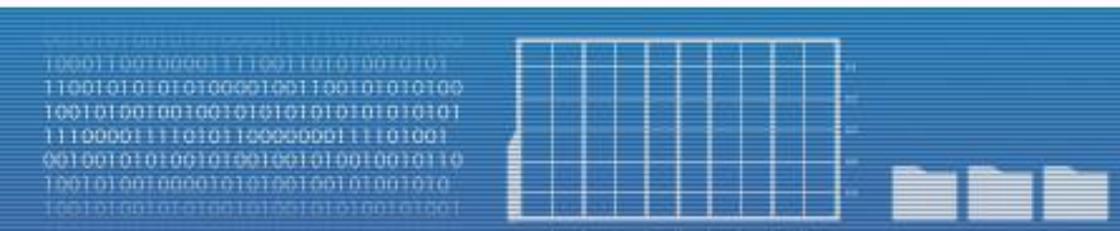
- `#include <SFML/Graphics.hpp>`
- Optional: `using namespace sf;`
  - Things are going to get confusing if you already use the `std` namespace, so I don't recommend doing this



SFML Window (Initialization)

```
int main( void )
{
        sf::RenderWindow window( sf::VideoMode(
                WINDOW_W, WINDOW_H ), "AppTitleHere" );
        window.setFramerateLimit( FPS );
        // ...
```

- ~~What, did you think we'd be doing text games forever?~~
- Purpose of variables in all-caps should be obvious



SFML Clock

- Timing is very important for most games
  - You want some things to happen after a certain amount of time has passed
  - You want to prevent unfair spamming of player abilities
  - You want to maintain a certain number of frames per second without burning out your player's video card
  - What other time-dependent events do you want?

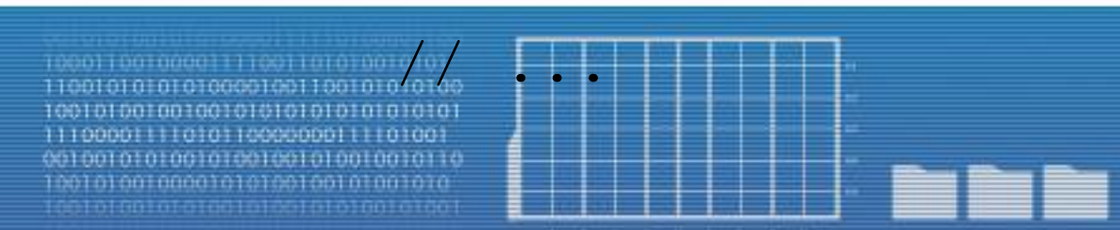
SFML Clock

```
// ...
sf::Clock clock;
sf::Time elapsedTime = sf::seconds( 0 );
sf::Time anotherTime = sf::milliseconds( 17914 );
// ...
```

- Remember, a constructor is called when you declare an object variable
  - Declaring an SFML clock starts it

SFML Clock

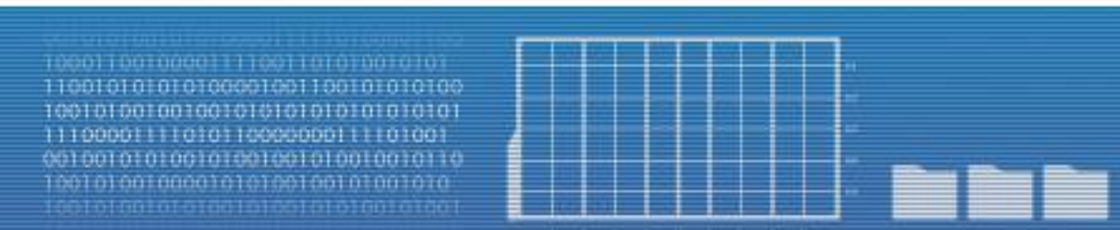
```
int main( void )
{
        // ...
        while( yourAppIsRunning )
        {
                // do stuff
                sf::Time iterationTime = clock.restart();
                elapsedTime += iterationTime;
                // do stuff that depends on how much time passed
                // since either the clock started (elapsedTime)
                // or the clock's last restart() call
                //                                (iterationTime)
        }
        // ...
```



DISCS

SFML Clock

- In other programs/libraries/etc., frame rate can be controlled using the system clock and a `sleep()` function
- Doesn't quite work very well in SFML so don't bother
  - Use `RenderWindow`'s `setFramerateLimit()` function instead



Reminders

- I have already provided a batch file to make your life easier (`make.bat`)
- Usage: `make FileToCompile.cpp`
- Assuming no errors, the result should be an executable (`a.exe`)
  - You may edit the batch file to change this, but this requires some knowledge of the `g++` compiler syntax

Homework

- Edit the program we used to test SFML (`main.cpp`) so that it prints out:
  - time it took for the last iteration (should be zero if this is the first), and
  - total time elapsed so far (should include current iteration, so this should NOT be zero when first displayed)

Homework

- In addition, disable/delete the mouse input portion of the code
- The program must be edited such that the circle's color will change to a random color every 3 seconds
  - The color must be chosen from a pool of 5 instead of 3 colors

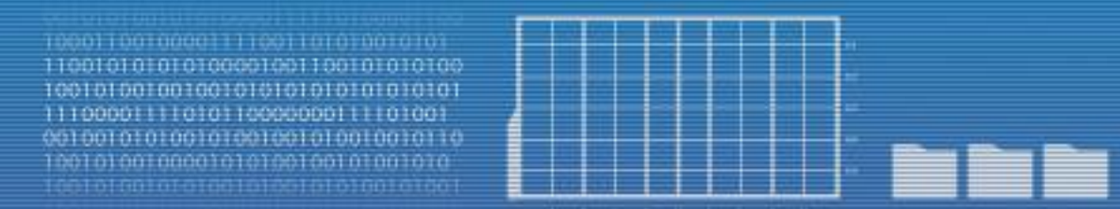
Homework

- Speaking of random:

```
#include <cstdlib>    // header for srand() & rand()
#include <ctime>      // header for time()


// somewhere in initialization
srand( time( NULL ) );

// to get a random number from 0 to some maximum
someInt = rand() % SOME_MAX_PLUS_ONE;
```

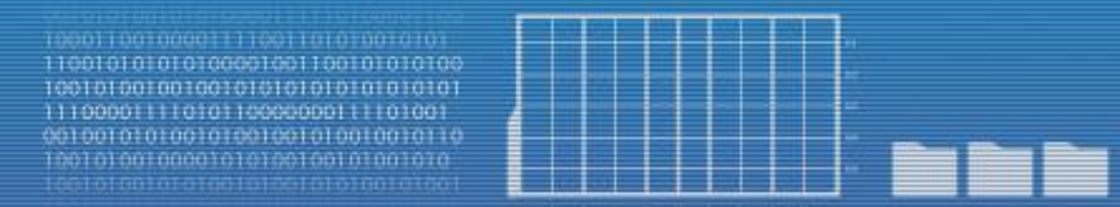


Homework

- For the time outputs, use the following format:
  - N seconds and X milliseconds
  - X should be a whole number between 0 and 999, inclusive
  - N should be a whole number greater than or equal to 0

Homework

- How to get the time values you need:
  - https://www.sfml-dev.org/tutorials/2.4/system-time.php
  - You can also dig around for preset color variables in other tutorials or the API docs
- How to format the values after you've extracted what you need:
  - You can use the cplusplus.com site to search for "truncate", "cast float as int", etc.



Homework

- And don’t forget peer evals
- Follow instructions stated in the syllabus
  - I may have ignored issues from the first peer evaluations, but I will start handing out deductions for those who don’t follow instructions starting today