

**Allegro:** a game programming API (application programming interface).

A collection of functions for:

- opening windows
- drawing into them
- handling user input (mouse, keyboard)
- (and more)

Provided as:

- a set of header files (allegro.h and others)
- a link library (alld.lib)

Installed in the CS Dept Windows Labs in these locations:

- C:\Program Files\Microsoft Visual Studio\VC98\Include
- C:\Program Files\Microsoft Visual Studio\VC98\Lib

Available from: [alleg.sourceforge.net](http://alleg.sourceforge.net)

Some graphics terms:

- pixel  $\equiv$  a location in memory
- all pixels => "framebuffer" ("display buffer", "buffer")
  - framebuffer = color information.
  - draw into the framebuffer then display it => one frame of animation
- "double-buffering"
  - Front and back buffers: Front = display, Back = draw into
  - "swap buffers"

**Win32 Console Application:** Settings -> Link tab, Category: General  
Project Options: /subsystem:console

- Operating system opens a *command-line* window
- I/O is text-based (keyboard only: iostream, cstdio)
- No graphics

**Win32 Application** Settings -> Link tab, Category: General  
Project Options: /subsystem:windows

- Operating system opens a *graphics* window
- I/O is GUI-based (mouse, etc). No cin, cout, printf !

- Allegro API makes it easier for programmer to manage opening windows, drawing into them, handling the mouse, etc.

Event-driven programming

Program behavior depends on 'events' that occur, which are then handled. Events can be things like mouse click, menu selection, key press, timer expires, ... etc. (for example, a Tetris piece will rotate when a certain keyboard key is pressed).

This programming paradigm is the way all graphical user interfaces work (like the MacOS, XP, ...), and how interactive graphical applications work (like WoW and other video games).