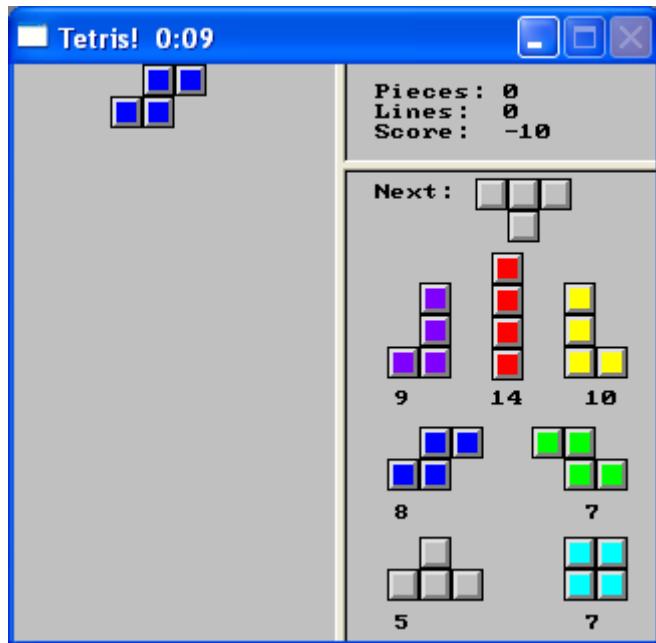


Big picture overview of our version of Tetris!

Display divided into 4 areas

1. title bar: time counts up
2. play area (left side)
3. score stats (upper right)
4. pieces left to play (lower right)



Interaction:

1. keyboard only
2. rotate a piece
3. translate a piece
4. pause the game

Game play:

1. fixed amount of play time
2. fixed # pieces to play
3. bonus for completed lines

Our approach to implementing this game:

1. Come up with a top-down design
2. Implement as a set of *stub functions*
3. Fill in the details:
 - a. Add basic display graphics
 - (1) the board
 - (2) pieces
 - (3) text
 - b. Add timer capability
 - c. Add basic keyboard interaction
 - (1) ESC quit
 - (2) pause
 - d. Game play

Top-down design of Tetris:

Tetris00

```
main
  init -----> init
  main_loop           init API
  cleanup            init game
```

- Create a **Win32 Application** Project named **Tetris00**
- Project Settings -> Link, Category: Input, Object/library modules ...

add "alld.lib" at the end of the list.
- This will link in the Allegro library.
- Add these two header and source code files:

```
----- // main.h ----- // init.h
#ifndef _MAIN_H
#define _MAIN_H

void main_loop( );
#endif

----- // main.cpp ----- // init.h
#include <allegro.h>

#include "main.h"
#include "init.h"

int main( )
{
    init( );
    main_loop( );
    cleanup( );

    return 0;
}
END_OF_MAIN()

void main_loop( )
{
}

----- // init.h ----- // init.cpp
#ifndef _INIT_H
#define _INIT_H

void init( );
void init_allegro( );
void init_tetris( );

void cleanup( );
#endif

----- // init.cpp -----
```

```
END_OF_MAIN() <-- no semicolon!
```

(This is an example of something called a *macro*). It is required in an Allegro program, and you MUST put this right after your function main().

It is #defined in allegro/platform/alwin.h.

It defines a function named WinMain, a required function for all Win32 Applications.

Note that it just sets up to call **our own function** main():

```
#define main _mangled_main
```

```
#define END_OF_MAIN() \
int __stdcall WinMain(void *hInst, void *hPrev, char *Cmd, int nShow) \
{ \
    return _WinMain((void *)_mangled_main, hInst, hPrev, Cmd, nShow); \
}
```

You don't need to understand anything more about Windows programming: allegro takes care of setting it up for us!

Top-down design of Tetris:

Tetris01

```
main
  init
    init API
      Initialize Allegro library
      Initialize Allegro event handlers
      Open a window
      Install timer
    init game
  main_loop
    clear the back buffer
    handle keyboard events
    draw
  cleanup
    say goodbye
    close Allegro library
```

1. Create a **Win32 Application** Project named **Tetris01**
2. Project Settings -> Link, Category: Input, Object/library modules ... add "alld.lib" at the end of the list.
3. Copy main.h, main.cpp, init.h, init.cpp from Tetris00 to Teris01 and add them as files to the Tetris01 Project
4. Edit as follows:

```

void main_loop( )
{
    // clear the back buffer
    // handle keyboard events
    // draw
}

void init_allegro()
{
    // init allegro library
    // init handlers
        // init timers
        // init keyboard
    // open a window
    // install timer
}

void init_allegro()
{
    // init allegro library
    if( allegro_init() != 0 )
        cleanup( );

    // init handlers
    // init timers
    // init keyboard

    // open a window
    if (set_gfx_mode(GFX_AUTODETECT_WINDOWED, 400, 400, 0, 0) != 0)
    {
        // If that failed, output an error message and quit
        set_gfx_mode(GFX_TEXT, 0, 0, 0, 0);
        allegro_message("ERROR: Couldn't set the video mode.");
        cleanup( );
    }

    // install timer
}

void cleanup( )
{
    allegro_message( "Game over!" );

    // Gracefully exit from Allegro
    allegro_exit();

    exit(0);
}

```