PROGETTO 2023/2024

CONTATTI

- Non esitate a contattarci per dubbi, problemi o domande

- Scriveteci a:

Bianca Raimondi: bianca.raimondi3@unibo.it
https://www.unibo.it/sitoweb/bianca.raimondi3

Filippo Bartolucci: filippo.bartolucci3@unibo.it
https://www.unibo.it/sitoweb/filippo.bartolucci3

- Per il ricevimento mandate una mail

IL PROGETTO SERVE PER

- Farvi sperimentare le tecniche viste a lezione su scala più grande, in direzione di quello che succede nel mondo reale
- I progetti reali sono spesso MOLTO più grandi anche di questo progetto
- Mettervi alla prova in uno scenario che lascia spazio alla vostra fantasia, ma pone anche alcuni vincoli

VALUTAZIONE DEL PROGETTO

- Il progetto è parte integrante dell'esame ed è **obbligatorio**
- Ha un voto massimo di 8 punti che si sommano al voto dello scritto (max 24)
- I punti diventano al massimo 7 se si consegna il progetto dopo l'appello di Settembre 2024
- Il lavoro viene svolto in gruppi, ma la valutazione del progetto è **individuale**
    - Studenti dello stesso gruppo possono prendere voti diversi!
- Il progetto **si consegna una volta sola** ed il voto è valido per l'intero anno accademico (anche per quello successivo se le regole non cambiano)
- Chi non ottiene almeno 2 punti alla discussione orale dovrà presentare un nuovo progetto (che vi sarà dato in seguito)

GRUPPI

- Il progetto si svolge in gruppi di 2-4 persone, ma il voto **non** è di gruppo
- NON è possibile fare gruppi di 5 o più persone
- Sono ammessi **progetti individuali in casi eccezionali** (ma il carico di lavoro non diminuisce)
- Auto-organizzatevi per la creazione di gruppi
- Comunicherete i componenti del gruppo in fase di prenotazione e/o discussione del progetto

QUANDO CONSEGNARE IL PROGETTO?

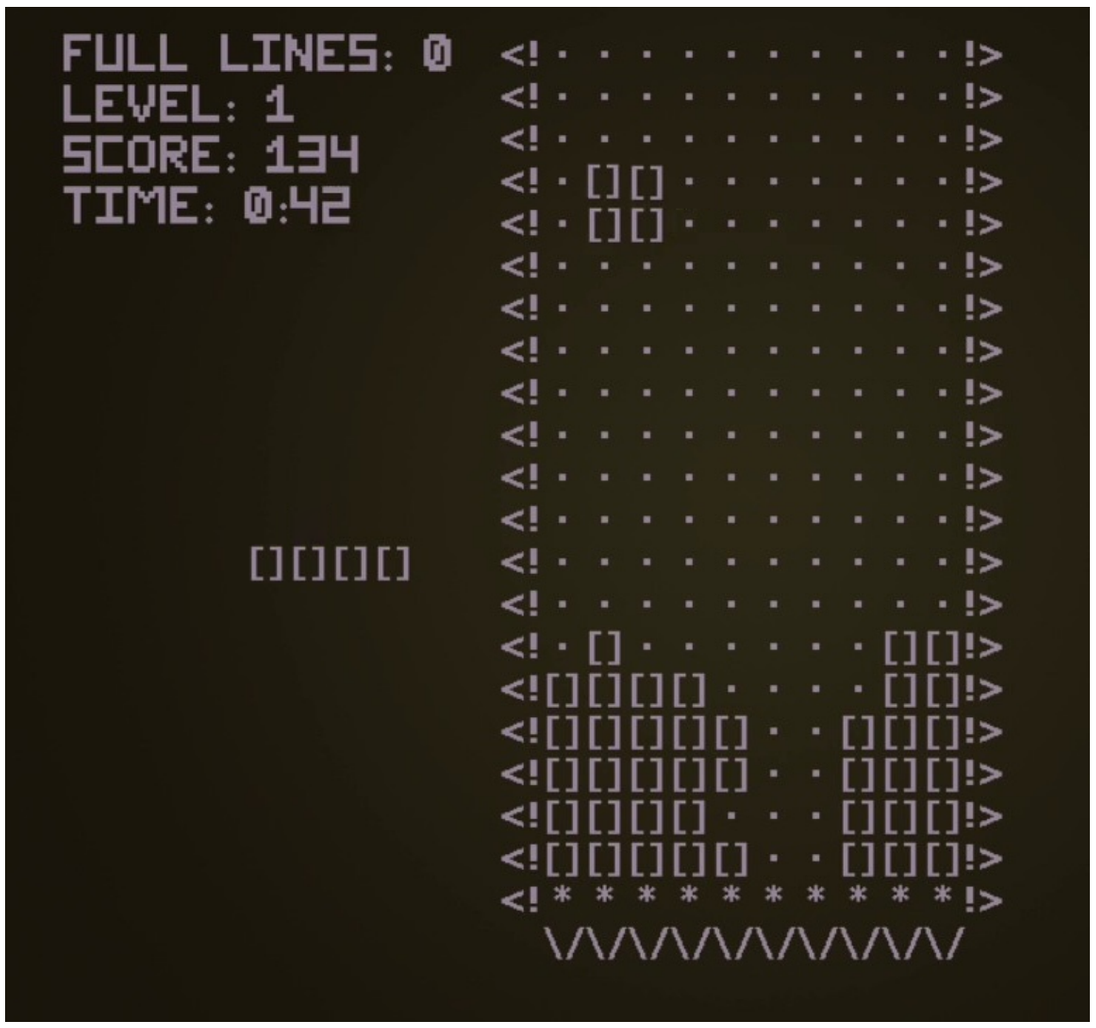
- Il progetto potrà essere presentato entro il 28/02/**2025**
- Ci saranno 5 appelli (uno/due per sessione)
- Le date degli appelli verranno comunicate tramite Virtuale

CONSEGNA E DISCUSSIONE

- Il progetto si consegna **una settimana** prima della discussione
- **TUTTI** i membri del gruppo devono partecipare alla discussione
- Il gruppo consegna:
  - Codice sorgente
  - File README
  - Screen recording che mostri l'esecuzione del gioco
  - Breve relazione (3/4 pagine) in cui si descrivono le principali scelte nell'implementazione del progetto
- Consegna da parte di **un solo membro** del gruppo tramite Virtuale
- Aggiungete alla consegna un commento con tutte le mail dei membri del gruppo

CONTENUTI DEL GIOCO

- Si richiede di implementare una versione ASCII semplificata di Tetris
- Con punteggio
- Il gioco si controlla da tastiera



REQUISITI E SVOLGIMENTO DEL GIOCO

1. Per il progetto **è possibile utilizzare SOLO le librerie grafiche curses/ncurses.h (e le librerie viste a lezione).**

2. La griglia deve essere realizzata con grafica ASCII e deve essere una matrice con dimensioni almeno 20x10.

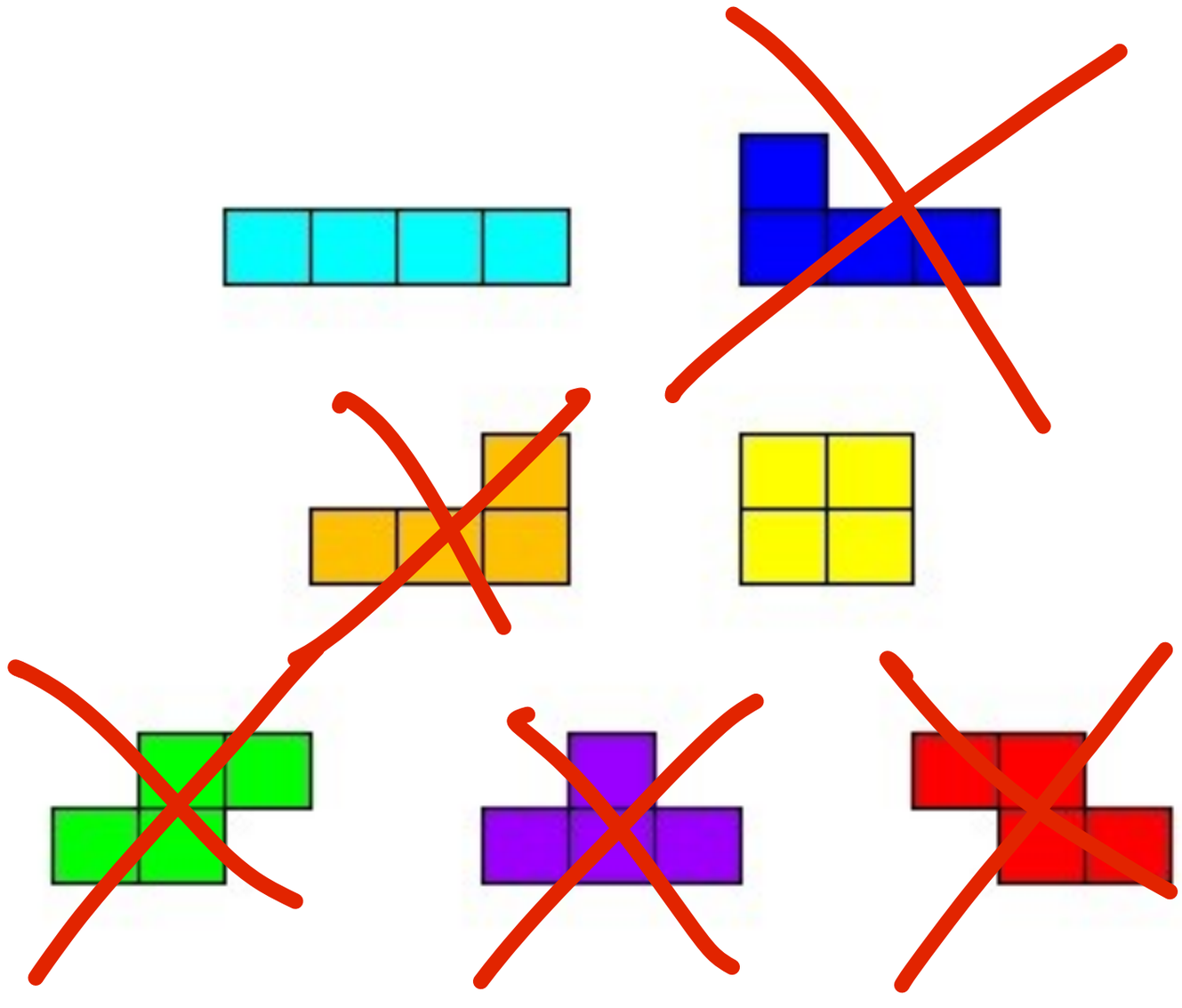
3. Le regole del gioco seguono quelle del Tetris classico: lo scopo è completare linee orizzontali nel campo di gioco. Quando una linea viene completata, scompare e il giocatore guadagna punti.

4. Non esistono traguardi.

5. GameOver: la partita termina quando un nuovo tetramino non può più essere inserito nel campo di gioco perché il nuovo pezzo ha toccato la parte superiore della griglia.

FORME DEI TETRAMINI

7. Si richiede di implementare **almeno** 2 tra tutte le forme di tetramini disponibili.



REQUISITI E SVOLGIMENTO DEL GIOCO

8. Caduta dei pezzi: i tetramini cadono dall'alto dello schermo verso il basso.
9. Ogni turno il nuovo pezzo (tetramino) da inserire è scelto casualmente tra l'insieme di forme predefinite.
10. Spostamento laterale: il giocatore può muovere il tetramino a sinistra o a destra finché non colpisce un altro pezzo o raggiunge il bordo del campo di gioco.
11. Rotazione: il giocatore può far ruotare il tetramino di 90° in senso orario o antiorario.
12. Punteggio: il punteggio aumenta ogni volta che il giocatore completa una linea. Completando più linee contemporaneamente si deve ottenere un punteggio maggiore.

REQUISITI E SVOLGIMENTO DEL GIOCO

8. Il gioco deve quindi prevedere un **menù** iniziale con le seguenti opzioni:

    a. Nuova partita

    b. Visualizza classifica ordinata in modo decrescente (punteggi delle partite terminate)

9. Al termine della partita (game over) il giocatore torna al menù iniziale.

IMPOSTAZIONE DEL PROGETTO

- Il progetto deve essere realizzato usando le classi (per esempio, la classe Tetramino è padre di tutti i pezzi)
- La classifica viene salvata in un file esterno
- Il progetto è organizzato in più file (no Main con tutto dentro)
- Ad ogni classe corrispondono due file:
  - *NomeClasse.cpp*
  - *NomeClasse.hpp*

ESEMPIO DIVISIONE DEL LAVORO ALL'INTERNO DI UN GRUPPO

- Componente #1 e #2: si occupano dei tetramini e del loro movimento
- Componente #3 e #4: si occupano della griglia di gioco e delle collisioni.
- Altri task da suddividere:
  - Punteggio
  - Gestione menù
  - Salvataggio file

ESEMPIO DI FILE

NomeClasse.hpp

```
class NomeClasse{
protected:
    int field;
    ...
public:
    ...
    void method();
    ...
};
```

NomeClasse.cpp

```
#include "NomeClasse.hpp"

void NomeClasse::method(){
    // do something
}
```

IN OGNI FILE IN CUI SI USA IL TIPO “NOMECLASSE” BISOGNA IMPORTARE NOMECLASSE.HPP

LETTURA DA FILE

- Per leggere e scrivere su file bisogna includere fstream
- ifstream è un **input file stream**: legge dati da un file

```
#include <iostream>
#include <fstream>
using namespace std;

int main() {
     ifstream inputFile; /* Dichiarazione di tipo */
     inputFile.open("file.txt");
     char ch;
     /* lettura dati */
     while(!inputFile.eof()){
          inputFile.get(ch);
          cout << ch << endl;
     }
     inputFile.close();
     return 0;
}
```

- ***while****(!inputFile.eof())* identifica un ciclo che finisce quando il file termina
- In questo contesto si tratta di un file testuale
- Tramite la funzione **get** si memorizza il carattere corrente del file nella variabile *ch*
- Alla fine del ciclo dobbiamo chiudere lo stream, ovvero il flusso di dati, proveniente dal file in ingresso, tramite l'istruzione ***inputFile.close()****;*

SCRITTURA SU FILE

- Per leggere e scrivere su file bisogna includere fstream
- ofstream è un **output file stream**: scrive dati su un file

```
#include <iostream>
#include <fstream>
using namespace std;

int main() {

    ofstream outputFile; /* Dichiarazione di tipo */
    outputFile.open("file2.txt"); /* Apertura del file */
    outputFile << "Prova di scrittura su file";
    outputFile.close();

    return 0;
}
```

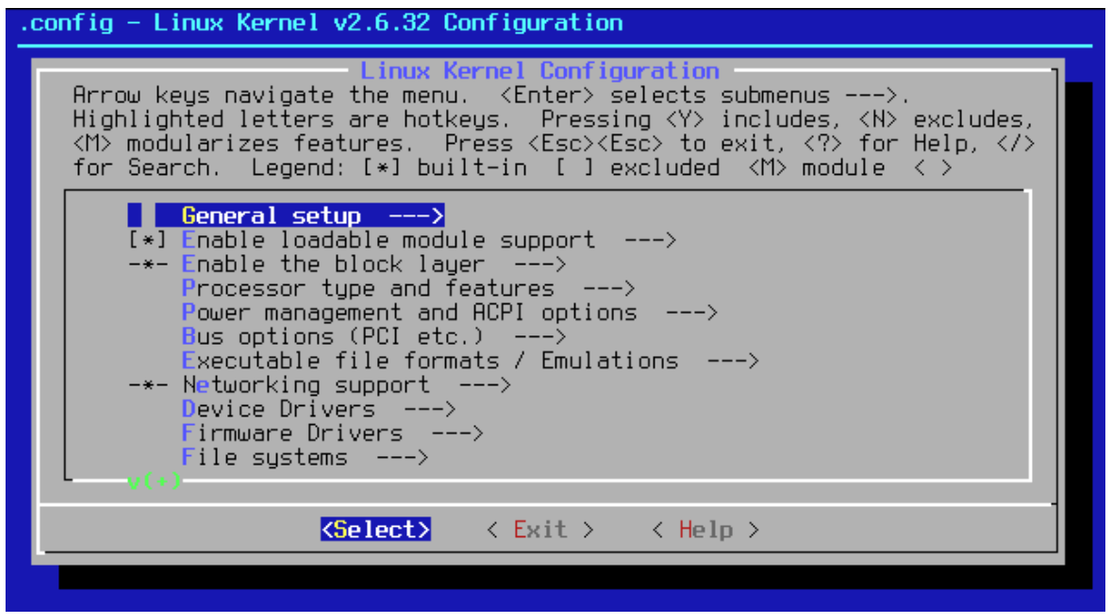
- Se il file da aprire in scrittura non è presente ne verrà creato uno vuoto con il nome specificato
- Possiamo stampare una frase su ***outputFile*** semplicemente ricalcando la sintassi del comando ***cout***
- Per scrivere sul file senza eliminarne il contenuto già presente, è sufficiente aprire il file con il comando:
  outputFile.open("file2.txt",ios::app);

LIBRERIA CURSES/NCURSES

- **ncurses** (new curses) è una libreria utilizzata per consentire al programmatore di scrivere TUI (Text-based User Interfaces).

- È un insieme di strumenti per lo sviluppo di applicazioni "GUI-like" che vengono eseguite tramite terminale.

- ncurses è un'emulazione di software libero delle **curses**
  - Su windows ncurses si chiama curses, fate attenzione in fase di import (*#include <curses.h>*)



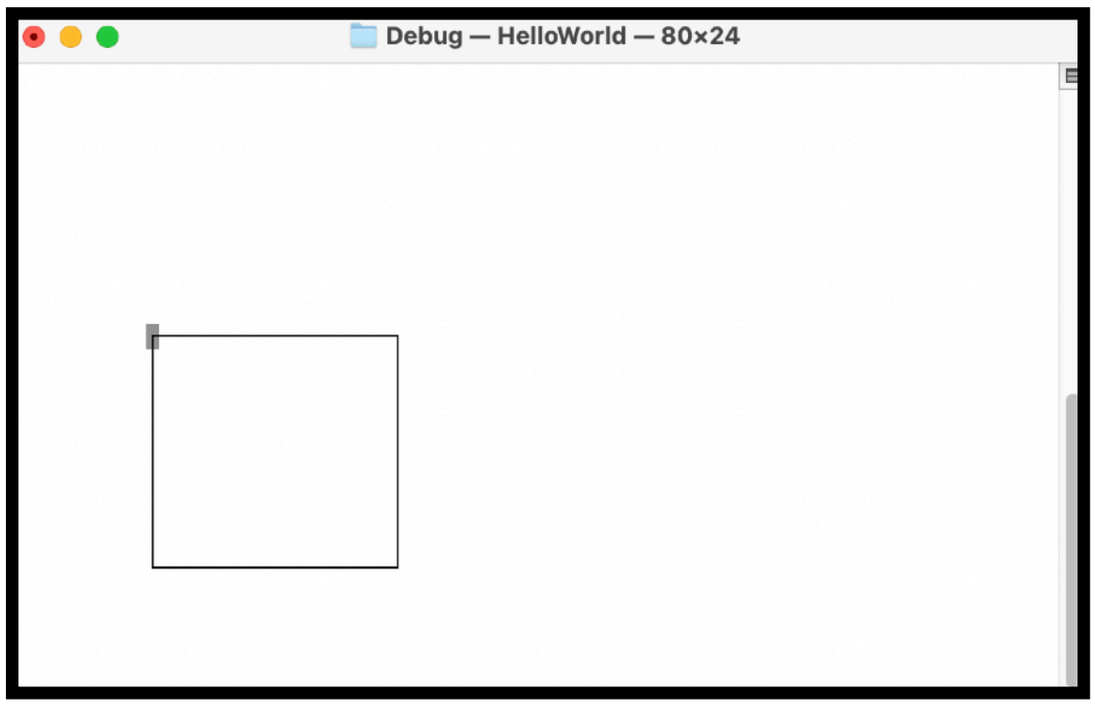
Esempio di applicazione creata con curses

LIBRERIA CURSES/NCURSES

- Esempio dell'utilizzo della libreria

HelloWorld.cpp

```cpp
#include <ncurses.h>
#include <iostream>

using namespace std;

int main(int argc, char ** argv){
    //initializes the screen
    initscr();

    int height, width, start_y, start_x;

    height = 10;
    width = 20;
    start_y = 10;
    start_x = 10;

    WINDOW * win = newwin(height, width, start_y, start_x);

    //refreshes the screen
    refresh();

    //show the window
    box(win, 0, 0);
    wrefresh(win);

    //wait for user input like "cin >> a;"
    getch();

    //deallocates memory and ends ncurses
    endwin();
}
```



- Qui potete trovare una guida per l'utilizzo della libreria: https://www.youtube.com/playlist?list=PL2U2TQ__OrQ8jTf0_noNKtHMuYlyxQl4v

LIBRERIA CURSES/NCURSES

Funzioni principali:

- **initscr()** : inizializza screen
- **endwin()** : dealloca memoria e chiude il programma
- **newwin(height, width, start_y, start_x)** : crea una nuova finestra
- **refresh()** : refresh dello screen con le informazioni aggiuntive
- **move(x, y)** : muove il cursore nella posizione indicata (x, y)
- **clear()** : cancella tutto ciò che è presente sullo screen

LIBRERIA CURSES/NCURSES – LINUX

- Installazione libreria:

sudo apt-get install libncurses5-dev libncursesw5-dev

- Compilazione:

g++ HelloWorld.cpp -lncurses -o HelloWorld

- Esecuzione:

./HelloWorld

LIBRERIA CURSES/NCURSES – MACOS

- Installazione libreria:

brew install ncurses

- Compilazione:

g++ HelloWorld.cpp -lncurses -o HelloWorld

- Esecuzione:

./HelloWorld

LIBRERIA CURSES/NCURSES – WINDOWS

- Installate una suite di tool di compilazione per Windows, ad esempio MinGW
- Scaricare il file **MinGW-w64-install.exe** da https://sourceforge.net/projects/mingw/
- Fate partire l'installer e andate avanti
- Selezionare le opzioni "A Basic MinGW installation" e "The GNU C++ compiler"
- Selezionare "Apply Changes" dal menu Installation
- Aggiungete al PATH di Windows il percorso: C:\MinGW\bin\

LIBRERIA CURSES/NCURSES – WINDOWS

- Da MinGW selezionare i seguenti pacchetti:

pdcurses

libncurses

libpdcurses

ncurses

- Compilazione:

```
g++ -I/mingw64/include/ncurses -o HelloWorld HelloWorld.cpp -lncurses -L/mingw64/bin -static
```

- Esecuzione:

HelloWorld

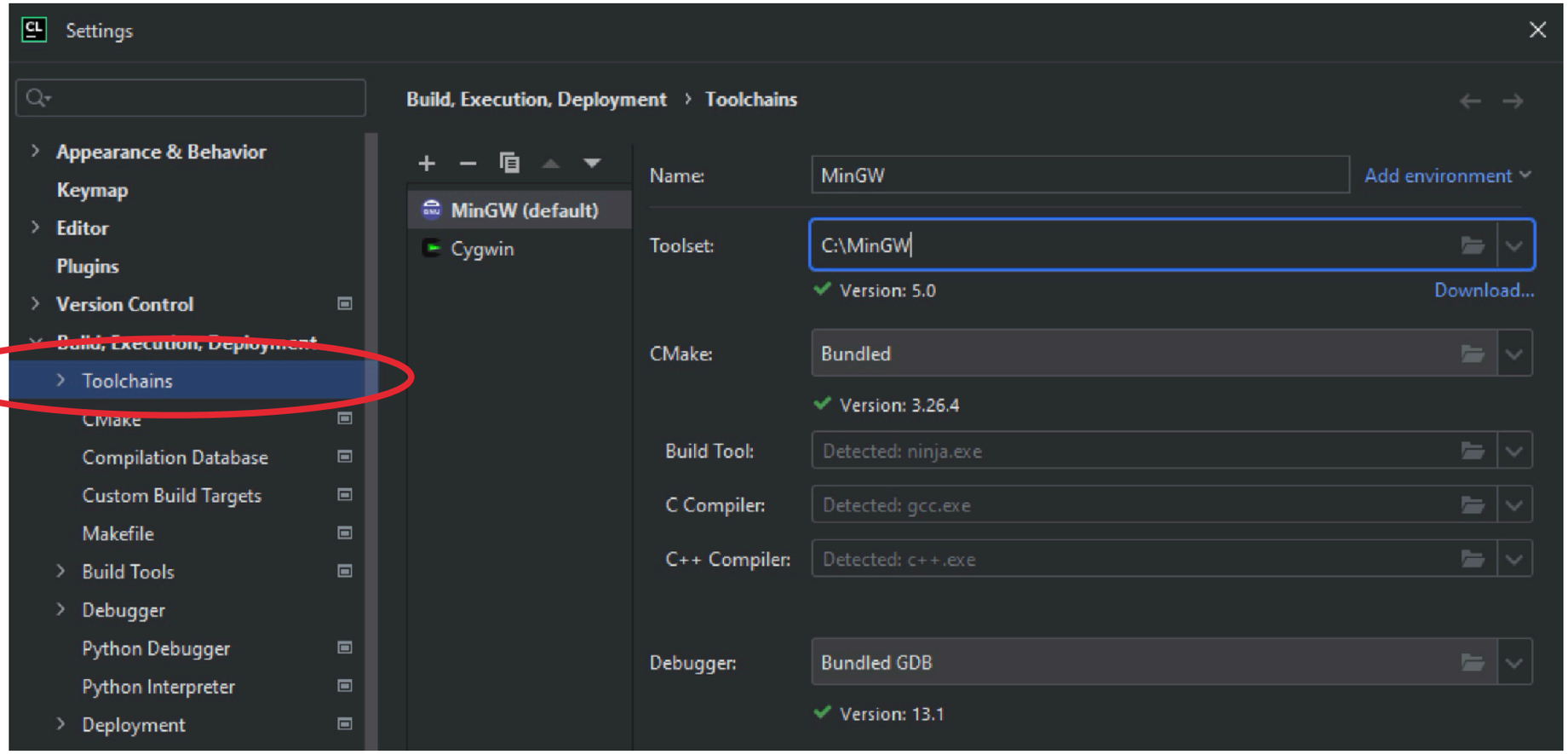
oppure

./HelloWorld

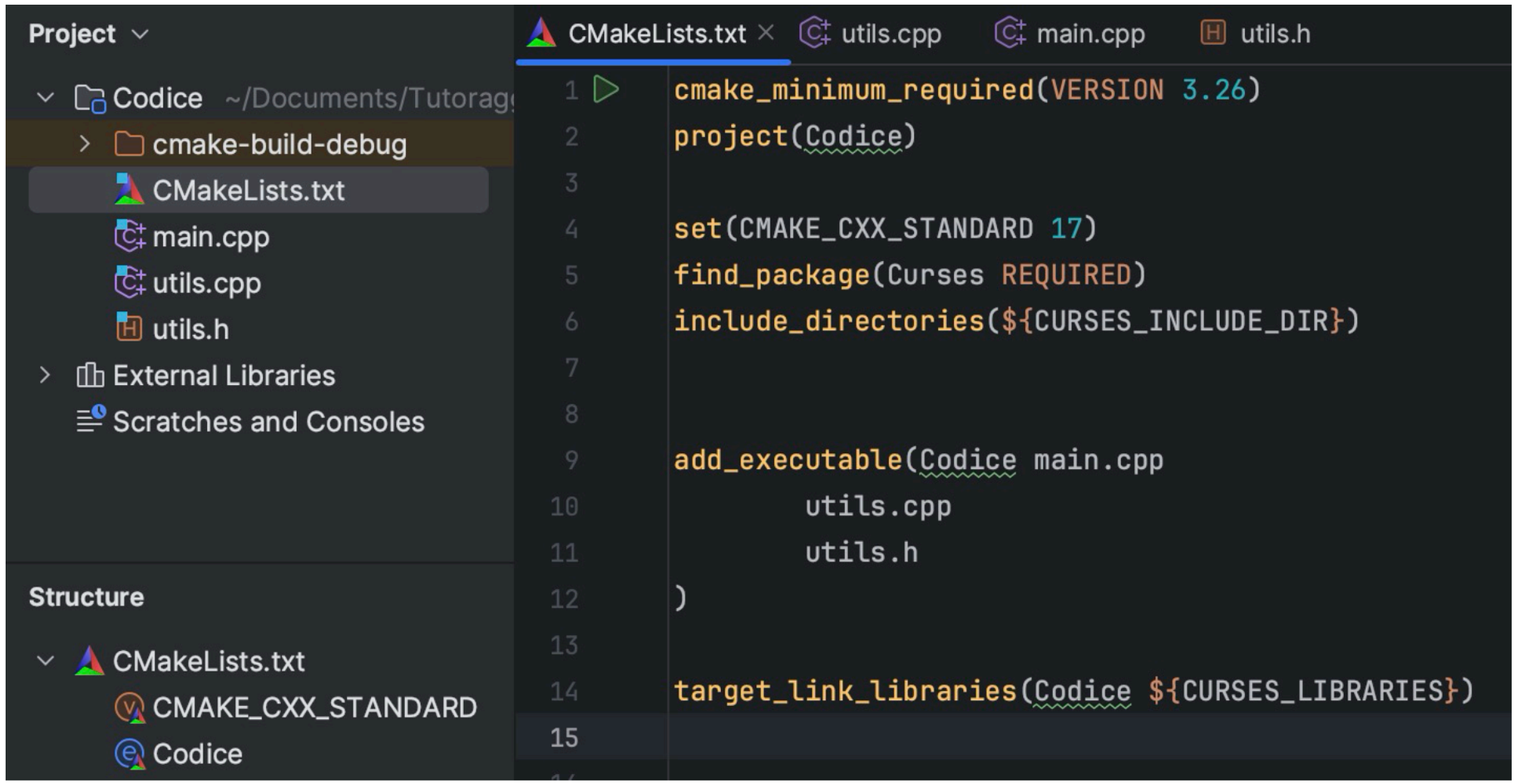
Nel caso non funzionassero i comandi è possibile eseguire il programma facendo doppio click sul file

LIBRERIA CURSES/NCURSES – PER WINDOWS

- Assicuratevi di aver selezionato il toolchain giusto per la compilazione
- Questa opzione in Clion si trova in
  - File -> Settings -> Build, Execution, Deployment -> Toolchains
  - Selezionare MinGW (default) e assicurarsi di non avere "Bundled MinGW" nel campo "Toolset"
  - In caso, dovrete modificarlo andando a scegliere la vostra installazione di MinGW (il percorso di default è C:\MinGW)



LIBRERIA CURSES/NCURSES – COMPILAZIONE CON CLION



- Aprire il file CMakeList.txt e aggiungere le seguenti linee evidenziate in grassetto:

find_package(Curses REQUIRED)
include_directories(${CURSES_INCLUDE_DIR})
add_executable(PROJECT_NAME main.cpp)
target_link_libraries(PROJECT_NAME ${CURSES_LIBRARIES})

- NB: PROJECT_NAME deve essere sostituito con il nome del vostro progetto CLION.
- Clion creerà l'eseguibile **all'interno della cartella cmake-build-debug**
- Eseguite poi il programma **utilizzando il terminale**