

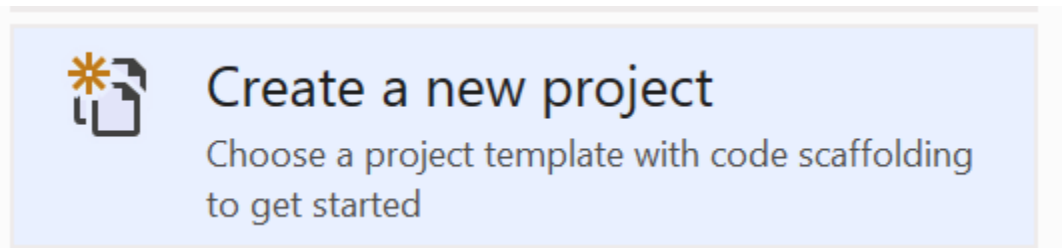
## Setting up the project

1. Download the project from the capstone website.

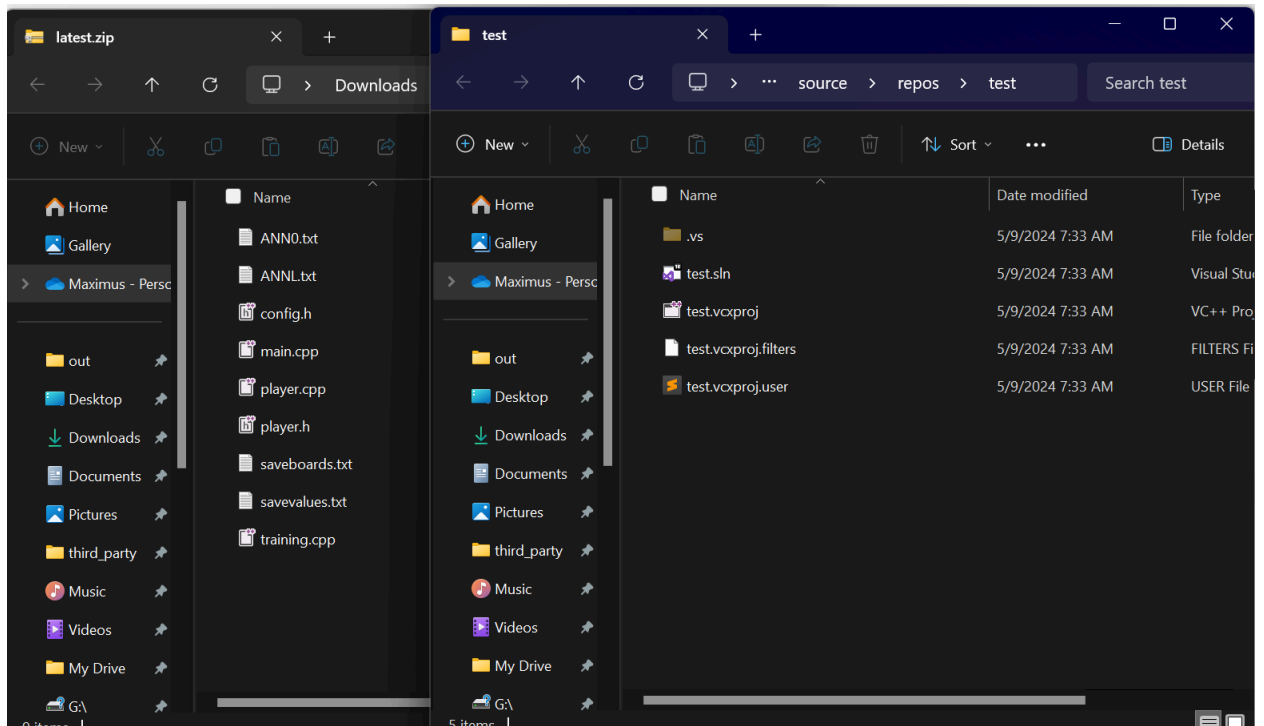
[Download Latest Zip](#)

2. Create a new visual studio 2022 project

*Note, the project uses <windows.h> to visualize the game, making the program only compatible with windows platform.*



3. Unzip the downloaded zip and transfer all files to the visual studio project directory.



#### 4. Add the files into the project via the solution explorer.

##### In Header Files

Add config.h

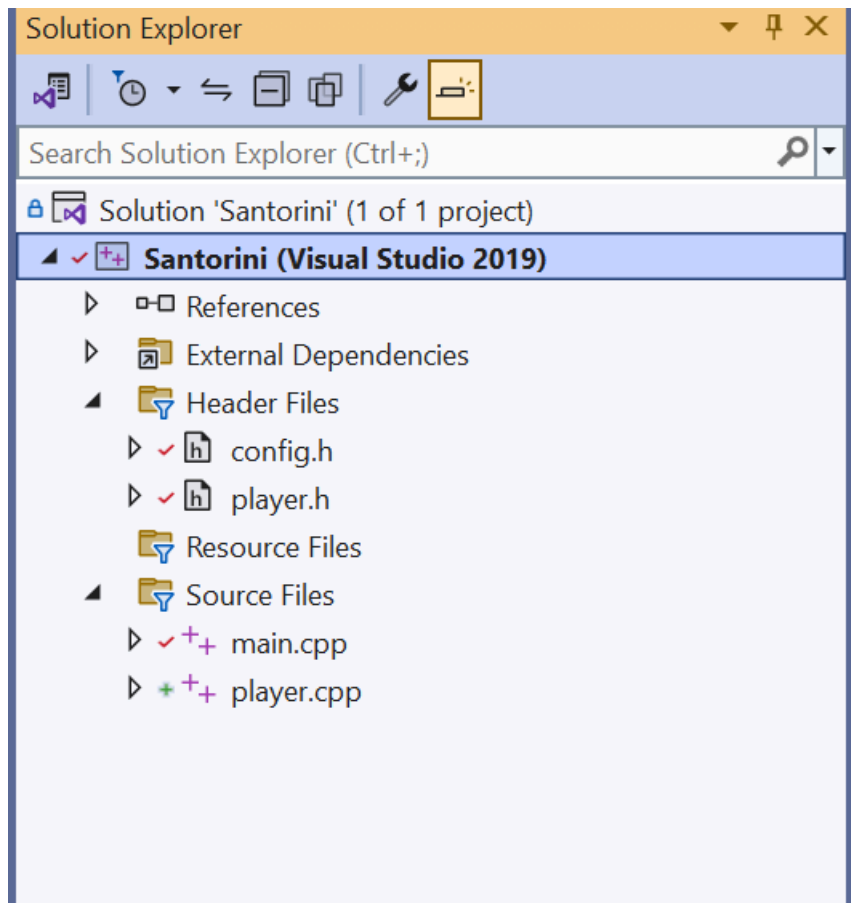
Add player.h

##### In Source Files

Add player.cpp

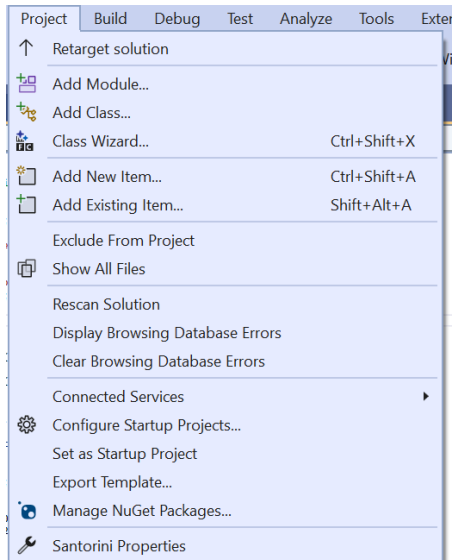
Add main.cpp **OR** training.cpp

**NOTE:** A program can only support one main() function. Both main.cpp and training.cpp feature a main function. The project will not run if both are added simultaneously. The differences between the two will be described below.

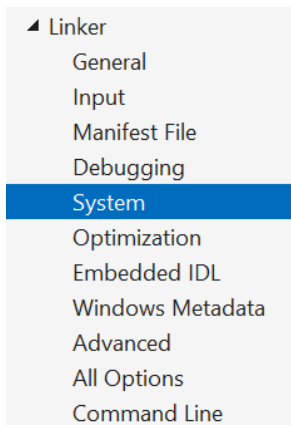


## 5. Allocating more stack memory for training

Select the project tab, then Santorini Properties.



Navigate to the Linker tab, then System.



Finally, locate “stack reserve size” and set it equal to 6291456

SubSystem	Console (/SUBSYSTEM:CONSOLE)
Minimum Required Version	
Heap Reserve Size	
Heap Commit Size	
Stack Reserve Size	<b>6291456</b>
Stack Commit Size	
Enable Large Addresses	
Terminal Server	
Swap Run From CD	No
Swap Run From Network	No
Driver	Not Set

## **Using Main.cpp**

Main.cpp is used for a user interacting with the main game interface. Each execution of this program will run one game. The player types can be chosen by the player.

### **Altering the ANN used.**

By default, selecting ANN will upload a specific version of the ANN player. The available versions are specific in the project downloaded file as textfiles labeled "ANN###.txt"

To alter the ANN version, locate the global variables ANN1File and ANN2File.

```
1  const string ANN1File = "ANNL.txt";  
2  const string ANN2File = ANN1File;
```

Changing the string value to the name of an alternative ANN file will upload a different version of the ANN.

## Using training.cpp

Training.cpp is used for training the ANN.

The variables start and end in main() specify which segments of training you wish to run. By default, they are set equal to each other (and only executing step specified in “start”)

```
int main() {  
    //step to start on  
    char start = 'A';  
  
    //step to terminate on (if range is true)  
    char end = 'C';  
  
    //whether or not to do multiple steps  
    end = start;  
}
```

The segments of execution are as follows:

- A. Plays 10 games between heuristic v. heuristic players and stores all boardstates encountered (non-repeating) in saveboards.txt.
- B. Plays 1000 games between heuristic v. heuristic players from each boardstate in saveboards.txt. The measured winrate of each boardstate is stored in savevalues.txt.
- C. Calculates gradient across the ANN (specified by ANNFile variable), referencing boardstates in saveboards.txt and respective values in savevalues.txt. Outputs a new ANN in newANN.txt
- D. Tests the outputs across two different ANNs for all boardstates found in saveboards.txt.