

Guess the Number, C++ Tutorial

Part I - Introduction

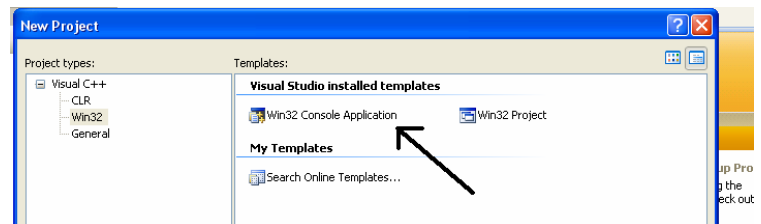
Here is a hello from Kevin at Making Indie Games. Here is a fun little Guess the Number game for those of you who want to make something useful. If you get lost or confused you can find the all the code on the last page. Well let's begin.

Part II - Creating a New Project

I am using Visual C++ 2008 Express Edition so the steps might be a little different depending on your IDE or Integrated Development Environment just to be technical.

The first thing you want to do is create a new project. Select Win32 Console Application and name the project "Guess the Number." If a template window comes up select finish.

If a precompiled header didn't show up use the following code so we are all on the same page.



```
#include "stdafx.h"
```

```
int _tmain(int argc, _TCHAR* argv[])  
{  
    return 0;  
}
```

Part III – Creating a Splash Screen and Rules

Let's make a little splash screen to tell the world who made this beautifully crafted program. I encourage you to add your name along with some funny jokes.

Step 1 is to make a new function that will contain our splash screen code. First we must declare the function so add the following code under `#include "stdafx.h"`.

```
void SplashScreen();
```

Void tells the computer that we will not return a variable. Next let's tell our main function to run the function. Insert the following code inside the main function but above return 0;

```
SplashScreen();
```

Your main function should look like this:

```
int _tmain(int argc, _TCHAR* argv[])
{
    SplashScreen(); //Splash and Rules

    return 0;
}
```

The green text is just a comment and isn't executed by the compiler. Before we make our SplashScreen function we need to include the standard C++ Library, iostream, and ctime because we will be using cout and cin along with random numbers. Under the line which reads `#include "stdafx.h"` insert the following code.

```
#include <iostream>
#include <cstdlib>
#include <ctime>
using namespace std;
```

Now we can make our function. At the end of our code file insert the following code.

```
void SplashScreen()
{
    cout << "-----" << "\n";
    cout << "    www.makingindiegames.com    " << "\n";
    cout << "-----" << "\n" << "\n";
    cout << "The rules of the game are simple. Keep " << "\n";
    cout << "guessing numbers until you are right. " << "\n";
}
```

cout << outputs what ever is between the “”. “\n” makes a new line. The ; just tells the computer the line has ended. Next on the list is getting a random number. Next on the list is making our game loop. Below is a diagram of what it will look like.

Part IV – Planning the Game Loop.

GameLoop()

1) Pick a Number

New Loop

- 2) Guess a Number
- 3) Check the Number

4) Ask for a New Game

Now that we have the loop planed out let's make a function called GameLoop(). Let's first declare it under `void SplashScreen();`

```
void GameLoop();
```

Now let's tell the computer to run the game loop. Change your main function to look like the one below.

```
int _tmain(int argc, _TCHAR* argv[])
{
    SplashScreen(); //Splash and Rules
    GameLoop();
    return 0;
}
```

Add the following function to the botom of your code file.

```
void GameLoop()
{
    int r; //local variable, the random number
    int guess; //players guess
    int quit; //quit game yes or no

    //Pick a Number
    r = (rand() % 100 + 1); // + 1 makes it so 0 never comes up.
}
```

The `int r;` are local variables because they are defined in a function. We will use them as we develop the GameLoop(). `r = (rand() % 100 + 1);` is a random number 1-100. Next we need to include the Guess Number Loop.

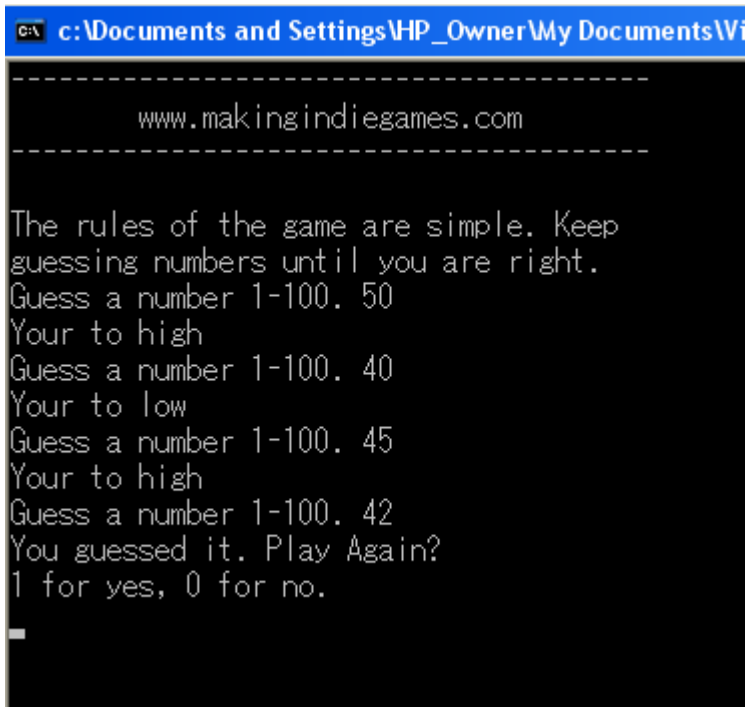
Add the following code to the bottom of the game loop.

```
guess = 0; //make sure guess != r
while(guess != r)
{
    cout << "Guess a number 1-100. ";
    cin >> guess; //sets input to guess
    if(guess > r)
    {
        cout << "Your to high" "\n";
    }
    if(guess < r)
    {
        cout << "Your to low" "\n";
    }
}
cout << "You guessed it. Play Again?" "\n";
cout << "1 for yes, 0 for no. " "\n";
cin >> quit;
if(quit == 1)
{
    GameLoop();
}
}
```

Run the game and have fun. I recommend moding this game. Instead of numbers 1-100 make it numbers 1-1000 or a million. You could also make a count of each guess. If you want to take a look at the entir code file it is on the last page.

Part V – Closing

I would just like to note that the first time I ran the program the number was 42, the answer to life. How weird? I then realized that the first one always comes out to 42. Well I really can't explain it so I am in need of your help. If you can solve this problem please email me at kevinoflaherty@makingindiegames.com



```
c:\Documents and Settings\HP_Owner\My Documents\Wi
-----
www.makingindiegames.com
-----
The rules of the game are simple. Keep
guessing numbers until you are right.
Guess a number 1-100. 50
Your to high
Guess a number 1-100. 40
Your to low
Guess a number 1-100. 45
Your to high
Guess a number 1-100. 42
You guessed it. Play Again?
1 for yes, 0 for no.
_
```

Part 6 – The Code

```
#include "stdafx.h"
#include <iostream>
#include <cstdlib>
#include <ctime>
using namespace std;

void SplashScreen();
void GameLoop();

int _tmain(int argc, _TCHAR* argv[])
{
    SplashScreen(); //Splash and Rules
    GameLoop();
    return 0;
}
```

```
void SplashScreen()
{
    cout << "-----" << "\n";
    cout << "          www.makingindiegames.com          " << "\n";
    cout << "-----" << "\n" << "\n";
    cout << "The rules of the game are simple. Keep " << "\n";
    cout << "guessing numbers until you are right. " << "\n";
}

void GameLoop()
{
    int r; //local variable, the random number
    int guess; //players guess
    int quit; //quit game yes or no

    //Pick a Number
    r = (rand() % 100 + 1); // + 1 makes it so 0 never comes up.

    guess = 0; //make sure guess != r
    while(guess != r)
    {
        cout << "Guess a number 1-100. ";
        cin >> guess; //sets input to guess
        if(guess > r)
        {
            cout << "Your to high" << "\n";
        }
        if(guess < r)
        {
            cout << "Your to low" << "\n";
        }
    }
    cout << "You guessed it. Play Again?" << "\n";
    cout << "1 for yes, 0 for no. " << "\n";
    cin >> quit;
    if(quit == 1)
    {
        GameLoop();
    }
}
```