

Forum:[Hack Forums](#)

Thread: [HiLo Game Tweaking](#)

ViciouS - June 23, 2011, 5:42 pm

Alright, so I am a beginner "coder" and I was messing around with a game called "HiLo" and I finally got it to display different results depending on how many guesses you attempt. However, I'm not sure how to implement my next step. What I want to do is tell the user a certain message and end the game if they reach too many guesses. Something like "You've used 20 guesses already. Forget it. Game Over." What line should I use or how should I put it in?

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

```
int main()
{
    srand(time(0));
    int guessNum = rand() % 100;
    int guess;
    int attempts = 0;

    cout << "This is the \"Hi Lo Game\"\n" << endl;
    cout << "Guess a number for 1 - 99: " << endl;
    while (guess != guessNum)
    {
        cin >> guess;
        if (guess > guessNum)
        {
            cout << "Sorry, that guess is too high. Guess again.\n" << endl;
        }
        if (guess < guessNum)
        {
            cout << "Sorry, that guess is too low. Guess again. \n" << endl;
        }
        attempts++;
    }

    if (guess == guessNum)
    {
        cout << "Game Over. That was the right guess. The correct number was " << guessNum <<
"." << endl;
    }

    if (attempts == 1)
    {
        cout << "Incredible!! You guessed the correct number in a single guess!\n" << endl;
    }
}
```

```

    if (attempts > 1 && attempts <= 5)
    {
        cout << "Well done. You only needed " << attempts << " guesses to get the right answer.\n" <<
endl;
    }

    if (attempts > 5 && attempts <= 10)
    {
        cout << "Meh. You could have done better. You needed " << attempts << " guesses to win.\n"
<< endl;
    }

    if (attempts > 10)
    {
        cout << "Terrible. Keep playing to improve. It took " << attempts << " attempts to get the
answer.\n" << endl;
    }

    system("pause");
    return 0;
}

```

Also, I have heard bad things about the system pause but I was taught that way. I replace with "cin.get()" or something like that, right?

Irritable - June 23, 2011, 11:35 pm

(Jun 23, 2011 05:42 PM)ViciouS Wrote: [\[CITE\]](#) Alright, so I am a beginner "coder" and I was messing around with a game called "HiLo" and I finally got it to display different results depending on how many guesses you attempt. However, I'm not sure how to implement my next step. What I want to do is tell the user a certain message and end the game if they reach too many guesses. Something like "You've used 20 guesses already. Forget it. Game Over." What line should I use or how should I put it in?

EDIT: Code removed.

Also, I have heard bad things about the system pause but I was taught that way. I replace with "cin.get()" or something like that, right?

Hello, ViciouS.

For your first question, regarding exiting prematurely based on a number of guesses, how would we accomplish this?

Because the word "guesses" means there is more than one guess, I am assuming we are using a loop.

This means we have to count (keep track of) the number of guesses the user has made.

Every iteration we can increment (add to) this count by one; To represent another guess.

When the number of guesses has exceeded a certain amount, we exit the loop!

Example:

Code #include <iostream>

#include <time.h>

using namespace std;

int main() {

```
srand(time(NULL));
```

```
cout << "Hello there! You have 5 guesses to pick a number I am thinking of between 1 and 20!\n";
```

```
int guess = 0;  
int number = rand() % 20;
```

```
cout << "ANSWER: " << number << endl;
```

```
cout << "You have: 5 guesses remaining.\n";
```

```
for (int count = 4; count >= 0; count--) {  
    cin >> guess;
```

```
    if (guess == number) {  
        break;  
    }
```

```
    cout << "You have: " << count << " guesses remaining.\n";  
}
```

```
    if (guess == number) {  
        cout << "You've won!";  
    } else {  
        cout << "You've lost... :(";  
    }
```

```
    return 0;  
}
```

I hope this leads to a solution for your problem! ?

Yes, you can use `cin.get()` to 'hold the execution window open.'

You can also use an IDE for that, or simply open it with command prompt.