

COMP1917: 05 Random Numbers

Sim Mautner

s.mautner@unsw.edu.au

August 2, 2016

Resources

- Solutions to exercises in these slides:

www.cse.unsw.edu.au/~simm/lectures/random-numbers/

Random Numbers

Uses (mostly game related):

- Rolling dice in games.
- Deciding how much damage your spell does.
- Deciding whether or not a Zubat appears based on some probability.

Random Numbers

How random numbers work:

- 1 We start with a number. This number is called a **seed**.
- 2 We use this number to calculate a new number.
- 3 That new number is used to calculate the next number.
- 4 And so on...

Keywords to Remember:

```
#include <stdlib.h>
```

```
srandom( /* seed goes here */ );
```

```
int num = rand();
```

```
// To produce a number between 0 and n-1 inclusive.
```

```
int num = rand() % n;
```

Seeds

- Starting with the same seed will produce the same sequence of random numbers.
- Ex 1: Write an application which produces 10 random numbers.
 - ▶ This set of random numbers should be the same sequence each time the application is run.
 - ▶ Each number should be between 0 and 10 inclusive.

Different Sequences

- To produce a difference sequence every time the code is run, we use the current time as the seed.

```
#include <time.h>
```

```
srandom(time(NULL));
```

- Ex 2: Write an application which produces 10 random numbers.
 - ▶ This set of random numbers should be different every time the application is run.
 - ▶ Each number should be between 0 and 10 inclusive.

Dice Rolls and Coin Flips

- Ex 3: Write an application which simulates rolling a die 6 times and prints the output.
- Ex 4: Write an application which simulates flipping a coin 10 times and prints the output.
- Ex 5: Write an application which simulates rolling 2 dice 10 times and prints the sum of the two dice each time. (Hint: This is not the same as generating random numbers between 2 and 12 inclusive.)

Guess My Number

- Ex 6: Write an application which takes in two numbers, a minimum and a maximum value. It then generates a random number between the two given numbers. The user enters a guess and is told whether the generated number is *higher* or *lower* than their guess. They repeat this process until they guess the generated number.