Be a Computer Scientist

Game Programming

1 PROGRAMMING VIDEO GAMES

1.1 *Big Budget Games*

- Typically use a complicated game engine
- Examples include Unreal Engine (Epic Games), Frostbite (EA), Anvil (Ubisoft)
- I'll be showing a little about Unreal Engine
- Made by many programmers, graphic designers, 3d modelers

1.2 *Our Game!*

SPACE RACER

1.3 *Our Schedule for the Week*

1. Learn basic programming
2. Explore 3D modeling
3. Learn about how the different parts of games fit together
4. Learn basic AI
5. Hold a competition!

2 BASIC PROGRAMMING

2.1 *What is programming?*

1. A computer follows instructions one by one
2. A *program* is an instruction set
3. A program might take in *input* from a user
4. A program often returns *output* the user can see

2.2 *Machine Code*

- The computer reads *machine code*
- Machine code is hard to write and hard to read
- Sometimes we use it for performance
- For the most part, we run away from it because the ”easy” version of it looks like...

```
org
  xor ax, ax
  mov ds, ax
  mov si, msg
boot_loop:lodsb
  or al, al
  jz go_flag
  mov ah, 0x0E
  int 0x10
  jmp boot_loop
go_flag:
  jmp go_flag
msg   db "hello world", 13, 10, 0
  times 510-($-$$) db 0
  db 0x55
  db 0xAA
```

2.3 *Higher Level Languages*

- A higher level language lets you write code that gets converted to something the machine understands
- The language may also check for errors for you
  - Errors at *compile time*
  - Errors at *runtime*
- Easier to understand, the one we'll be using looks like...

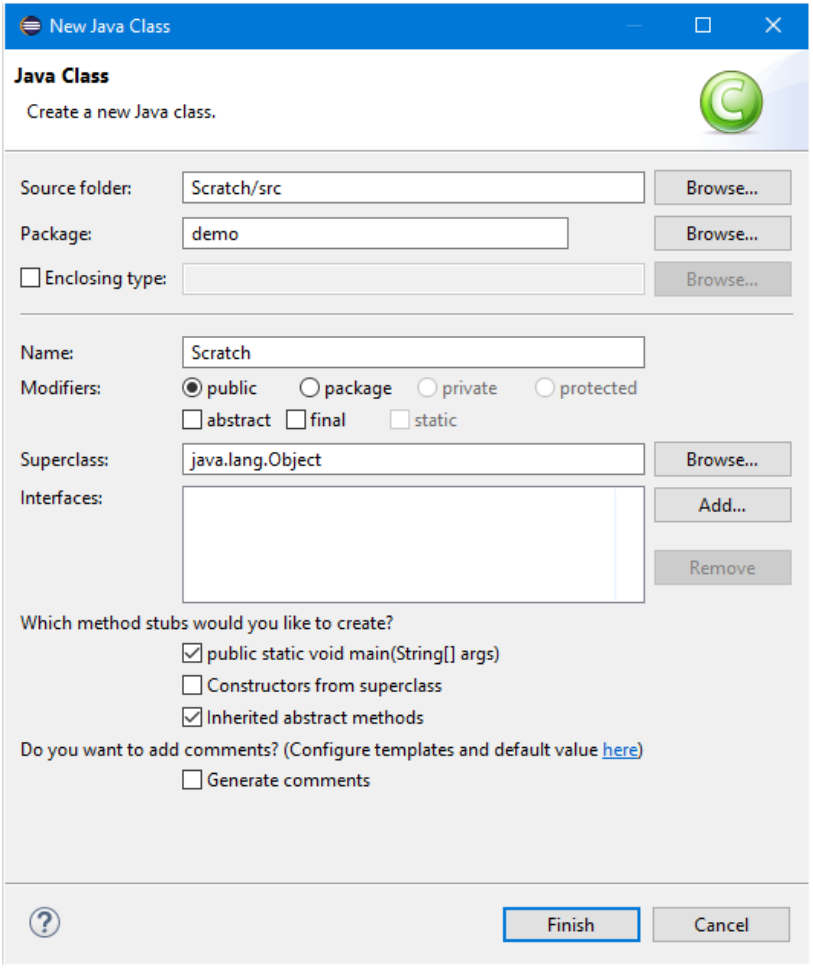
```java
public class HelloWorld {

    public static void main(String[] args) {
        // Prints "Hello, World"
        // to the terminal window.
        System.out.println("Hello, World");
    }

}
```

2.4 *Java*

- High level programming language
- Good choice if working on simple *cross platform* programs
- Is *object oriented*
- Can work from the *command line*, but we're going to use an *integrated developer environment*

2.5 *Making a scratch project*

1. Open Eclipse (circular icon left side of screen)
2. Select ”New Project...”
3. Make New Java Project
4. Then we’re going to make a new class


New Java Class
Java Class
Create a new Java class.
Source folder:
Scratch/src
Browse...
Package:
demo
Browse...
Enclosing type:
Browse...
Name:
Scratch
Modifiers:
public
package
private
protected
abstract
final
static
Superclass:
java.lang.Object
Browse...
Interfaces:
Add...
Remove
Which method stubs would you like to create?
public static void main(String[] args)
Constructors from superclass
Inherited abstract methods
Do you want to add comments? (Configure templates and default value here)
Generate comments
Finish
Cancel

2.6 *Variables and Assignment*

```
int five; // tells the computer to make space
          // for a number we called number

five = 5; // now the value of the "five" is 0
five = 6; // now it's 6....
five = 5; // let's change it back to 5
```

2.7 *Types of Variables*

```
int aNumber = 54920;
double aDecimalNumber = 43902.3290;
String aSentence = "Hello world!";
char ACharacter = 'c';
```

2.8 *Programming Together*

- We're going to all program together now
- Programming examples will be provided for the lab section