

Exam Covers

- o Weeks 1-4
 - o C# differences
 - o Abstract classes
 - o Inheritance
 - o Interfaces
 - o Exceptions
 - o Basic File I/O

Project 2 is up

- o It includes a sample answer to project #1 – please look at it

File I/O

- o Games have been moving more towards "data driven" designs. File I/O becomes very important in these content driven games.

Files Store/Load

- o Graphical Models
- o Maps
- o User preferences
- o AI
- o Mods
- o Textures

Files

- o Used to store data in secondary storage
- o We will only look at *sequential* file access.
- o System.IO

Streams

- o Definition: a communication channel between the program and another object
- o Streams can be used to:
 - o Connect a program with the terminal for input/output
 - o Connect a program with a file
 - o Connect with temporary storage in memory
 - o Connect to another program over the network

Input and Output Streams

- o An input stream delivers information from the source (for example, a file) to the program.
- o An output stream delivers information from the program to a destination (for example, a file).
- o "Standard" streams:
 - o Input
 - o Output
 - o Error

File I/O

- o How would you figure out how to read from and write to files?

Steps for file I/O

- o Open/Create the file (returns a file handle)
- o Attach the handle to a reader/writer object
- o Read/Write to the file in order
- o Close the file

Different Types of File I/O

- o BinaryReader
- o BinaryWriter
- o Stream
 - o BufferedStream
 - o FileStream
 - o MemoryStream
- o TextReader
 - o StreamReader
 - o StringReader
- o TextWriter
 - o StreamWriter
 - o StringWriter

Do and Discuss

- o Please copy the following class and run it
- o What are the potential issues with this class?

Exception Handling

- o There can and will be exceptions
 - o What exceptions can be thrown?
 - o try getting some of them to be thrown
 - o using
 - o Finalizers and Dispose

using

```
using ( StreamReader sr = new StreamReader("ganked.txt") ){  
    ...  
}  
  
is pretty much equivalent to:  
  
try {  
    StreamReader sr = new StreamReader("ganked.txt");  
    ...  
} finally {  
    if (sr != null)  
        sr.Dispose();  
}
```

Destructors and Dispose

- o Destructor: A method called Finalize that is called when your object is garbage collected
- o Dispose: implement the IDisposable interface and use a public method called Dispose
 - o Normally, GC.SuppressFinalize(this) so that garbage coll. calls your dispose

Binary Streams

- o Binary is often more efficient than text
 - o BinaryReader
 - o BinaryWriter