**Test 1 – Fixing Right Arrow Crash**

To fix this bug, I had to add a condition to both the Key Press checks and the DataFile::GetRecord() function. This check was to make sure the passed integer value was not lower than 0 or greater than DataFile::GetRecordCount() – 1. If these conditions were met, then just set the current viewing record to 0.

Graphical user interface, text

Description automatically generated

A picture containing text, screenshot, monitor, screen

Description automatically generated

This was eventually changed to instead use the recordCount variable, since it holds how many potential records we could have.

**Test 2 – Changing how we read**

Originally, I made a vector which holds all the Texture2D objects. At the start of the program, it loads all of them into the vector.

Text

Description automatically generated

Text

Description automatically generated

This didn’t work because of the GetRecordCount()-1. Removing -1 solved this.

Text

Description automatically generated

After realizing that the customer only wants to have one customer loaded at a time, I decided to scrap the idea of texture caching, as it would still lead to the same problem as before. If storing Record pointers was bad according to the customer, then certainly they would not like texture storage in memory.

I began changing a lot of functions that accessed the Record vector, and instead made it reference a single Record variable.

This allowed me to set up the bare bones of how the new system would work.

Of course, I was not able to build the project for a while since there was a lot to be changed.

**Test 3 – Fixing File Read**

A screenshot of a computer

Description automatically generated with medium confidence

I reworked the load function to a function that simply reads the record count and the filename, and stores these for use later.

Text

Description automatically generated

Since the load function is going to work a lot differently now, I will need to adapt the old load code with the new variables I introduced.

A picture containing text, indoor, monitor, screenshot

Description automatically generated

A picture containing text, indoor, screenshot, electronics

Description automatically generated

Here the GetRecord function was reworked to search for the given index in the binary file. It would then set the Record to whatever it finds.

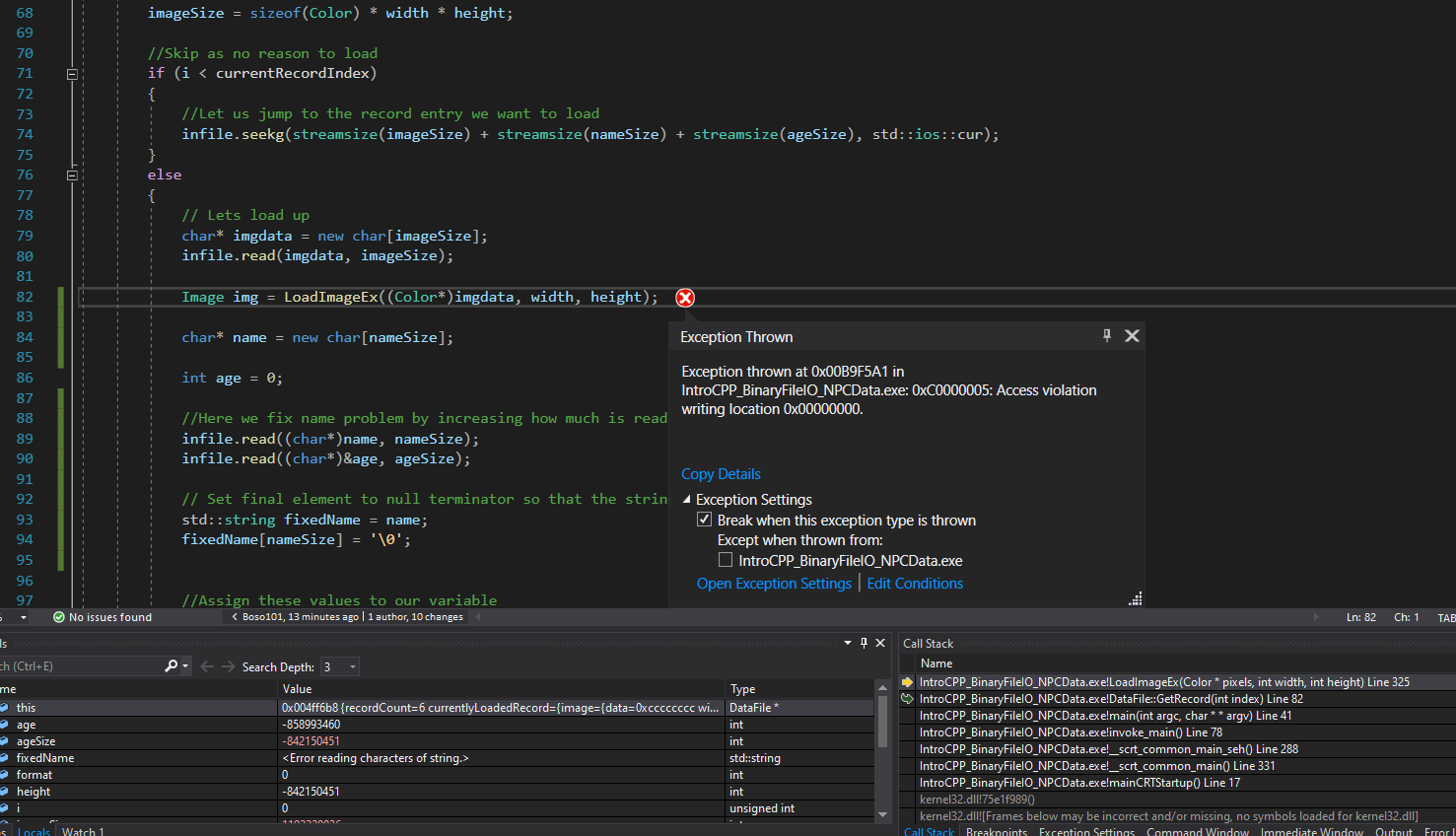
**Test 4 –** Fixing Add Player

Text

Description automatically generated

Add player was extremely simple to fix. Instead of add player creating a new Record, we instead just pass this new record to the Save function, which then appends this new record to the file.

The hard part was remaking the Save function so that it can handle appending files.



At first, this resulting in a bunch of errors, such as the Record not being written in the correct location.

A picture containing text, screenshot, indoor, computer

Description automatically generated

To test if it worked, I simply called it and passed in custom parameters, which did indeed change my file to include the new record.

A picture containing text, monitor, screenshot, screen

Description automatically generated

**Test 5 –** Fixing Name Reading

Graphical user interface, application

Description automatically generated

Even though we can load records according to our customers specification, we still have a bug with how our names are read.

Text

Description automatically generated

After debugging, we can see that the nameSize is 3 for “Tom”, yet we get a lot of garbage after the third element.

This is because index 0 = ‘T’, index 1 = ‘o’ and index 2 = ‘m’.

Due to the fact that more than this is being read, we can just set index 3 to something that stops the rest of the garbage from being read.

To solve this problem, I simply set the char arrays last value to ‘/0’, which is a null terminator.

Text

Description automatically generated with medium confidence

Doing this allowed the names to print perfectly.

A picture containing icon

Description automatically generated