Terrain2STL Review

Thatcher Chamberlin

February 24, 2021

Code Reference

Code referenced in these slides is from

- Terrain2STL 1a, commit b2731563

Small Code Block

```
void startSTLfile(FILE * file, int numTriangles){
  rewind(file);
  //write the 80 byte STL header (can be whatever)
  for(int i = 0; i < 80; i++){
    fwrite("t",1,1,file);
  }
  //write the number of triangles (4 bytes)
  fwrite((uint32_t *)&numTriangles,4,1,file);
}
```

Small Code Block - Correct Line Numbers

```
61 void startSTLfile(FILE * file, int numTriangles){
62   rewind(file);
63   //write the 80 byte STL header (can be whatever)
64   for(int i = 0; i < 80; i++){
65     fwrite("t",1,1,file);
66   }
67   //write the number of triangles (4 bytes)
68   fwrite((uint32_t *)&numTriangles,4,1,file);
69 }
```

Small Code Block - Language Label

Here I've told Beamer to present these lines as C code

```c
void startSTLfile(FILE * file, int numTriangles){
  rewind(file);
  //write the 80 byte STL header (can be whatever)
  for(int i = 0; i < 80; i++){
    fwrite("t",1,1,file);
  }
  //write the number of triangles (4 bytes)
  fwrite((uint32_t *)&numTriangles,4,1,file);
}
```

Small Code Block - `linerange` Test

Here I'm taking lines 61-63 and 67-69 - note how the line numbers don't reflect the discontinuity! Not ideal for my purposes

```
61 void startSTLfile(FILE * file, int numTriangles){
62   rewind(file);
63   //write the 80 byte STL header (can be whatever)
64   //write the number of triangles (4 bytes)
65   fwrite((uint32_t *)&numTriangles,4,1,file);
66 }
```

Large Code Block

On this slide I've tried to include too many lines...

```c
#include <stdio.h>
#include <stdint.h>
#include "STLWriter.h"


int voidCutoff = 0;
char endTag[2] = {0,0};

//Determines the normal vector of a triangle from three vertices
vect3 normalOf(vect3 p1, vect3 p2, vect3 p3){
  vect3 u = {0,0,0};
  vect3 v = {0,0,0};
  vect3 r = {0,0,0};
  u.x = p2.x-p1.x;
  u.y = p2.y-p1.y;
  u.z = p2.z-p1.z;
  v.x = p3.x-p1.x;
  v.y = p3.y-p1.y;
  v.z = p3.z-p1.z;
  r.x = u.y*v.z-u.z*v.y;
  r.y = u.z*v.x-u.x*v.z;
  r.z = u.x*v.y-u.y*v.x;
  return r;
}

//Creates a triangle and its normal vector from three vertices
triangle createTriangle(vect3 j, vect3 k, vect3 l){
  triangle t;
  t.a = j;
  t.b = k;
  t.c = l;
  t.normal = normalOf(j,k,l);
  return t;
```

Large Code Block - Overflow Into Next Slides I

... but here I've made them overflow onto subsequenct slides!

```c
#include <stdio.h>
#include <stdint.h>
#include "STLWriter.h"


int voidCutoff = 0;
char endTag[2] = {0,0};

//Determines the normal vector of a triangle from three vertices
vect3 normalOf(vect3 p1, vect3 p2, vect3 p3){
  vect3 u = {0,0,0};
  vect3 v = {0,0,0};
  vect3 r = {0,0,0};
  u.x = p2.x-p1.x;
  u.y = p2.y-p1.y;
  u.z = p2.z-p1.z;
  v.x = p3.x-p1.x;
  v.y = p3.y-p1.y;
  v.z = p3.z-p1.z;
  r.x = u.y*v.z-u.z*v.y;
  r.y = u.z*v.x-u.x*v.z;
  r.z = u.x*v.y-u.y*v.x;
  return r;
}

//Creates a triangle and its normal vector from three vertices
triangle createTriangle(vect3 j, vect3 k, vect3 l){
  triangle t;
  t.a = j;
  t.b = k;
```

Large Code Block - Overflow Into Next Slides II

```
  t.c = l;
  t.normal = normalOf(j,k,l);
  return t;
}

//Writes a triangle into the STL file
void addTriangle(FILE * file, triangle t){
  //normal vector1
  fwrite(&t.normal.x, sizeof(float), 1, file);
  fwrite(&t.normal.y, sizeof(float), 1, file);
  fwrite(&t.normal.z, sizeof(float), 1, file);

  //vertices
  fwrite(&t.a.x, sizeof(float), 9, file);
  /* vvv replace by ^^^
  fwrite(&t.a.x, sizeof(float), 1, file);
  fwrite(&t.a.y, sizeof(float), 1, file);
  fwrite(&t.a.z, sizeof(float), 1, file);

  fwrite(&t.b.x, sizeof(float), 1, file);
  fwrite(&t.b.y, sizeof(float), 1, file);
  fwrite(&t.b.z, sizeof(float), 1, file);

  fwrite(&t.c.x, sizeof(float), 1, file);
  fwrite(&t.c.y, sizeof(float), 1, file);
  fwrite(&t.c.z, sizeof(float), 1, file);
  */
  fwrite(endTag, 1, 2, file);
}

void startSTLfile(FILE * file, int numTriangles){
  rewind(file);
```

Large Code Block - Overflow Into Next Slides III

```
  //write the 80 byte STL header (can be whatever)
  for(int i = 0; i < 80; i++){
    fwrite("t",1,1,file);
  }
  //write the number of triangles (4 bytes)
  fwrite((uint32_t *)&numTriangles,4,1,file);
}
```

Added Source Name

Finally I defined a macro to encapsulate this code excerpt pattern and add a title containing the source file, so my final \excerpt macro produces results like:

../Terrain2STL/src/STLWriter.c:

```c
void startSTLfile(FILE * file, int numTriangles){
  rewind(file);
  //write the 80 byte STL header (can be whatever)
  for(int i = 0; i < 80; i++){
    fwrite("t",1,1,file);
  }
  //write the number of triangles (4 bytes)
  fwrite((uint32_t *)&numTriangles,4,1,file);
}
```