

Reverse engineer .555 files

Import files

```
In[1]:= FileNames[]
Out[1]:= {Temple_nile.sg3, Temple_ptah.555, Temple_ptah.sg3, Temple_ra.555, Temple_ra.sg3,
          Temple_seth.555, Temple_seth.sg3, Title.555, title555.nb, title555.pdf}

In[2]:= names = FileNames["*.555"]
Out[2]:= {Temple_ptah.555, Temple_ra.555, Temple_seth.555, Title.555}

In[3]:= titles = Import["*.555", "UnsignedInteger16"];
In[4]:= Length/@titles
Out[4]:= {609 366, 610 723, 632 878, 307 200}

In[5]:= FactorInteger /@ Length /@ titles
Out[5]:= {{{2, 1}, {3, 1}, {101 561, 1}}, {{401, 1}, {1523, 1}}, {{2, 1}, {316 439, 1}}, {{2, 12}, {3, 1}, {5, 2}}}
```

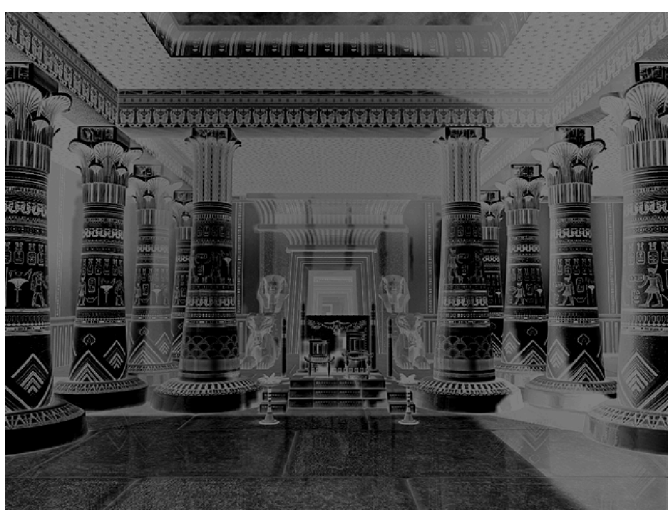
Fold file into lines

```
In[6]:= lines = Partition[titles[[-1]], 640];
```

Plot as 16-bit grayscale

```
In[7]:= ArrayPlot[lines]
```

Out[7]=



Function to extract 5-bit RGB color from 16-bit integer

```
In[8]:= rgb555 = Function[uint, RGBColor[
  (BitAnd[BitShiftRight[uint, #1[[1]], 2^#1[[2]] - 1])
  / (2^#1[[2]] - 1) & /@ {{10, 5}, {5, 5}, {0, 5}}]]]

Out[8]= Function[uint,
  RGBColor[(BitAnd[BitShiftRight[uint, #1[[1]], 2^#1[[2]] - 1])
  / (2^#1[[2]] - 1) & /@ {{10, 5}, {5, 5}, {0, 5}}]]]
```

Map RGB extract function to file

```
In[9]:= coltitle = Map[rgb555, titles[[-1]]];
```

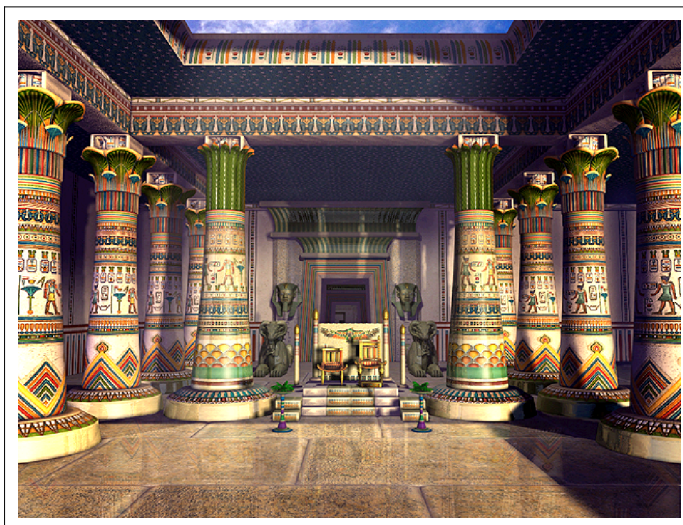
Fold file into lines

```
In[10]:= titlearray = Partition[coltitle, 640];
In[11]:= Dimensions[titlearray]
Out[11]= {480, 640}
```

Display as RGB graphics file

```
In[12]:= ArrayPlot[titlearray]
```

Out[12]=



References

Maps are Folds in Functional Programming. Here's how:

<https://medium.com/@anirudheka/maps-are-folds-in-functional-programming-heres-how-979b90e-b657a>