# Obfuscated_code_or_piece_of_art

May 4, 2017

## 1 Table of Contents



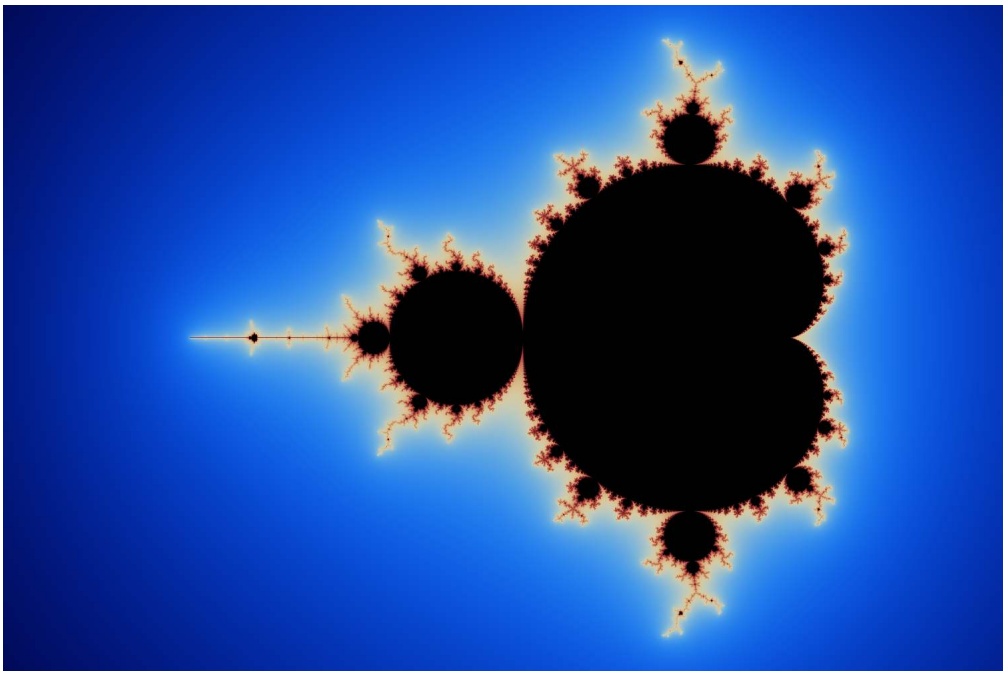
## 2 Obfuscated code or piece of art?

This short notebooks shows a few examples of Python code, designed to draw something, and shaped as what they will draw...

### 2.1 Mandelbrot set

This nice little code will write a visualization of the Mandelbrot set, on the domain $[-3,3] \times [-3i,3i]$, for $1500 \times 1500$ points, as a Bitmap (written manually in binary).

```
In [2]: %%time
        b_                                        =   (
                                                  255,
                                                lambda
                                          V       ,B,c
                                       :c   and Y(V*V+B,B,  c
                                         -1)if(abs(V)<6)else
                          (              2+c-4*abs(V)**-0.4)/i
                            )  ;v,      x=1500,1000;C=range(v*x
                             );import  struct;P=struct.pack;M,\
                      j  ='<QIIHHHH',open('art/M.bmp','wb').write
        for X in j('BM'+P(M,v*x*3+26,26,12,v,x,1,24))or C:
                      i  ,Y=_;j(P('BBB',*(lambda T:(T*80+T**9
                            *i-950*T  **99,T*70-880*T**18+701*
                           T  **9     ,T*i**(1-T**45*2)))(sum(
                          [             Y(0,(A%3/3.+X%v+(X/v+
                                         A/3/3.-x/2)/1j)*2.5
                                       /x   -2.7,i)**2 for  \
                                         A       in C
```



```
                                                    [:9]])
                                                      /9)
                                                     )   )
```

```
/usr/local/lib/python2.7/dist-packages/ipykernel/__main__.py:20: DeprecationWarning: integer ar
```

```
CPU times: user 3min 43s, sys: 296 ms, total: 3min 44s
Wall time: 3min 43s
```

---

## 2.2 Penrose patterns

This second nice little code will write a visualization of a Penrose tiling (infinite pattern) to a PNG image, of resolution $2000 \times 2000$.

```
In [3]: %%time
        _                                       =\
                                              """if!
                                            1:"e,V=200
                                          0,(0j-1)**-.2;
                                        v,S=.5/  V.real,
                                      [(0,0,4      *e,4*e*
                                    V)];w=1          -v"def!
```

```
                          E(T,A,              B,C):P
                       ,Q,R=B*w+               A*v,B*w+C
                  *v,A*w+B*v;retur             n[(1,Q,C,A),(1,P
        ,Q,B),(0,Q,P,A)]*T+[(0,C              ,R,B),(1,R,C,A)]*(1-T)"f
    or!i!in!_[:11]:S       =sum([E           (*x)for       !x!in!S],[])"imp
      ort!cair               o!as!O;      s=O.Ima               geSurfac
       e(1,e,e)               ;c=O.Con  text(s);               M,L,G=c.
         move_to                ,c.line_to,c.s                et_sour
           ce_rgb                a"def!z(f,a)                :f(-a.
            imag,a.       real-e-e)"for!T,A,B,C!in[i       !for!i!
              in!S!if!i["""];exec(reduce(lambda x,i:x.replace(chr
               (i),"\n "[34-i:]),   range(   35),_+"""0]]:z(M,A
                 );z(L,B);z         (L,C);         c.close_pa
                 th()"G             (.4,.3             ,1);c.
                 paint(             );G(.7             ,.7,1)
                 ;c.fil             l()"fo             r!i!in
                 !range             (9):"!             g=1-i/
                 8;d=i/          4*g;G(d,d,d,          1-g*.8
                 )"!def     !y(f,a):z(f,a+(1+2j)*(     1j**(i
                 /2.))*g)"!for!T,A,B,C!in!S:y(M,C);y(L,A);y(M
                 ,A);y(L,B)"!c.st            roke()"s.write_t
                 o_png('art/                    penrose.png')
                 """                                    ))
```

```
CPU times: user 4.17 s, sys: 20 ms, total: 4.19 s
Wall time: 4.18 s
```

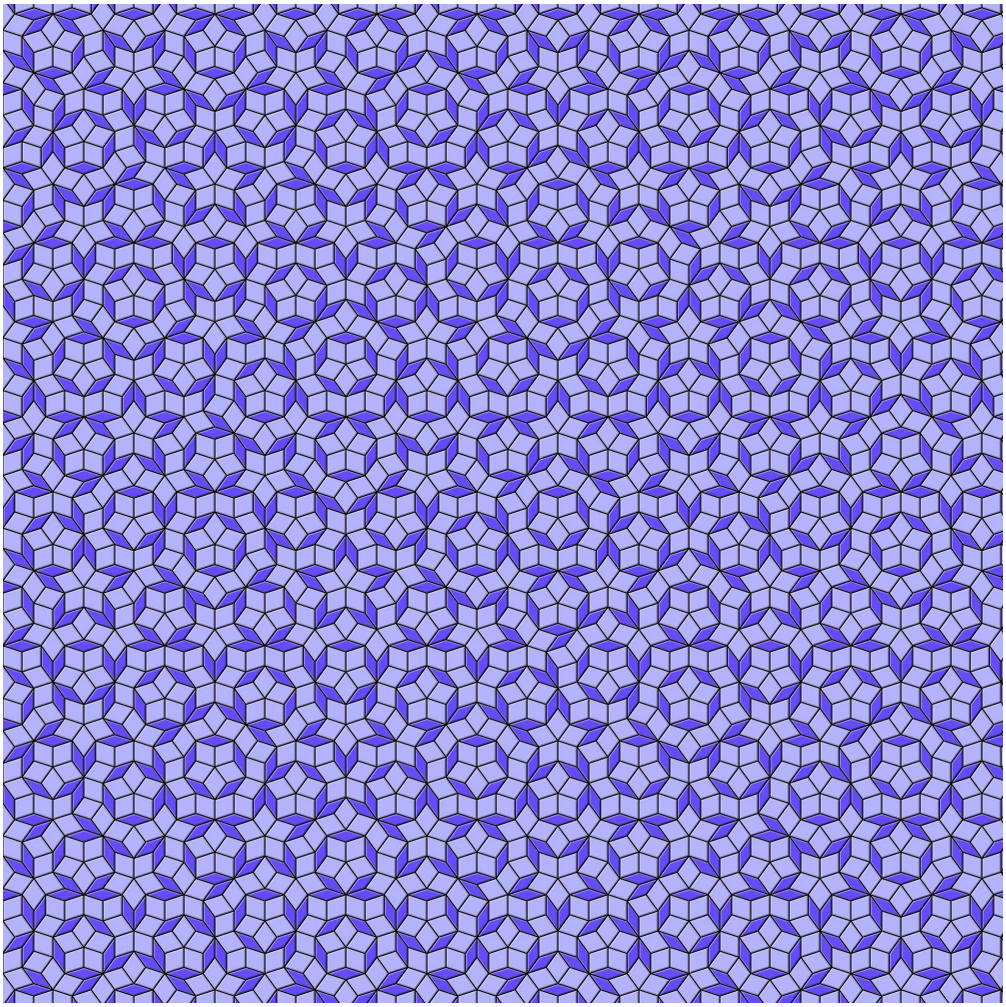
---

### 2.3 Bitcoin address & private key generator

This is the most concise (and the most sexy!) implementation of the Bitcoin protocol to generate a new address and private key!

In [26]: %%time

```
_                    =r"""A(W/2,*M(3*G
                  *G*V(2*J%P),G,J,G)+((M((J-T
                )*V((G-S)%P),S,T,G)if(S@(G,J))if(
              W%2@(S,T)))if(W@(S,T);H=2**256;import&h
            ashlib&as&h,os,re,bi    nascii&as&k;J$:int(
         k.b2a_hex(W),16);C$:C    (W/    58)+[W%58]if(W@
        [];X=h.new("rip           em    d160");Y$:h.sha25
       6(W).digest();I$                 d=32:I(W/256,d-1)+
      chr(W%256)if(d>0@"";                U$:J(k.a2b_base
     64(W));f=J(os.urando       m(64))       %(H-U("AUVRIxl
    Qt1/EQC2hcy/JvsA="))+      1;M$Q,R,G       :((W*W-Q-G)%P,
    (W*(G+2*Q-W*W)-R)%P)       ;P=H-2**       32-977;V$Q=P,L=
```



```
1,O=0:V(Q%W,W,O-Q/W*                      L,L)if(W@O%P;S,
T=A(f,U("eb5mfvncu6                     xVoGKVzocLBwKb/Nst
zijZWfKBWxb4F5g="),      U("SDra         dyajxGVdpPv8DhEI
qP0XtEimhVQZnEfQj/       sQ1Lg="),       0,0);F$:"1"+F(W
 [1:])if(W[:1           ]=="\0"@""       .join(map(B,C(
  J(W))));K$:               F(W          +Y(Y(W))[:4]);
   X.update(Y("\4"+                      I(S)+I(T)));B$
    :re.sub("[OOIl    _]|            [^\\w]","","".jo
     in(map(chr,ra    nge    (123))))[W];print"Addre
       ss:",K("\0"+X.dig    est())+"\nPrivkey:",K(
         "\x80"+I(f))""";exec(reduce(lambda W,X:
            W.replace(*X),zip(" \n&$@",["","",
               " ","=lambda W,")else "])
                    ,"A$G,J,S,T:"+_))
```

```
Address: 1PMc1Xt6AMaBhDkX4De6nEPMA8XtDDLJhL
Privkey: 5Jrof4v1UX3X7Zzj724rr6sviw6RhpsczqZEcV8a3MJw1u5eBEb
CPU times: user 64 ms, sys: 24 ms, total: 88 ms
Wall time: 72.9 ms
```

---

Disclaimer: I am *not* the author of these small examples!

That's it for today!