名稱：冷爆的月亮伊布(Frozen Eevee)
在新竹的月亮伊布覺得冷於是在原地三段式抖動
然後因為劇烈抖動而產生巨大能量而發光!!
匯聚能量壓縮之後變成煙火發射似的注上飛
最後的碎片是冷爆的月亮伊布煙火✲✲✲

換掉原本伊布的皮，
改用hw3的Phong shader
看起來亮亮的比較生動!!

```
153  void display() {
154      //Clear the buffer
155      glClearColor(0.0f, 0.0f, 0.0f, 0.0f);
156      glClear(GL_COLOR_BUFFER_BIT);
157      glClearDepth(1.0f);
158      glEnable(GL_DEPTH_TEST);
159      glDepthFunc(GL_LEQUAL);
160      glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
161
162      Timer(0);
163
164      if (flag == -1)   DrawUmbreon(Explosionprogram);
165      else{
166          if (flag == 0) DrawUmbreon(Normalprogram);
167          DrawUmbreon(Phongprogram);
168      }
169      glutSwapBuffers();
170  }
```

Set Timer :
用於Normal sparkling和Explosion的時間參數

```
59  static void Timer(int value) {
60      if (angle >= 6) {// 原地抖動
61          angle = 0.0;
62          counter++;
63          if (counter == 2500)   flag = 1;// 往上飛
64      }
65
66      if (flag == 0) {//三段式抖動
67          if (counter <= 1000) angle += 0.1;
68          else if (counter > 1000 && counter <= 3000) angle += 1;
69          else if (counter > 3000 && counter < 6000) angle += 5;
70      }
71      else if (flag == 1) {
72          angle += 0.5;
73          height += 0.0001;
74          counter++;
75          if (height > 6) {
76              flag = -1;// explode
77              height = 0;
78          }
79      }
```

```
80      else if (flag == -1) {
81          angle=0;
82      }
83      glutPostRedisplay();
84      // 100 milliseconds
85      glutTimerFunc(100, Timer, 0);
86  }
```

讓normal隨時間變色、製造閃爍感，並利用三種三角函數來調色

```
if(index==0) color=vec4(abs(sin(t)), abs(cos(t)), abs(tan(t)), t);
else if(index==1) color=vec4(abs(tan(t)), abs(sin(t)), abs(cos(t)), t);
else color=vec4(abs(cos(t)), abs(tan(t)), abs(sin(t)), t);
```

Explosion!!!

```
15  vec4 explode(vec4 position, vec3 normal)
16  {
17    float magnitude = 10.0f;
18    vec3 direction = normal * ((sin(t) + 1.0f) / 2.0f) * magnitude;
19    return position + vec4(direction, 0.0f);
20  }
21
22  vec3 GetNormal(int k)
23  {
24    vec3 a = vec3(gl_in[k].gl_Position) - vec3(gl_in[k+1].gl_Position);
25    vec3 b = vec3(gl_in[k+2].gl_Position) - vec3(gl_in[k+1].gl_Position);
26    return normalize(cross(a, b));
27  }
```