

01 – ポリゴン描画

コールバック関数

あるいは、関数ポインタと呼ばれるもの。関数を直接呼び出すのではなく、引数として渡し、引き渡し先のプログラム内で「呼び出し返す」(callback)処理。WindowsAPI のプログラムでは、Windows から投げられたメッセージを、事前に登録しておいたウィンドウプロシージャで処理している。

メッセージループ	ウィンドウプロシージャ
<pre>while (1) { if (メッセージが届いているか確認) { 届いたメッセージを処理するプログラムの呼び出し } else { // 更新処理 / 描画処理 など } }</pre>	<pre>LRESULT CALLBACK WndProc(/*省略*/) { switch (メッセージ) { メッセージごとに処理 } }</pre>

freeglut を利用したプログラムでは、呼び返される関数が細かく用意されており、glutMainLoop()内で関数ごとに呼び出されるべきタイミングで個別にコールバックされる。

メイン関数	※glutMainLoop()内の想像	描画に関するコールバック関数
<pre>glClearColor(0.8f, 0.9f, 1.0f, 1.0f); glutDisplayFunc(display); glutMainLoop();</pre>	<pre>Func g_displayFunc; void glutDisplayFunc(Func display) { g_displayFunc = display; } void glutMainLoop() { while(メッセージループなど) { if(g_displayFunc) g_displayFunc(); } }</pre>	<pre>void display() { glClearColor(GL_COLOR_BUFFER_BIT); glFinish(); }</pre>

更新処理

freeglut の描画処理は、内部で描画イベントが呼ばれた時のみ実行される(winapi でも同様)。初期化のプログラムのままでは一度だけしか呼び出されないため、タイマーを利用して再描画イベントを発生させる。

メイン関数	更新処理
<pre>glutDisplayFunc(display); glutTimerFunc(33, timer, 0); glutMainLoop();</pre>	<pre>void timer(int value) { glutPostRedisplay(); glutTimerFunc(33, timer, 0); }</pre>

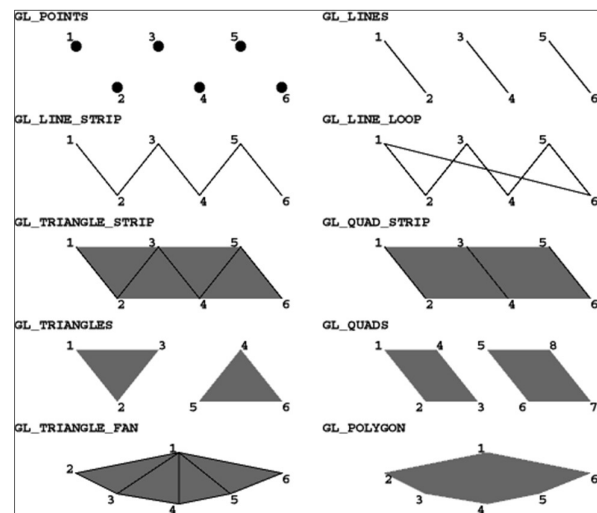
図形の表示

図形の描画は glBegin 関数～glEnd 関数の間で、各頂点の座標を設定する glVertex \blacksquare f 関数を呼び出す(\blacksquare は座標系 2D(2)or3D(3)を指定)。3D ゲームのオブジェクトは

「 \blacksquare 」で表現することが基本であり、頂点は「 \blacksquare 」で指定する。

glBegin 関数の引数には、指定された頂点からどのように図形を描画するのかを指定する。詳しくは右図を参照。

三角形の描画
<pre>glBegin(GL_TRIANGLES); glVertex2f(-0.5f, 0.5f); glVertex2f(-0.5f, -0.5f); glVertex2f(0.5f, -0.5f); glEnd();</pre>



参考: GLUT による「手抜き」OpenGL 入門
<https://tokoik.github.io/opengl/libglut.html#5.2>