コンピュータグラフィックス基礎　第 5 回 課題

課題の目標
・パラメトリック曲線の 1 つであるベジェ曲線の仕組みを理解する
・制御点を入力することで、ベジェ曲線を描画するアプリケーションの開発を行う
・C++言語の便利な機能の一つである STL の vector（可変長配列）を使えるようになる

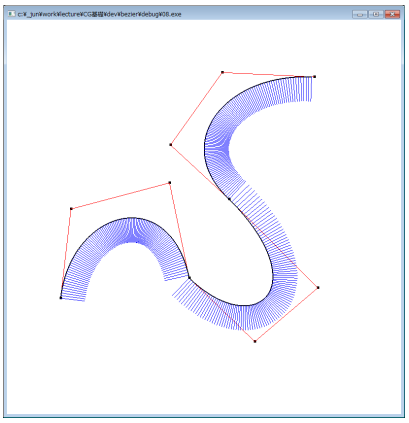
課題の内容

(1)　サンプルコード stl_vector_example.cpp の中身を確認し、C++言語で 2 次元ベクトルを扱ったり、サイズを変更可能な配列 vector を扱う方法を確認しなさい（2 次元ベクトルを扱う Vector2d クラスと、可変長配列を扱う vector は名前が似ているがまったく異なるものです)。その後、コメント文に記された課題の内容に従ってプログラムコードを追加し、実行結果を確認しなさい（プログラムコードと実行結果をレポートに含める)。

(2)　サンプルコード kadai05_sample.cpp に対して、下記の条件を満たすように display()関数を完成させなさい。プログラムができたら、なるべく滑らかに∞記号を描いてみなさい。また、なるべく綺麗な円を描いてみなさい。

（条件)

1. マウスの左クリックで制御点を追加、右クリックで削減する（サンプルコードで実現済み)。
2. 入力された制御点を使って、3 次の Bezier 曲線を描画する。下図のように制御点の数に応じて、複数のベジェ曲線が連結するようにすること。



つまり、制御点を P_0，P_1，P_2，P_3・・・とすると、1 つめのセグメントは P_3 が入力されたタイミングで P_0～P_3 の 4 点を使って描画され、2 つめのセグメントは P_6 が入力されたタイミングで P_3～P_6 の 4 点を使って描画される（制御点 P_3 を 1 つめのセグメントと共有する)。3 つめのセグメントは P_9 が入力されたタイミングで P_6～P_9 の 4 点を使って描画される（制御点 P_6 を 2 つめのセグメントと共有する)。以下同様。セグメント数に制限は設けない。

3. 上図のように Bezier 曲線の法線（接線を時計回りに 90 度回転させたもの）を描画する（長さはてきとうに決めてよい)。

(発展課題：オプション)

課題 3 で表示する法線の長さを、曲率に応じて変化するようにしなさい（曲率が大きい場所ほど長くする)。曲率の計算方法は、授業スライドを参考にするとよい。