

CS130 - LAB - Bézier curves

Name: _____

SID: _____

In this lab, we will render an approximation of a parametric curve known as the Bézier.

Consider the parametric equation of a segment between two control points P_0 and P_1

$$B(t) = (1 - t)P_0 + tP_1 \quad (1)$$

For n control points, we can recursively apply Eq. (1) to consecutive control points until we are left with only $P(t)$. For three control points, $B(t) = (1-t)[(1-t)P_0+tP_1]+t[(1-t)P_1+tP_2]$.

1. Given n control points, what is the degree of the polynomial equation for the Bézier curve? In general, $B(t)$ for $n + 1$ points is given by:

$$B(t) = \sum_{i=0}^n \binom{n}{i} t^i (1-t)^{n-i} P_i$$

■

2. Since we may need the factorial $n!$, combination $\binom{n}{k}$, and polynomial of $B(t)$ in this lab, complete the code to for these functions below.

```
float factorial(int n)
{
```

```
}
```

```
float combination(int n, int k)
{
```

```
}
```

```
float polynomial(int n, int k, float t)
{
```


N

3

v

{

}



2

5. Assumption: $P(0) = A$. Use this to solve for $a_0 = \underline{\hspace{1cm}}$, $b_0 = \underline{\hspace{1cm}}$, and $c_0 = \underline{\hspace{1cm}}$.
■
6. Assumption: $P(1) = C$. Use this to solve for $a_1 = \underline{\hspace{1cm}}$, $b_1 = \underline{\hspace{1cm}}$, and $c_1 = \underline{\hspace{1cm}}$.
■
7. Assumption: If $A = B = C$, then $P(t) = A$ for all t . Use this to solve for $b_2 = \underline{\hspace{1cm}}$.
■
8. Assumption: $P'(0)$ depends on A and B , but it does not depend on C . Use this to solve for $c_2 = \underline{\hspace{1cm}}$.
■
9. Assumption: $P'(1)$ depends on B and C , but it does not depend on A . Use this to solve for $a_2 = \underline{\hspace{1cm}}$.
■
10. Substituting in all of the coefficients and *factoring the resulting polynomials* produces $P(t) = (\underline{\hspace{2cm}})A + (\underline{\hspace{2cm}})B + (\underline{\hspace{2cm}})C$.
■
11. One can show that $P'(0) = \alpha(B - A)$ and $P'(1) = \beta(B - C)$. Find α and β .
■

Part 2: Coding

Download the skeleton code and modify `main.cpp` as follows:

- Define a global vector to store the control points.
- Push back the mouse click coordinates into the vector in the function `GL_mouse`.
- Write the code for the `factorial`, `combination` and `binomial`.
- Draw line segments between points along the Bézier curve in `GL_render()`.
- You can use `GL_LINE_STRIP` to draw line segments between consecutive points.
- You can iterate t between 0 and 1 in steps of 0.01.

Optional: Rather than using the general equation for the Bézier curve to write your program, you can write a program where you recursively apply Eq. (1) to consecutive points to get $B(t)$. Alternatively, you can use the more efficient algorithm `coefficients`.