CS411 – Report 4

Chengxi Shi

A20353344

Writing Quesitons:

Writing Questions answer should in "Chengxi_Shi_CS411_Assignment 4.pdf"

Coding Questions:

All function implements should in file "cs411-assignment4-template.js"

Summary for program design:

Caused by the skeleton code, the things we need to finish is implementing the interpolation into Cardinal spline, adding/decreasing the tension and then refresh all the curves.

1. Cardinal spline:

A: we can choose both matrix form and blending function for implement. It is on the page 7 and 8 of the handouts. Here I will pick up blending function. Since it gives four points with totally 4 x and 4 y value as pk_n1.x/y, pk.x/y, pk_p1.x/y, pk_p2.x/y, we tried to use the blending function to get x and y separately. For x, it will equal to ((0-tension)*u*u*u+2*tension*u*u-tension*u)*(pk_n1.x)+((2-tension)*u*u*u+(tension-3)*u*u+1)*(pk.x)+((tension-2)*u*u*u+(3-2*tension)*u*u+tension*u)*(pk_p1.x)+(tension*u*u*u-tension*u*u)*(pk_p2.x);

It will be the similar as y, which only change x to y to get the y value.

After getting both x and y, it will do the push and finish interpolate.

2. Tension:

A: we want to increase/decrease the tension of the curves and refresh all the curves with the new tension. Thus, I will put the changing functions into tensionUp/tensionDown function. We will get the idea from click function. It asks the value of the total points, and each time pick up 4 points to do interpolation. As the same way, we will check the points numbers in the new tension function. If points are

equal or bigger than 4, we will go to next step, if not, just return the tension value.

For points that is 4 or more, we clean all the things in the intrPts. We then go to the beginning, which is nCtrlPts = 4, to do the interpolation. It will stop at nCtrlPts<=ctrlPts.length/2.

Questions faced and solution:

Writing part:

1. How to use the given points

A: P_k, P_{k+1}, dP_k, dP_{k+1} will all put in a 4x2 Matrix vertically. Thus, [a, b, c, d]^T =

$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{pmatrix}^{-1} \begin{pmatrix} 0 & 0 \\ 2 & 2 \\ 1 & 1 \\ 1 & -1 \end{pmatrix}$$

2. Problem (f) and (g) unequal results

A: for question f, the matrix should be, where s is the tension and it s = 0.5 in the

$$\begin{pmatrix} 1 & 1 & 1-\frac{1}{s} & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ \frac{3}{s} & \frac{2}{s} & \frac{1}{s} & 1 \end{pmatrix}^{-1}$$

questions. I calculated it originally with 3*s, 2*s, 1*s, which give me a different answer with problem (g)

3. How to use knot vector? (question (l) and (m))

A: handouts can be found on page 12 and 13. Totally speaking, for a u and weight

$$B_k^2(u) = \frac{u-u_k}{u_{k+1}-u_k} B_k^1(u) + \frac{u_{k+2}-u}{u_{k+2}-u_{k+1}} B_{k+1}^1(u) = \begin{cases} \frac{u-u_k}{u_{k+1}-u_k} & u_k \le u < u_{k+1} \\ \frac{u_{k+2}-u}{u_{k+2}-u_{k+1}} & u_{k+1} \le u < u_{k+2} \\ 0 & \text{otherwise} \end{cases}$$

graph, at any u value, the sum value for all the points on that u will be 1. Furthermore, the B-spline blending function can be calculated by formula on page 13 which should be

Coding part:

1. How to do the new tension?

A: since it requires refresh all the curves with new tension value, so we need to clean it all up and then do the calculation from the beginning. Actually, I found for the ctrlPts for a while to get the original interpolation calling.

2. Debugging:

a. var i = 0, not int i = 0

b. intrPts.length, not length(interpolate)