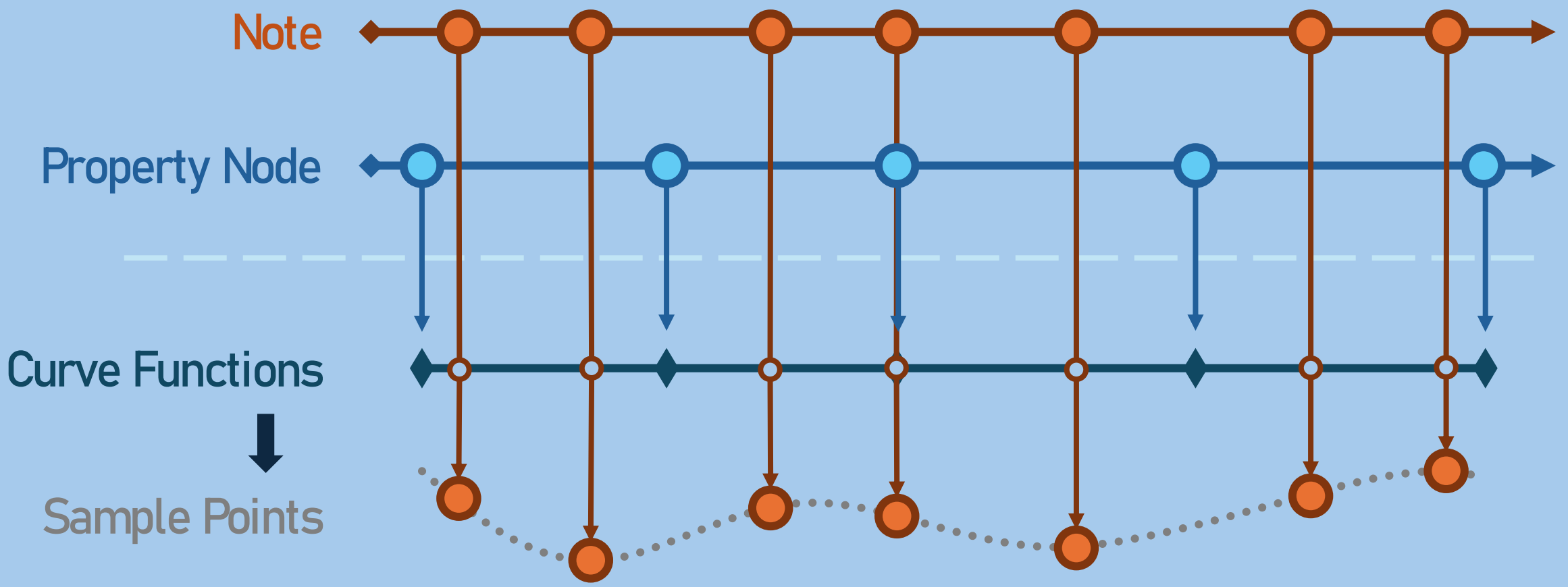
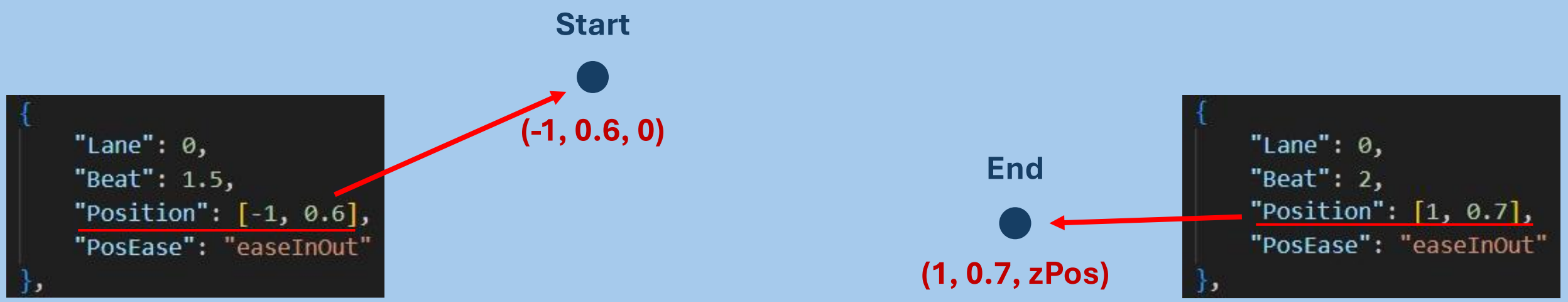
CS Project Progress Report

111550037 嚴偉哲

1- Lane Generation (One-Lane Example)



1- Lane Generation (One-Lane Example)



1- Lane Generation (One-Lane Example)

Start

(-1, 0.6, 0)

```
{
    "Lane": 0,
    "Beat": 1.5,
    "Position": [-1, 0.6],
    "PosEase": "easeInOut"
},
```

End

(1, 0.7, 4.44)

```
{
    "Lane": 0,
    "Beat": 2,
    "Position": [1, 0.7],
    "PosEase": "easeInOut"
},
```

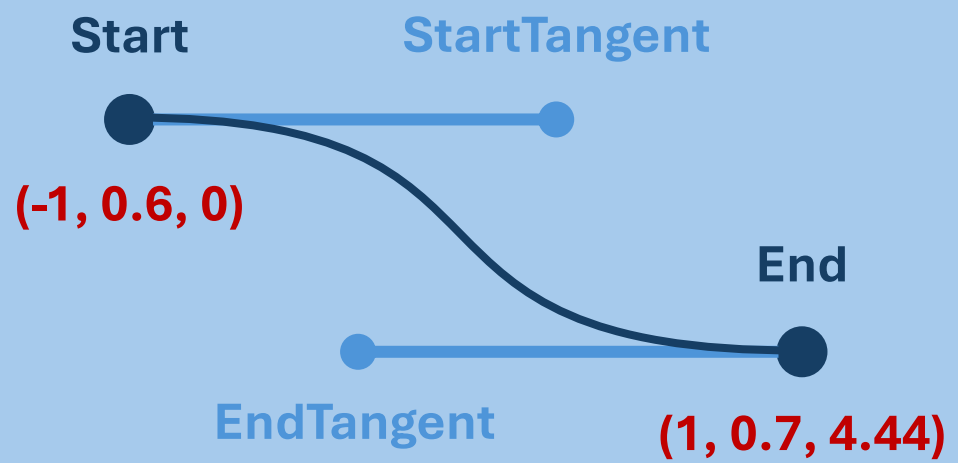
Default = 1

```
Compute.speed(nodes[nodeID].Speed) * Compute.beatToSecond(nodes[nodeID].Beat, nodes[nodeID + 1].Beat, chart.bpm);
```

zPos = (speed * 20) * ((endBeat - startBeat) * 60 / bpm)

1- Lane Generation (One-Lane Example)

```
{
    "Lane": 0,
    "Beat": 1.5,
    "Position": [-1, 0.6],
    "PosEase": "easeInOut"
},
```




```
{
    "Lane": 0,
    "Beat": 2,
    "Position": [1, 0.7],
    "PosEase": "easeInOut"
},
```

```
case "easeInOut":
    startTangent = new Vector3(0, 0, 0.76f);
    endTangent = new Vector3(1, 1, 0.24f);
```

- Use Bezier curve to implement
- Scale & translate the control points first

1- Lane Generation (One-Lane Example)

Sampling -

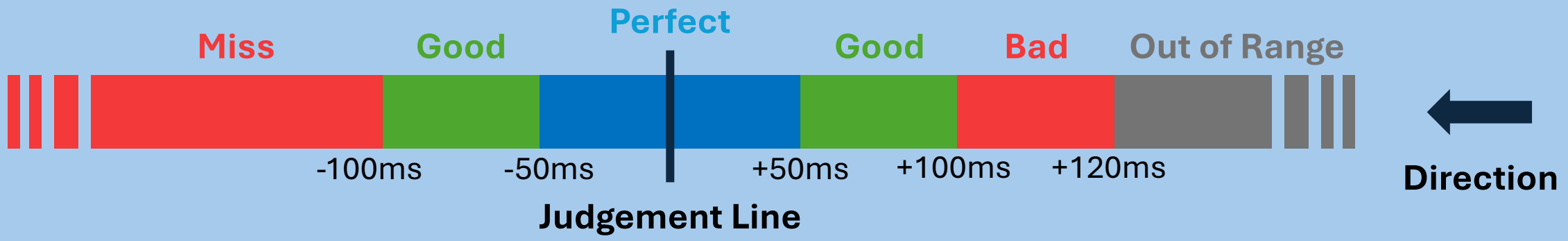
```
// get the local position of a point on the curve (0 <= t <= 1)
3 個参考
public Vector3 GetCurvePoint(float t)
{
    float it = 1 - t;
    float it2 = it * it;
    float t2 = t * t;

    return start * (it2 * it) + startTangent * (3 * it2 * t) + endTangent * (3 * it * t2) + end * (t2 * t);
}
```

Approximating (Bisection Method) -

- Note position is in Cartesian form
- Curve position is accessed in parametric form

```
// Approximate the position for a given z-axis distance
1 個参考
public Vector3 ApproxPoint(float d)
{
    // bisection method
    float a = 0;
    float b = 1;
    while (b - a > 1e-5f)
    {
        if (GetCurvePoint((a + b) / 2).z < d)
        {
            a = (a + b) / 2;
        }
        else
        {
            b = (b + a) / 2;
        }
    }
    return GetCurvePoint((a + b) / 2);
}
```

2- Scoring / Accuracy



- Time-based instead of distance-based