# Unit 3 – Lesson 8. Waypoint Implementation: Catmull – Rom, Interpolation and Spline

**Aim:**

* How do we present curves in games?

**Objectives:** After the lesson, students should be able to:

* Obtain basic understanding of curves, interpolation and spline

**CLASS PROCEDURE:**

***Do Now:*** In the CarWayPointsBased AI example, the Catmall-Rom equation is used to get the route positions. Here are the C# methods. What is the Catmall – Rom equation? Why do we need to use the Catmall – Rom equation to smooth the way points defined route?

public Vector3 GetRoutePosition(float dist)

{

int point = 0;

if (Length == 0)

{

Length = distances[distances.Length - 1];

}

dist = Mathf.Repeat(dist, Length);

while (distances[point] < dist)

{

++point;

}

// get nearest two points, ensuring points wrap-around start & end of circuit

p1n = ((point - 1) + numPoints)%numPoints;

p2n = point;

// found point numbers, now find interpolation value between the two middle points

i = Mathf.InverseLerp(distances[p1n], distances[p2n], dist);

if (smoothRoute)

{

// smooth catmull-rom calculation between the two relevant points

// get indices for the surrounding 2 points, because

// four points are required by the catmull-rom function

p0n = ((point - 2) + numPoints)%numPoints;

p3n = (point + 1)%numPoints;

// 2nd point may have been the 'last' point - a dupe of the first,

// (to give a value of max track distance instead of zero)

// but now it must be wrapped back to zero if that was the case.

p2n = p2n%numPoints;

P0 = points[p0n];

P1 = points[p1n];

P2 = points[p2n];

P3 = points[p3n];

return CatmullRom(P0, P1, P2, P3, i);

}

else

{

// simple linear lerp between the two points:

p1n = ((point - 1) + numPoints)%numPoints;

p2n = point;

return Vector3.Lerp(points[p1n], points[p2n], i);

}

}

1. What does the CatmullRom() method do?

private Vector3 CatmullRom(Vector3 p0, Vector3 p1, Vector3 p2, Vector3 p3, float i)

{

// comments are no use here... it's the catmull-rom equation.

// Un-magic this, lord vector!

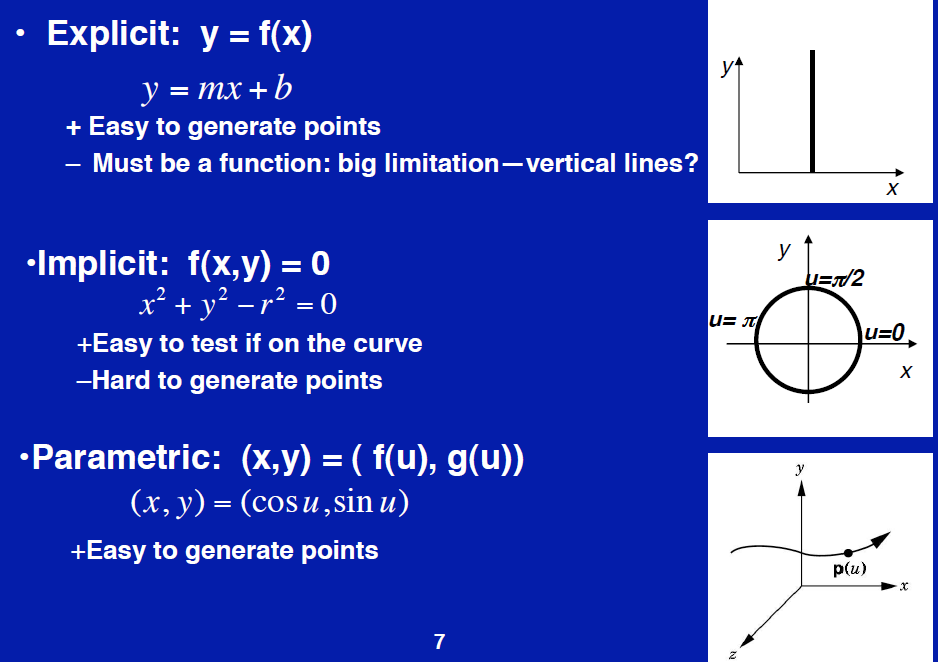
return 0.5f\*

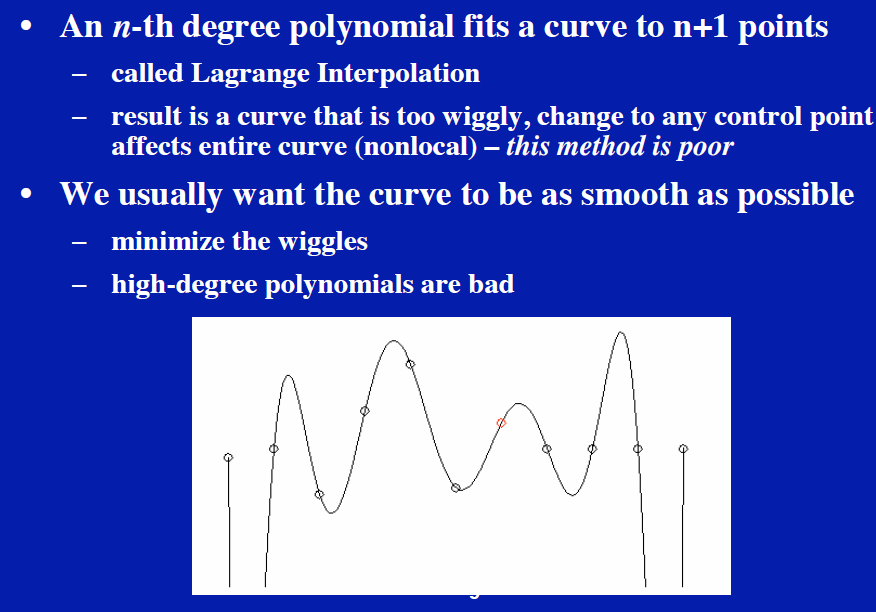
((2\*p1) + (-p0 + p2)\*i + (2\*p0 - 5\*p1 + 4\*p2 - p3)\*i\*i +

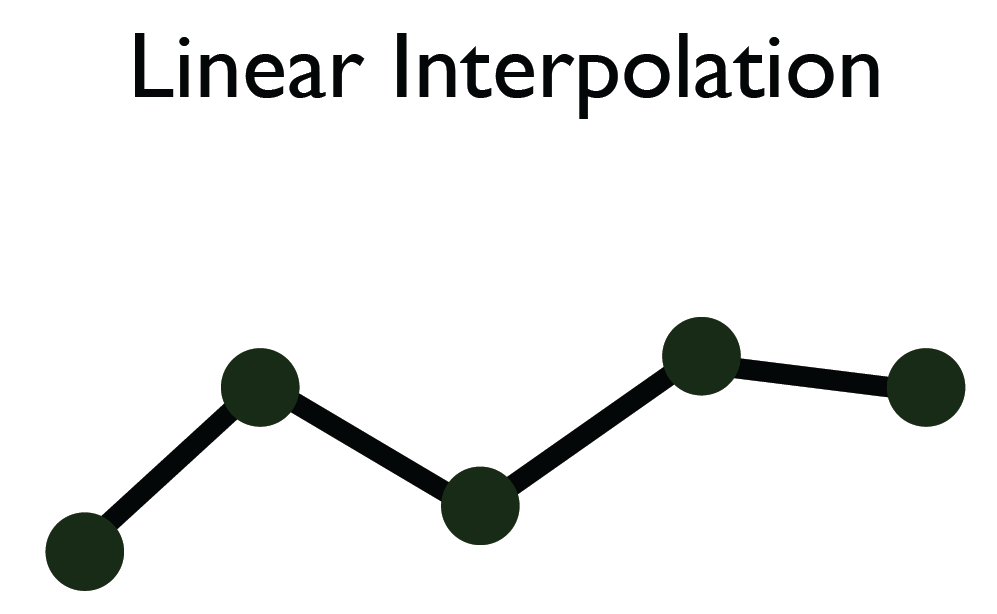
(-p0 + 3\*p1 - 3\*p2 + p3)\*i\*i\*i);

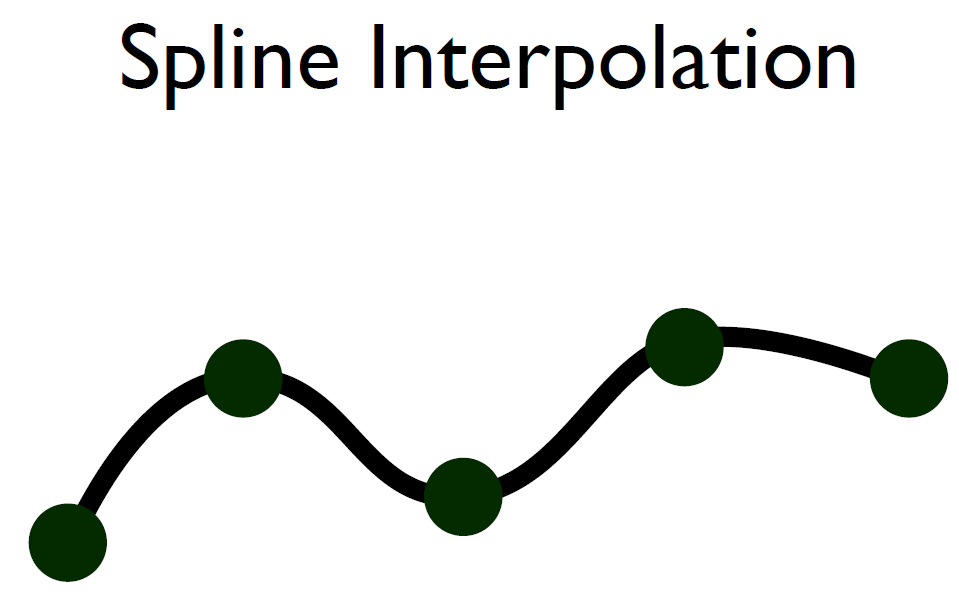
}

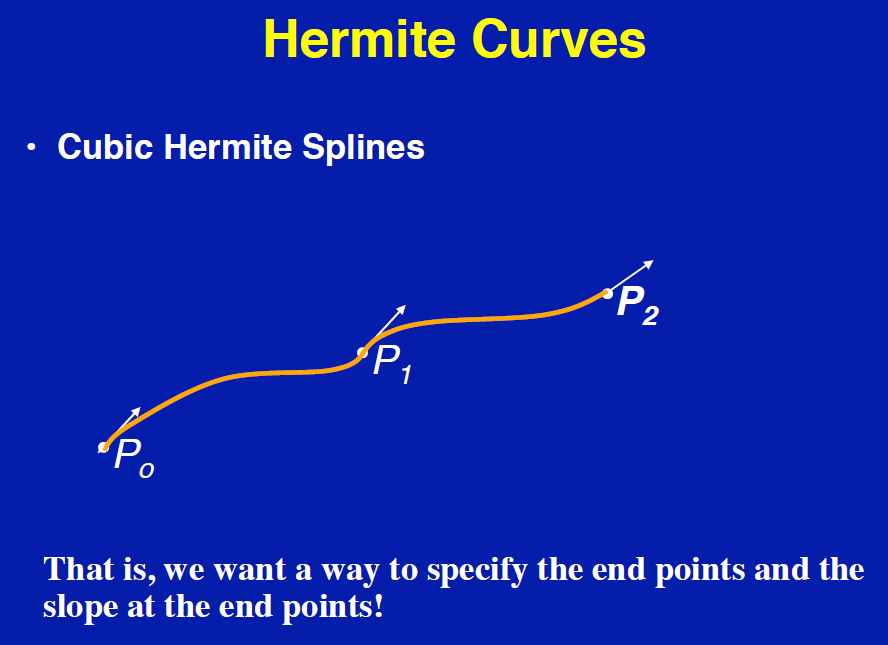
***Class Discussion / Presentation:*** How do we present curves and what is the Catmull – Rom equation?

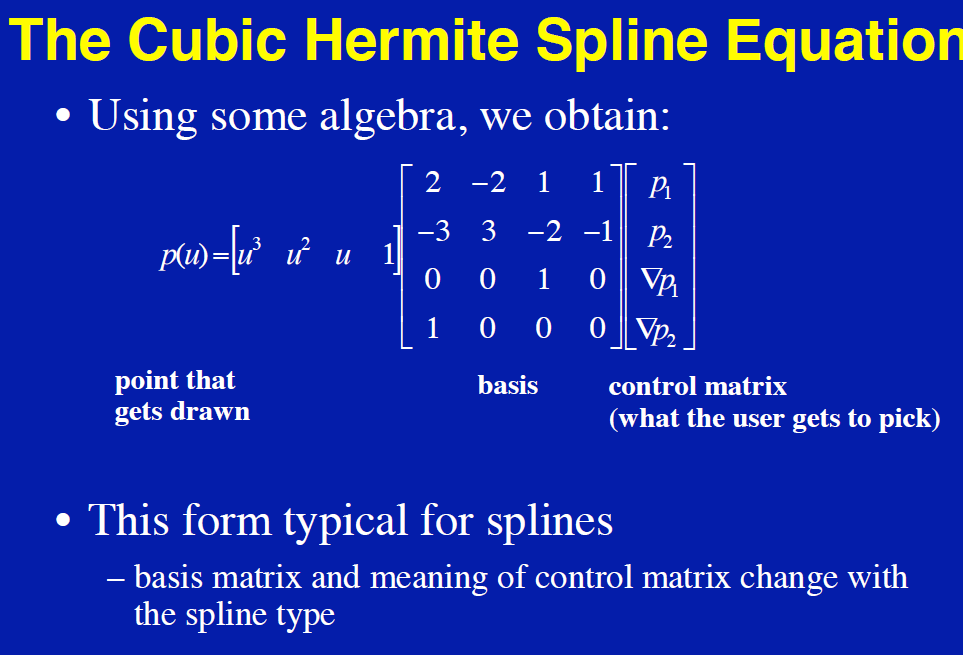


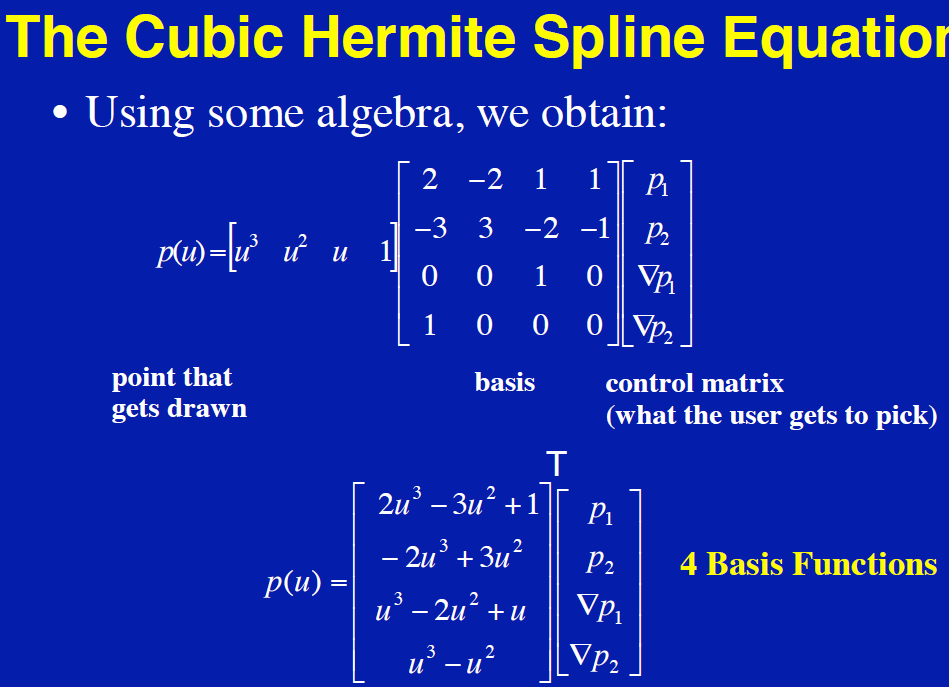


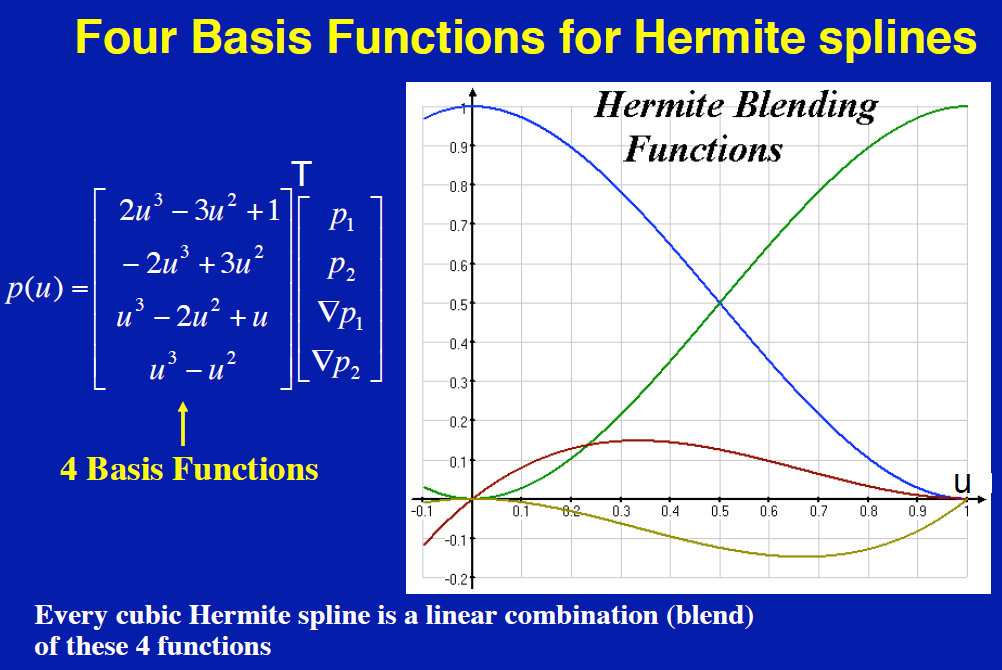


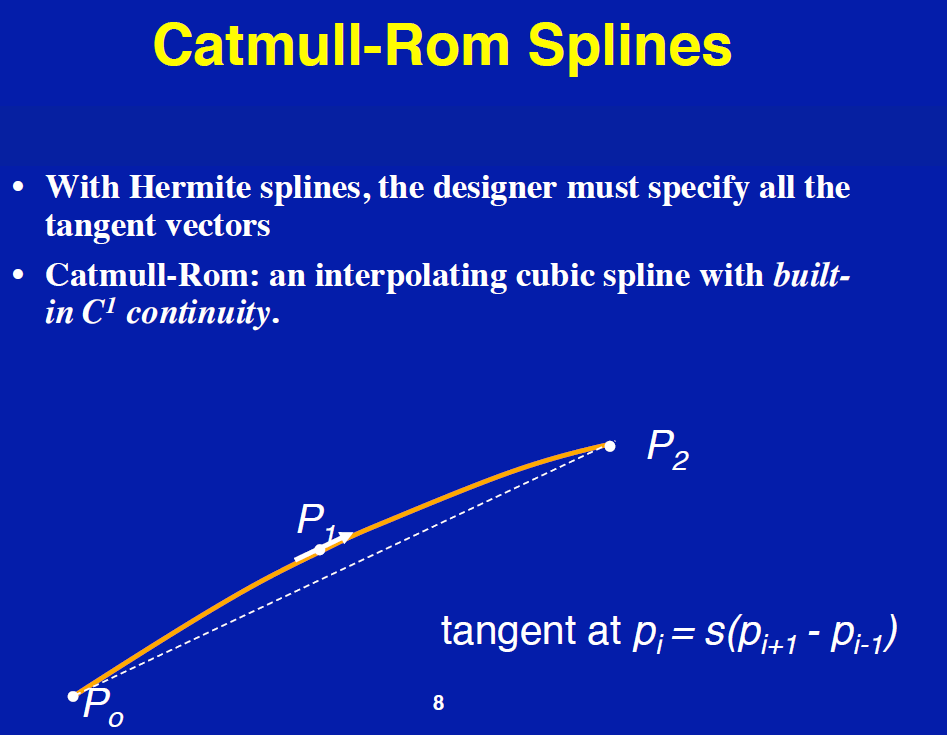




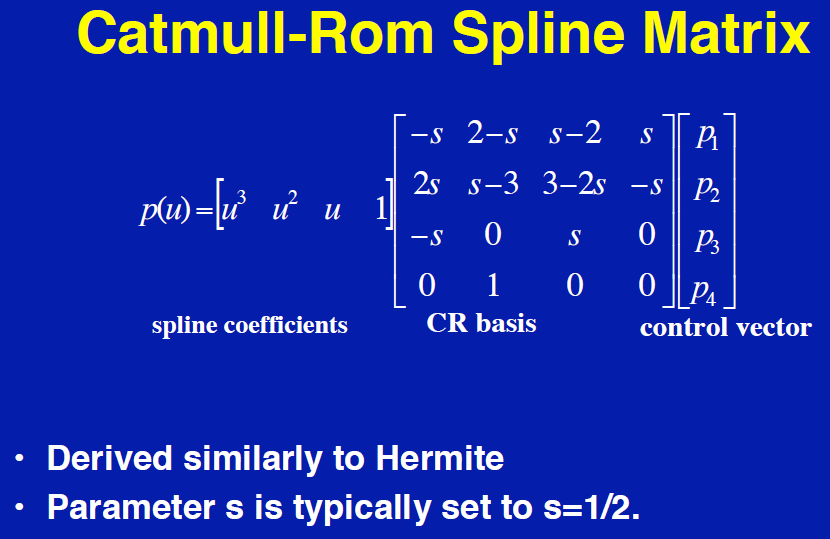








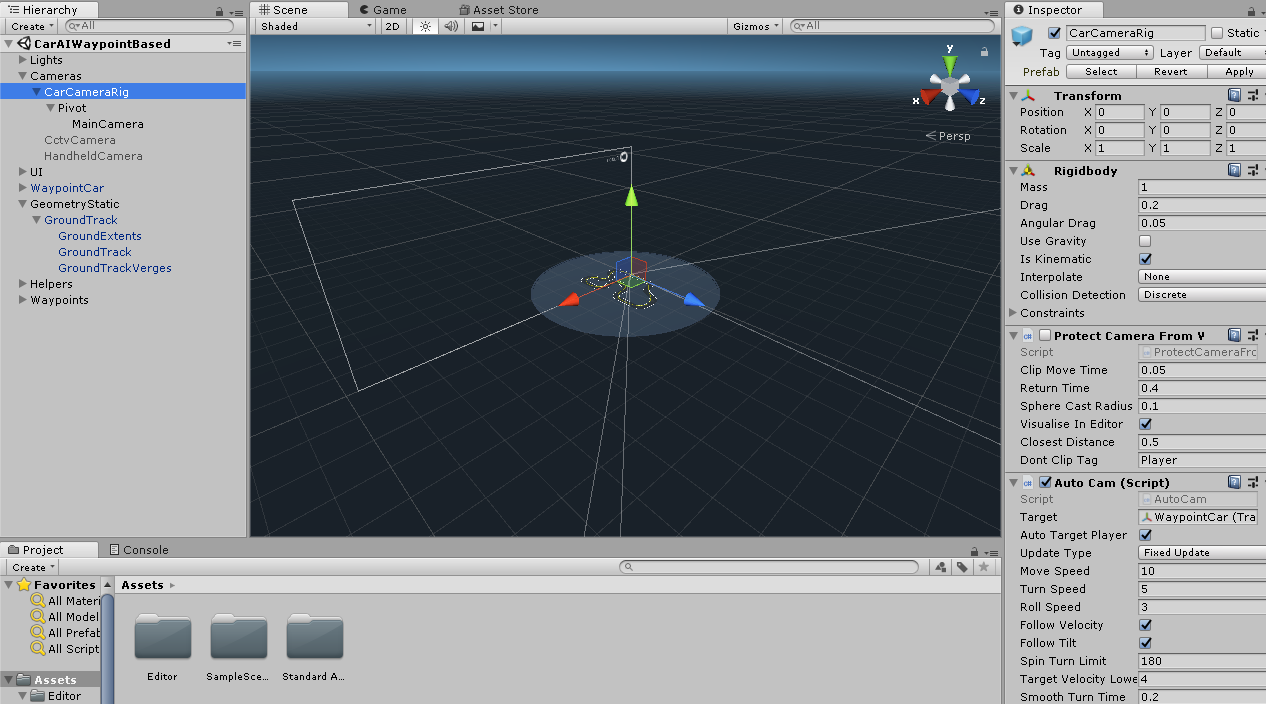




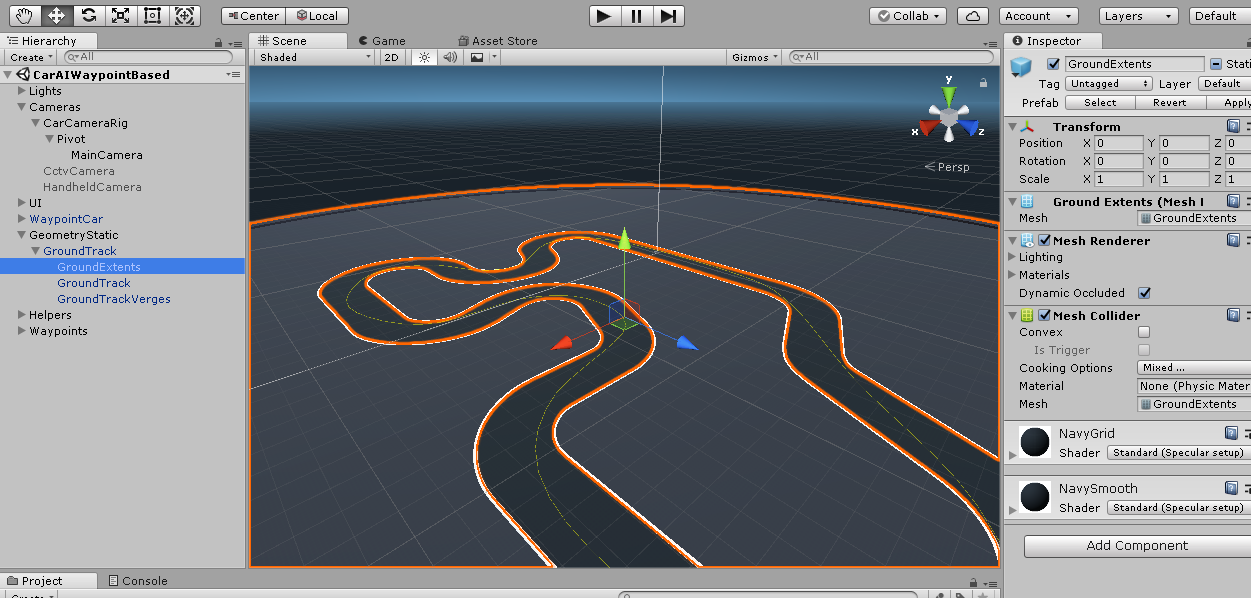
***Pair – sharing Activity:***

Explore the sample Car AI pathfinding games and projects. Answer the following questions:

1. How does the project switch the cameras?



1. What is the GeometryStatic? Why do we need it in this game?



***Pair – sharing Activity:***

Work with your team on the Maze Runner Game Design Document. Due: December 15th. Submit your work to Google Classroom.