**1. Matrix effect Sequence - 23 Math for Game Developers - Prop Positioning (TRS Matrices)**

number 1:Scaling

number 2:Rotation

number 3:Translation

But take it to equation, it become: Translation \* Rotation \* Scaling \* V

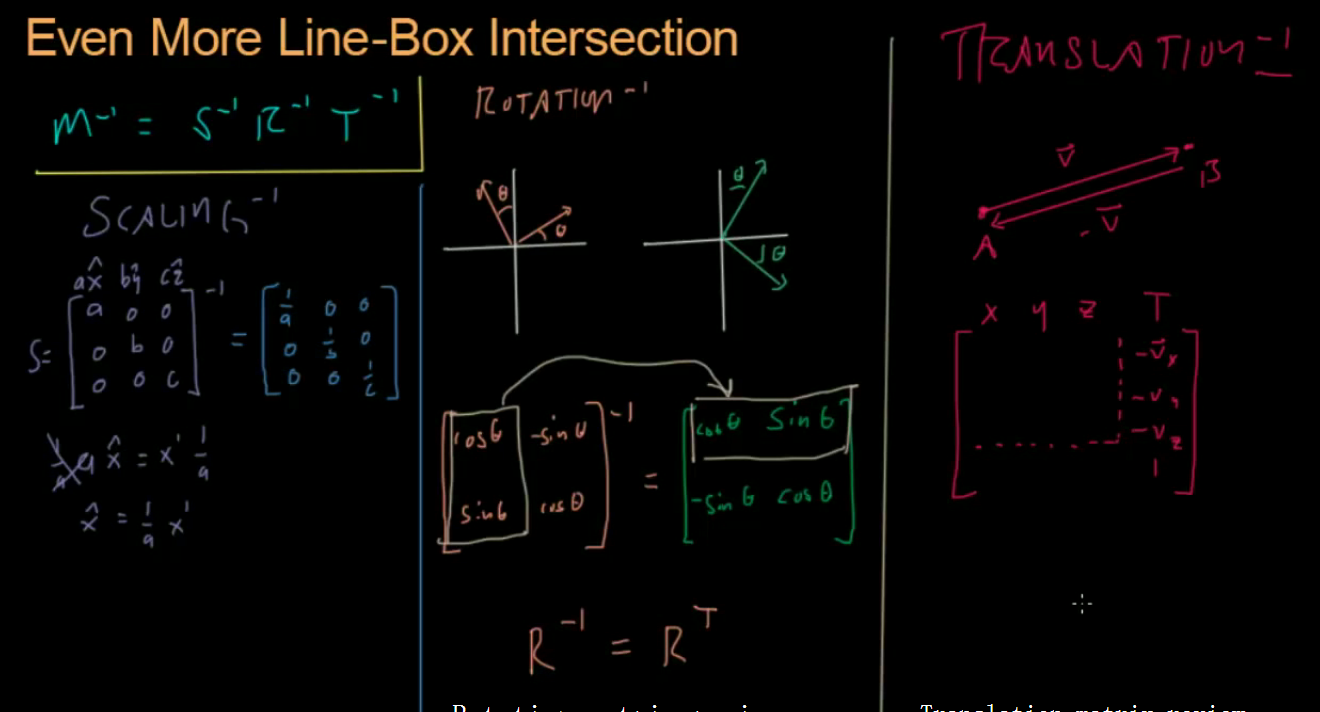
Because the one nearest to the v will first make effect.

**2. inverse matrix - 24-25 Updated Bullet Collisions (Coordinate Systems)**

Inverse matrx can do the revert operation of a matrix

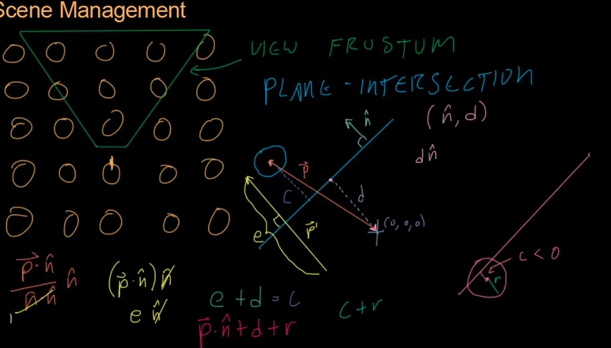
make intersection between no-algin AABB and a line, you can transform line to the local coordinate system of AABB

These tutorial also teach you how to get inverse matrix



3. View Frustum Cull - Math for Game Developers - Frustum Culling

Jude by the side of 6 plane.



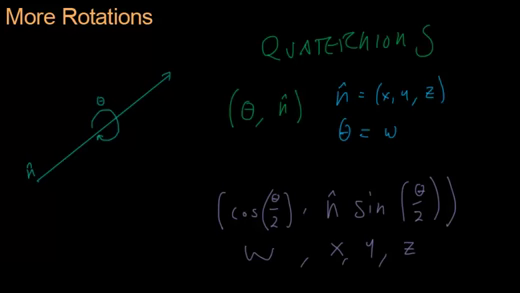
4. merge sort - Code for Game Developers - Drawing Transparent Characters (Mergesort)

For faster index rendering entities.

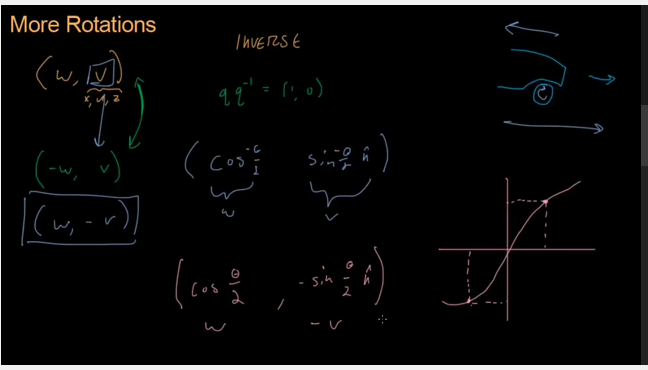
5. Quaternion – 31 Math for Game Developers - Rotation Quaternions

The purpose is to rotate by different axis together.

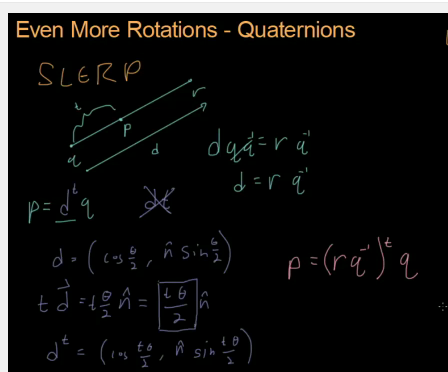
The equation from angle, axis vector to quatenion:



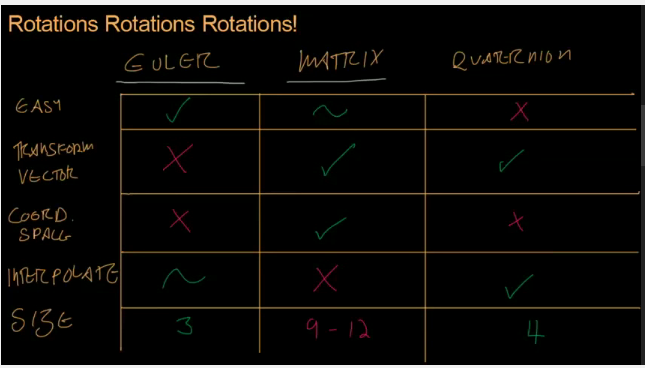
6. How to get inverse quaternion. – 32 Math for Game Developers - Quaternion Inverse



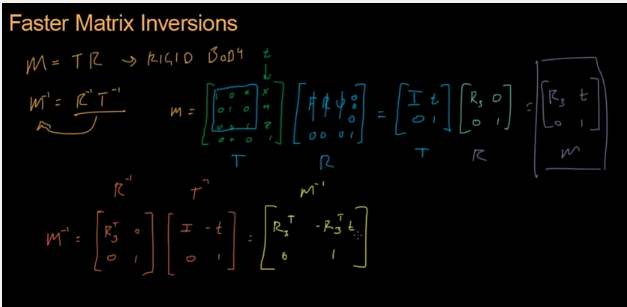
7. Quaternion liner interpolation



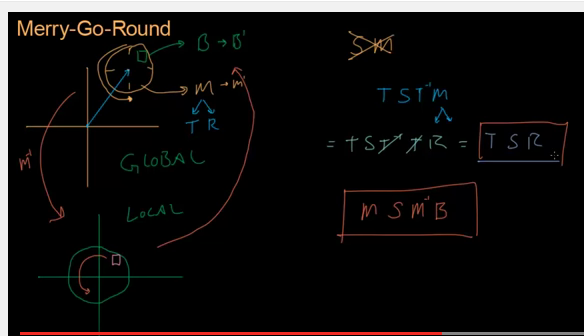
8. Compare Euler, Matrix and Quaternion rotation.



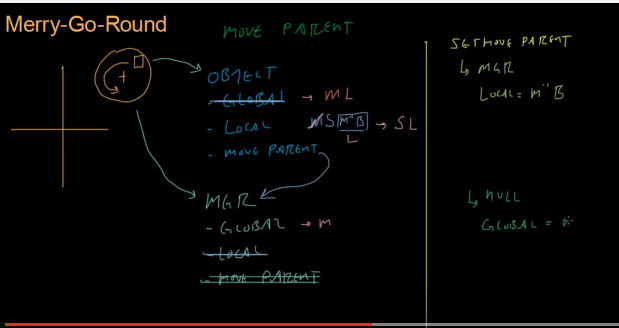
9. Faster Matrix Inversion – without scaling



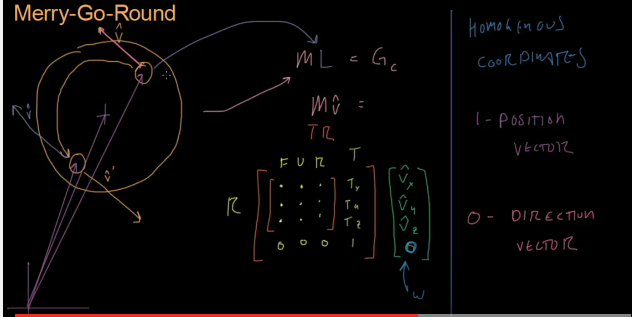
10. Object rotate in local coordinate – 40 Math for Game Developers - Merry-Go-Round (Matrix Transformations)



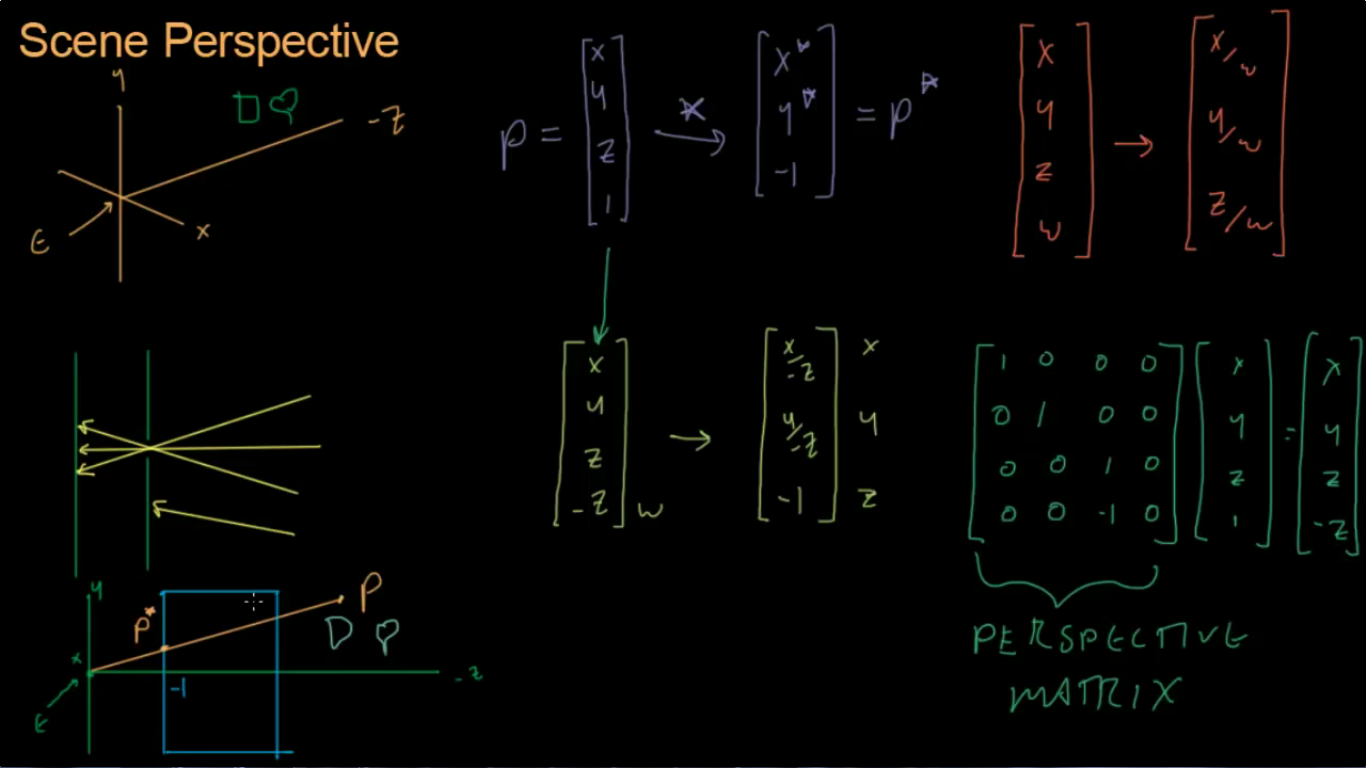
11. move parent – 40 Code for Game Developers - Move Parents



12. rotate vector – 41 Math for Game Developers - Homogenous Coordinates



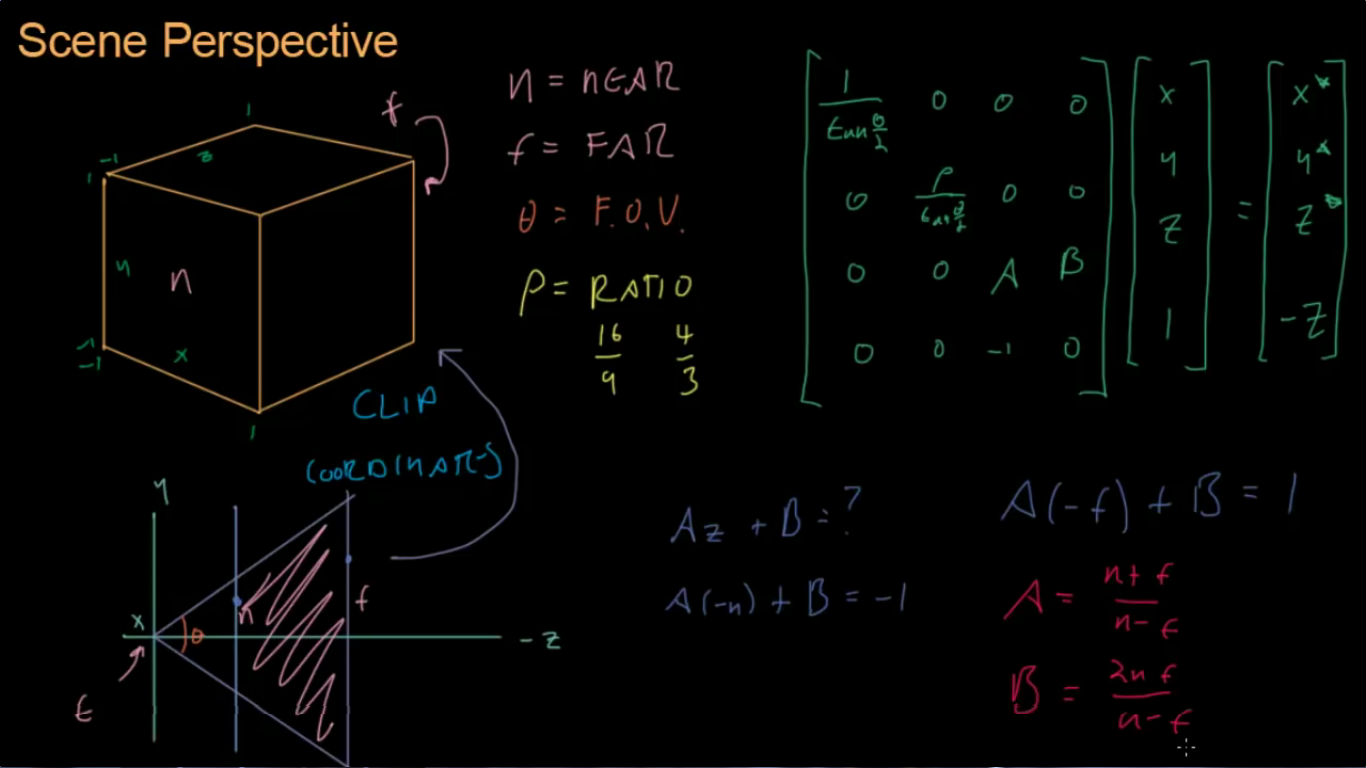
13. Perspective matrix part1 (project to a plane) – 42. Math for Game Developers - Perspective Matrix



14. Perspective matrix part2 – 43 Math for Game Developers - Perspective Matrix Part 2

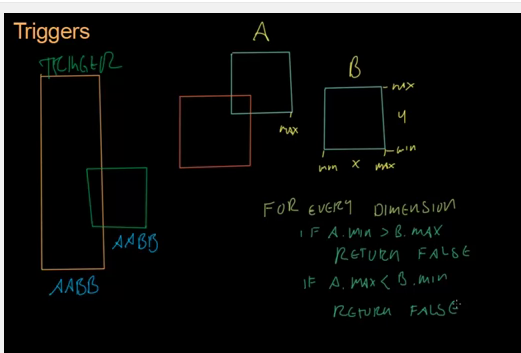
Project object from near and far plane to a cube area

Clip Coordinate

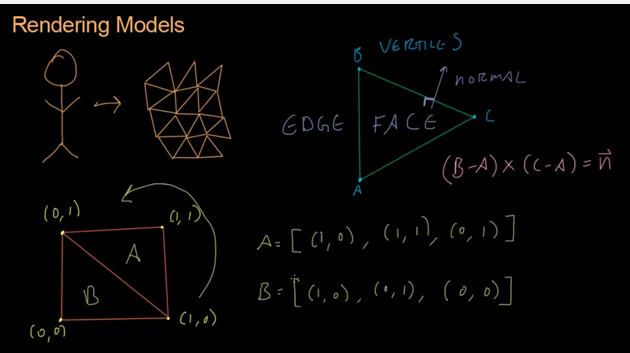


15.AABB intersection - Math for Game Developers - Trigger Areas (AABB Intersection)

If all dimension overlap, the two AABBS intersected.

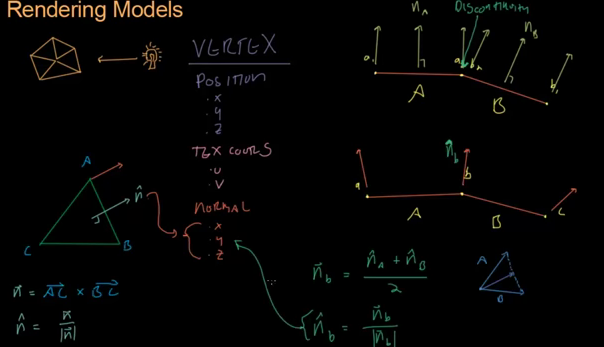


16. Rending models – 45 Math for Game Developers - Triangle Meshes

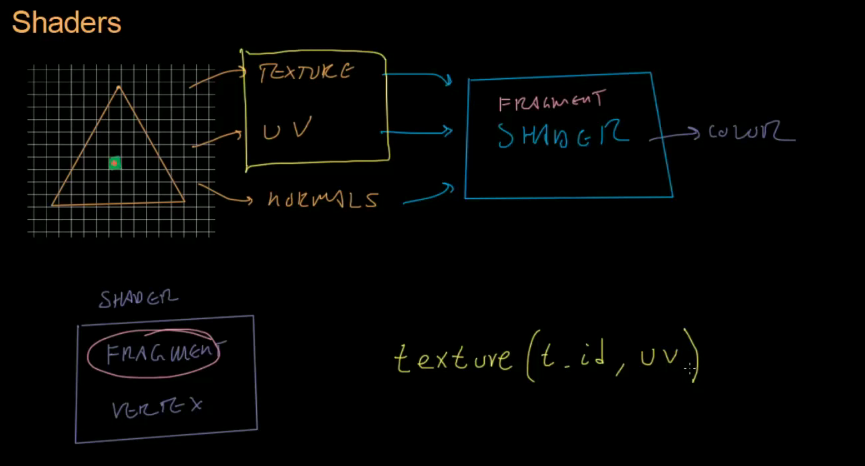


17. The normal of vertex in triangle – 50 Math for Game Developers - Triangle Mesh Normals

It is the average normal of adjoining triangle.



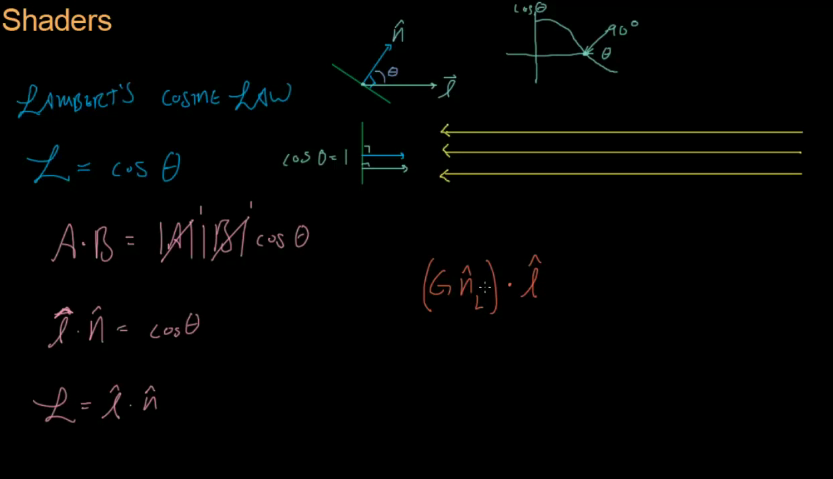
18. Shaders – 54 Math for Game Developers - Fragment Shaders



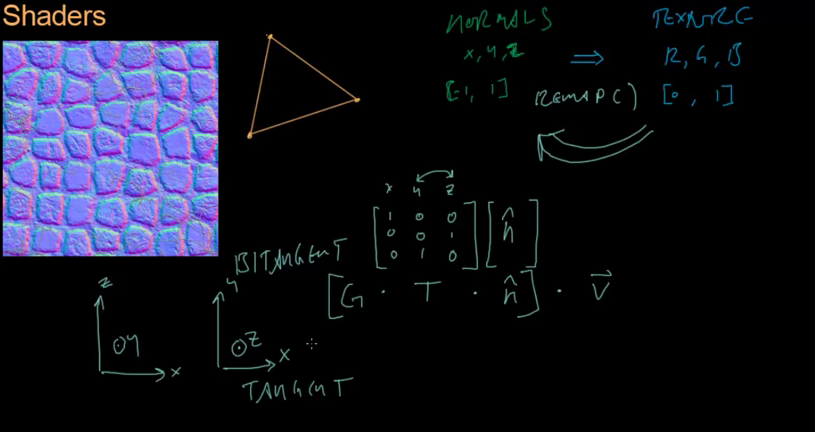
19. First shading program –Math for Game Developers - Lambertian Surfaces

Introduce the basic lighting model (Cosine Law) in fragment shader.

Use RemapVal function to reduce the contrast.



20. Normal map – 52 Math for Game Developers - Normal Maps

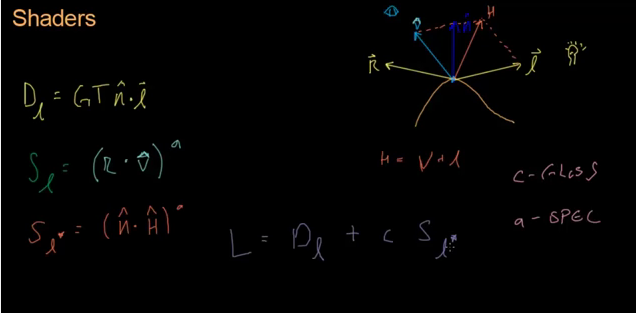


21. Specularity – 55 Math for Game Developers – Specularity

Specular is to calculate the angle between eye direction and reflection. While diffuse is to calculate the angle between the normal direction and the light direction.

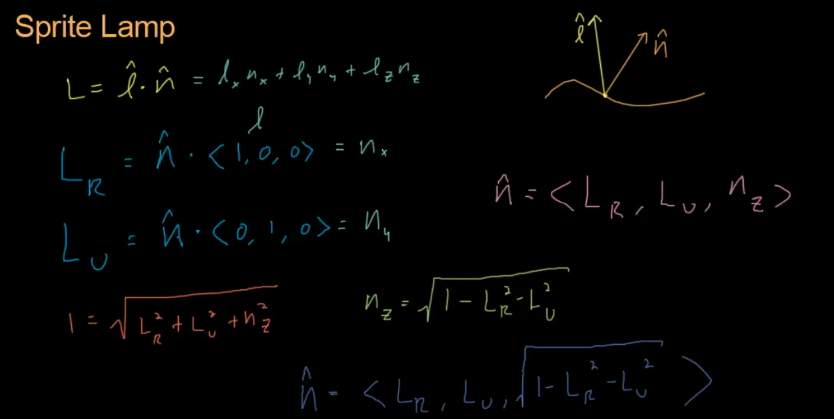
It also can store a or c to a texture which like normal map.

Since it take high calculator spend, it find the other way to measure specular which I cannot understand.

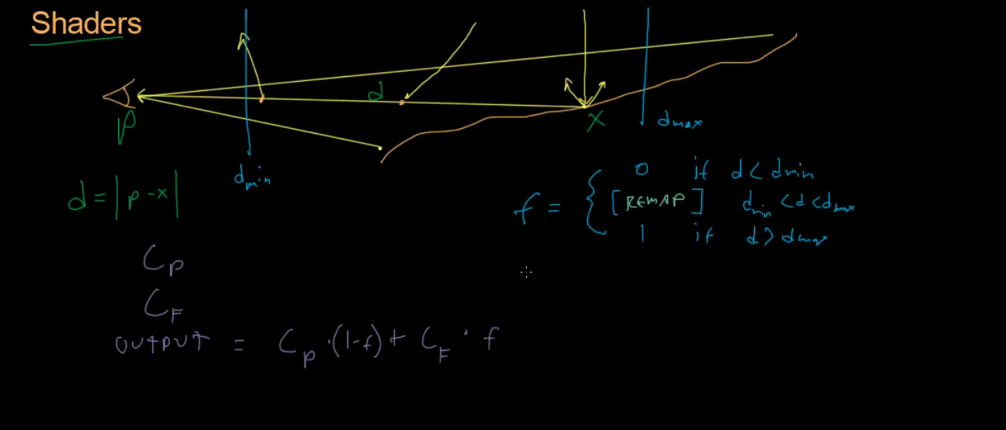


22. Sprite Lamp – 56 Math for Game Developers - Sprite Lamp Normal Generation

Make 2d character like 3d character.



23. Fog - Math for Game Developers - Fog



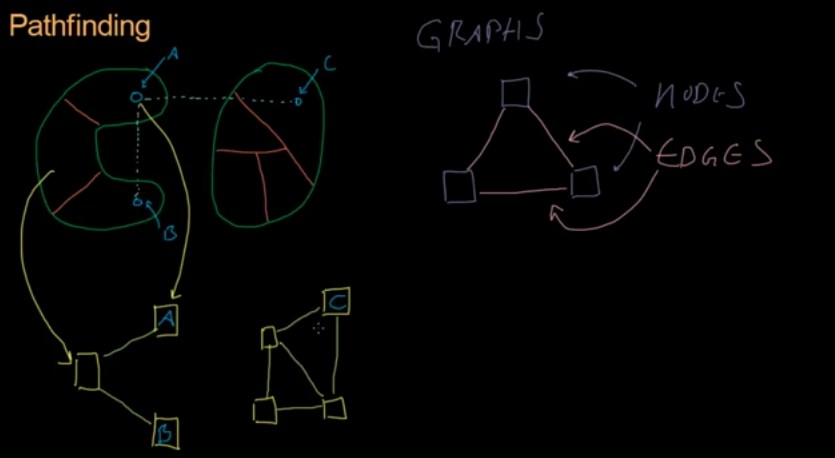
24 Rim Lighting – 58 Math for Game Developers - Rim Lighting

Can use to highlight character.



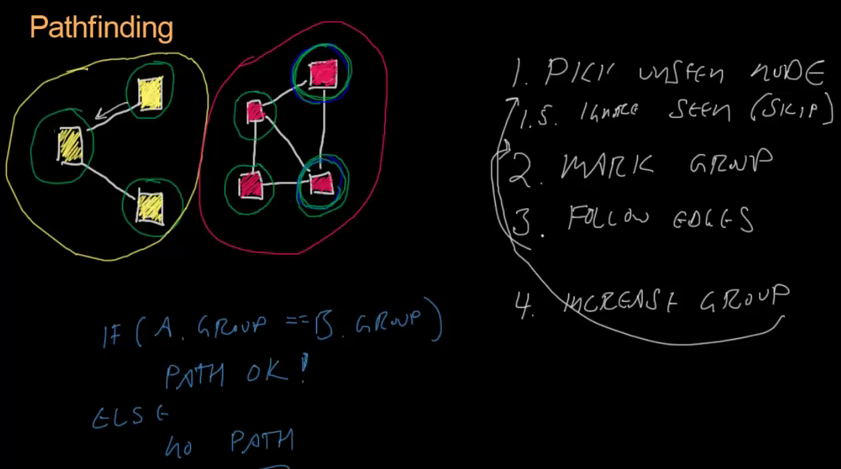
25. Graph for path finding – 60 Code for Game Developers - Pathfinding (Graphs)

Graph comprises nodes and edges. Graph can help us to find the path between nodes or to confirm whether the nodes are connected.



26. Graph for judging connectivity – 60 Code for Game Developers - More Pathfinding (Graphs)

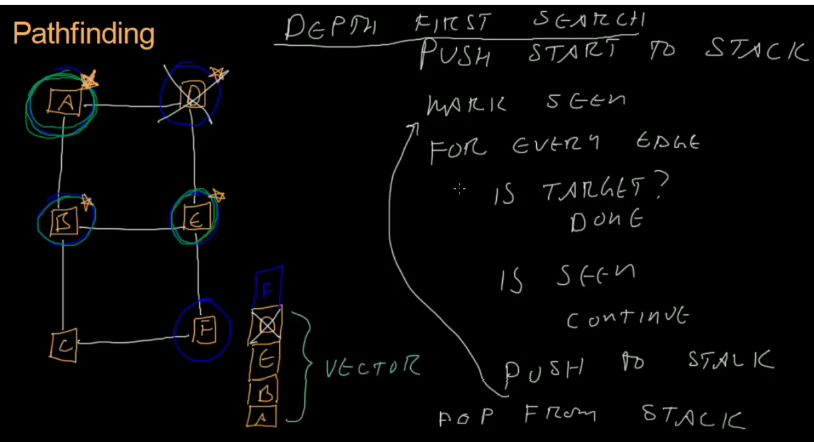
Loop nodes to dividing them to different group. The nodes in the same group have connectivity.



27. Graph for searching the path of nodes – 61 Code for Game Developers - Depth First Search

The drawback is that it is not the most efficient way to find the path.

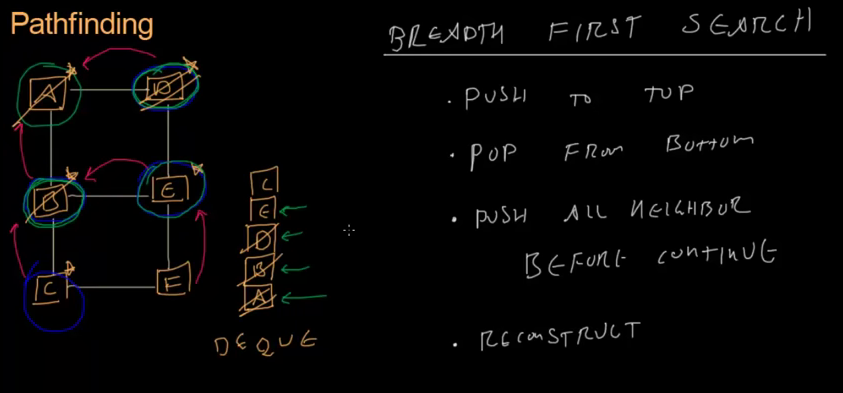
Use the structure of stack.



28. Graph for searching the path of nodes - 62 Code for Game Developers - Breadth First Search

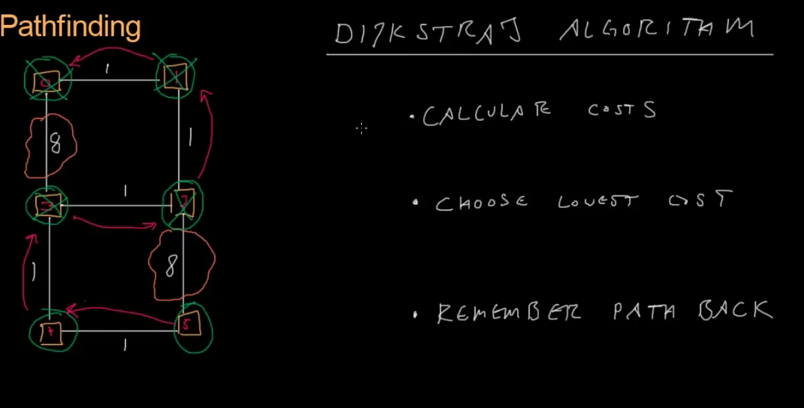
Use the structure of queue

It can get the shortest path because item from our every start point just is 1 step.



29. Dijkstra's algorithm is an algorithm for finding the shortest paths between nodes in a graph.

63 - Code for Game Developers - Dijkstra's Algorithm



30. Heap (63 Code for Game Developers – Heaps)

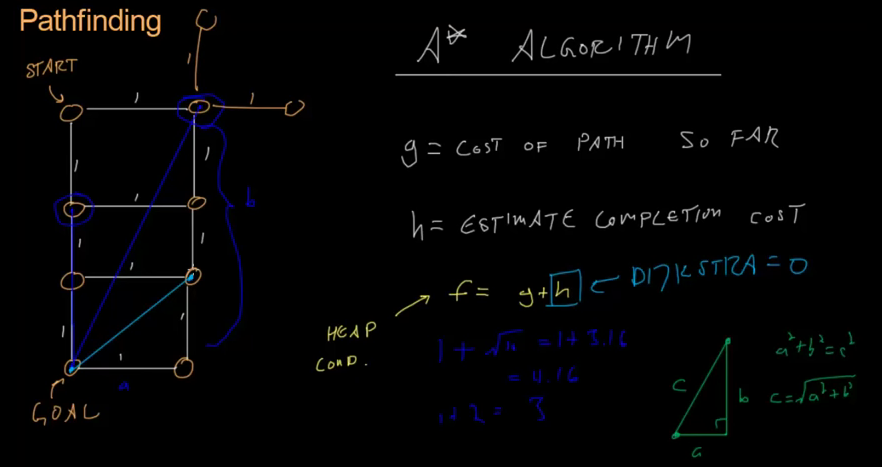
It is a fast way to automatic order data. Heap is a binary tree, and the smallest node always on the top, the bigger one on the bottom. It is easy to reorder when pop the top node. Efficiency is Log2N.

Can use array to implement heap. Children are 2 \* n + 1

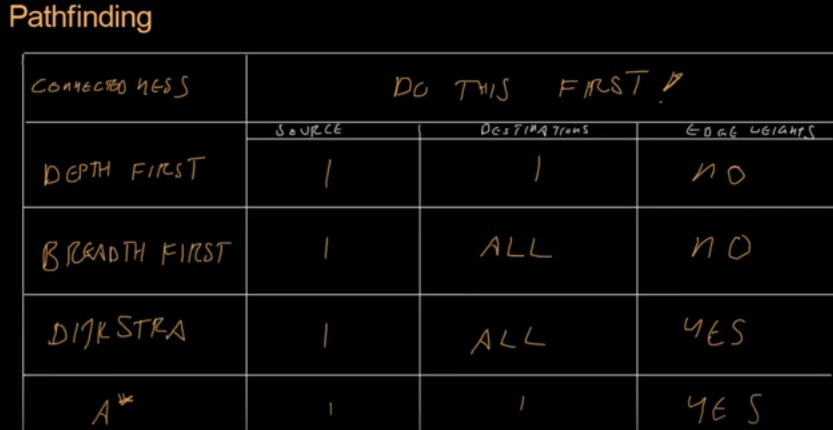
STL: std::pop\_heap

31. Code for Game Developers - A\* Pathfinding 66

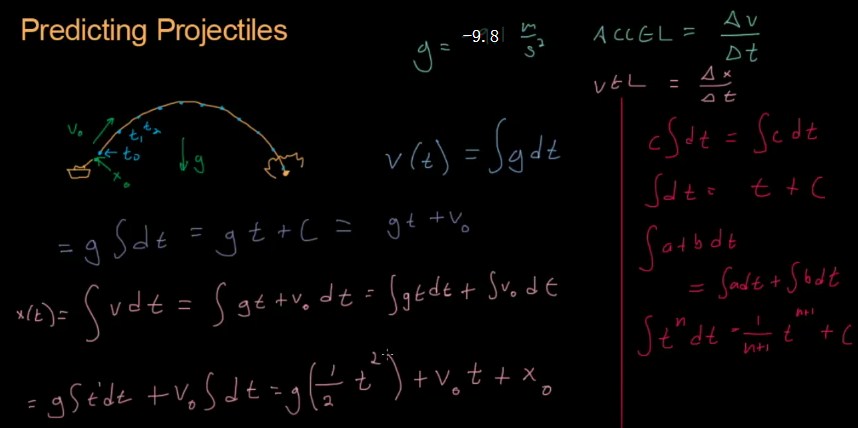
The faster way to get the shortest path. The rule is similar to Dijkstra but we add a factor h (using distance to estimate completion cost).



32. Path finding summary – 66 Code for Game Developers - Graphs and Pathfinding Review



33. Integrals - Math for Game Developers - Predicting Projectiles (Integration) – 69



34. Common Integral Knowledge – Math for Game Developers - Common Integrals 70

