**ASSIGNMENT 3: GAMES CONCEPT**

**BY: TANVI PRAKASH GAVALI**

Document

The Grade I expect:

Above 80 and 90

1 TL-Engine = 1m

**The length of the hover-car in TL-Engine** = 15 TL-Engine units

**The length of the hover-car in metres** = 15 m

**The scale of objects the game (units/metre)**

Isle = 13 units and 13m

Walls = 20 units and 20 m

Checkpoints = 40 units and 40 m

Checkpoint Radius = 1.0f

**The coefficient of the drag** = -0.001f

**The maximum speed of the hover car (without boost) in**

1. Metres/second = 80.55 m/s (approx.)
2. Kilometres/hour = 290 km/hr (approx.)
3. Miles/hour = 180.1976458 miles/hr(approx.)

**The maximum speed of the hover car (with boost) in**

1. Metres/second = 136.11 m/s (approx.)
2. Kilometres/second = 490 km/hr (approx.)
3. Miles/hour = 304.4718842 miles/hr(approx.)

**The maximum thrust force applied (length of thrust vector) in TL-Engine units and scaled to represent acceleration in metre/second/second. –** 0.075 metre/second/second

**A brief explanation of resolving collision between hover car and other objects**

**Sphere-Box Collision (hover car and walls, hover car and checkpoints)**

Use of sphere to box collision – For the sphere to box collision one must need the spheres x and z position and spheres old x and z positions with sphere radius. Also in case of box, one must need x and z box position and its width and depth.

In case of car and checkpoint: The hover car is the sphere and the checkpoint is the box.

And in case of car and walls: the hover car is the sphere and the wall is the box

Creating the calculation of the collisions in a function makes it easier to call as it consumes time for the calculations. Getting the calculated function in an enumeration lets list of possible values it can collide with (left, right, front, back, no side). In the function calculate the boundaries of the square collision box. Get the minimum X, maximum X and minimum Z, maximum Z the boundaries can have from the above positions of the sphere and the box. To find the minimum X subtract the x position of the box by box width/2 and subtract it by sphere’s radius. Similarly,

float minX = boxxPos - boxwidth / 2 - sphererad;

float maxX = boxxPos + boxwidth / 2 + sphererad;

float minZ = boxzPos - boxdepth / 2 - sphererad;

float maxZ = boxzPos + boxdepth / 2 + sphererad;

In the function give a check to see if the sphere is inside the box. If the x position of the sphere is greater than the calculated minX, then the old position of the sphere which is less than the minX will result left side.

if (spherexPos > minX && spherexPos < maxX && spherezPos > minZ && spherezPos < maxZ)

{

if (oldspherexPos < minX) result = leftSide;

else if (oldspherexPos > maxX) result = rightSide;

else if (oldspherezPos < minZ) result = frontSide;

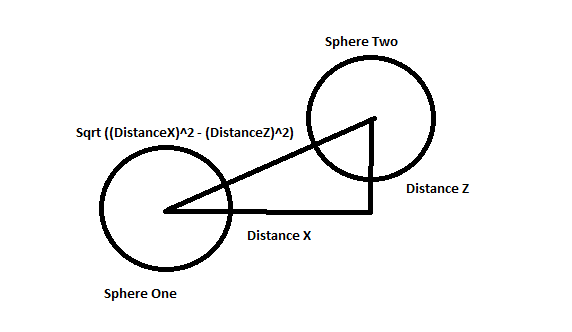
else result = backSide;

}

return(result);

**Sphere to Sphere Collision: (AI hover car and dummy model, hover car and AI hover car, hover car and checkpoint strut, hover car and large tank, hover car and small tanks, hover car and trees)**

For the sphere to sphere collision one need x and z position of both the spheres and the sphere radius for both. Use of functions makes it easier to call the calculation and implement the collisions where ever necessary. Calculate the X distance between x position of sphere two and sphere one by subtracting them, similarly calculate the Z distance. Use Pythagoras theorem to get the distance between the middle of the two circles (the length of the hypotenuse), the distance X and distance Z are the horizontal and vertical distances between the two spheres. If the hypotenuse is smaller than the addition of the both the sphere radius, then it is considered to be a collision.

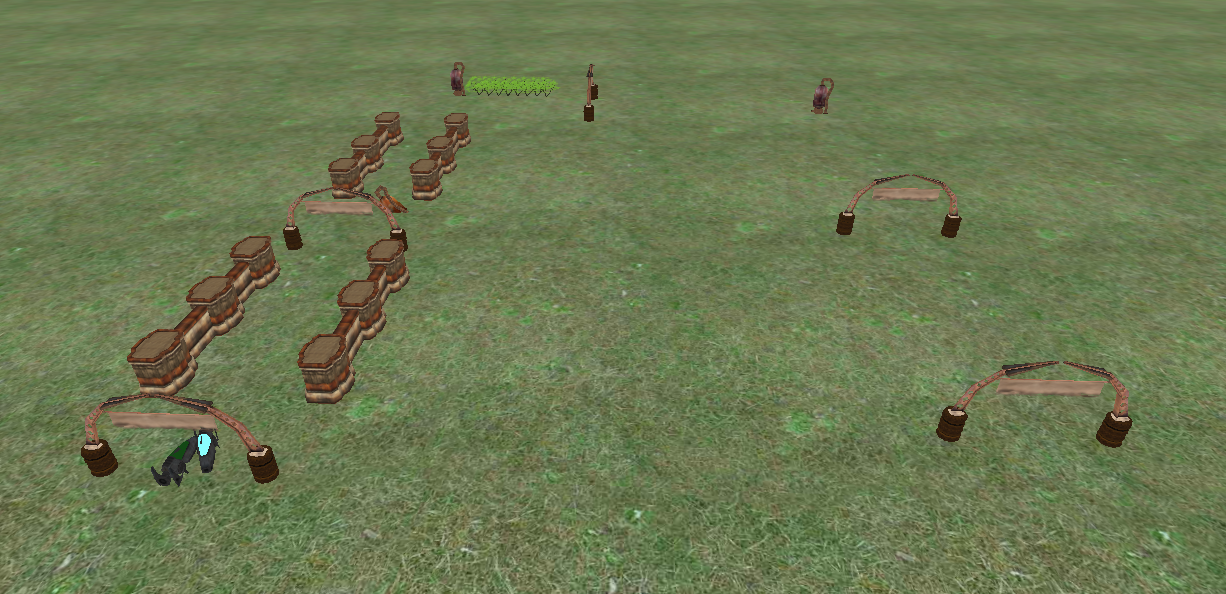


float distX = sphere2xPos - sphere1xPos;

float dust = sphere2zPos - sphere1zPos;

float distance = sqrt(distX\*distX + distZ \* distZ);

return (distance < (sphere1rad + sphere2rad));

Scale Race Map:

1 TL-Engine is equal to 1 metre so the calculation would be the same

The length of the map approx. 330 metres.