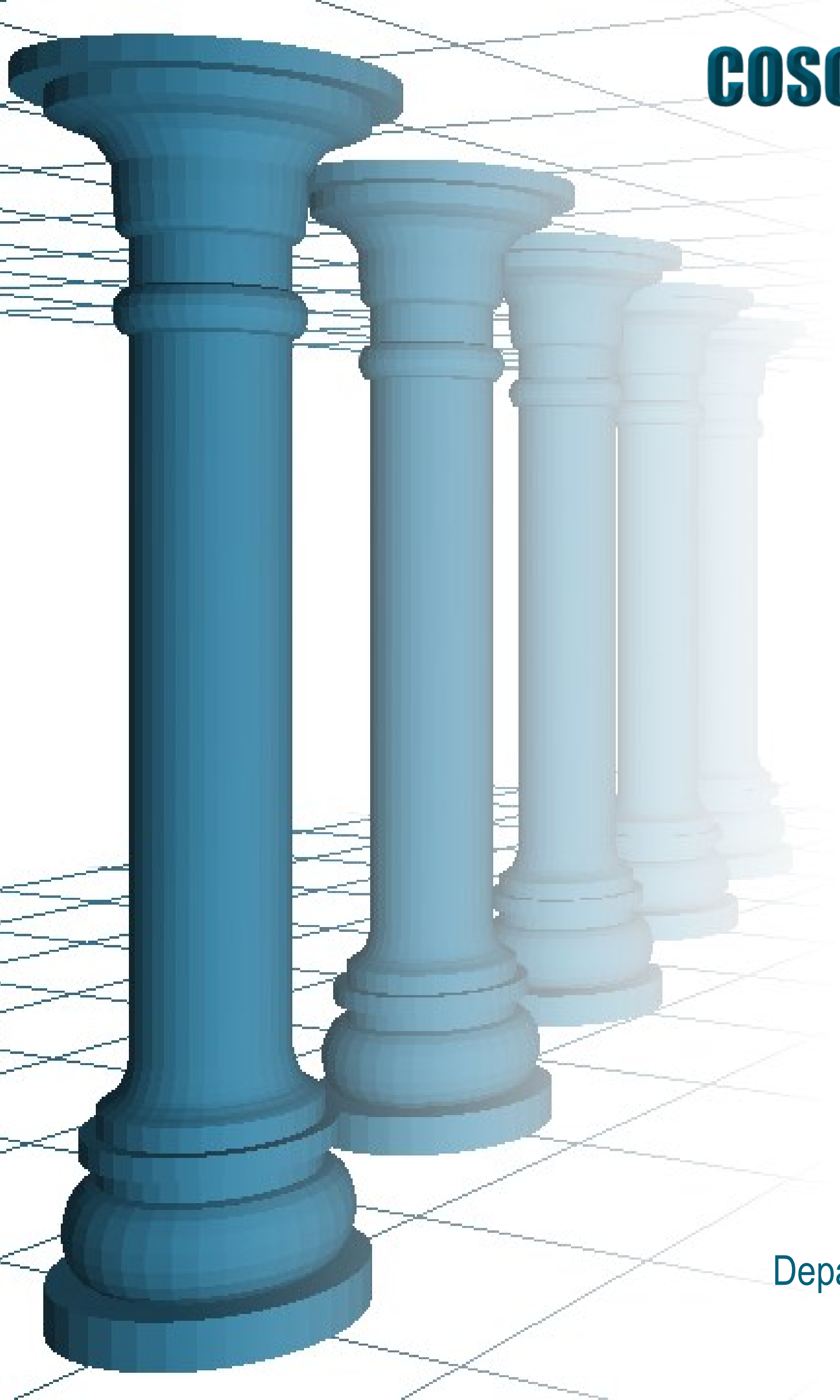


COSC363 Computer Graphics

Assignment-1

Imagination

↓

Creativity

↓

OpenGL

↓

COSC363 Assignment ☺

R. Mukundan (mukundan@canterbury.ac.nz)
Department of Computer Science and Software Engineering
University of Canterbury, New Zealand.

Assignment-1

- Due: 11:55pm, 8 April 2022.
- Maximum Marks: 20
- Assignment handout available on Learn page.
- Use only C/C++ programming language and OpenGL API
- Not a group project. Your submission must represent your own individual work
- Students are encouraged to discuss assignment related problems using course forum. However, code segments or any part of your assignment submission should not be posted on Learn.

Minimum Requirements (8 Marks)

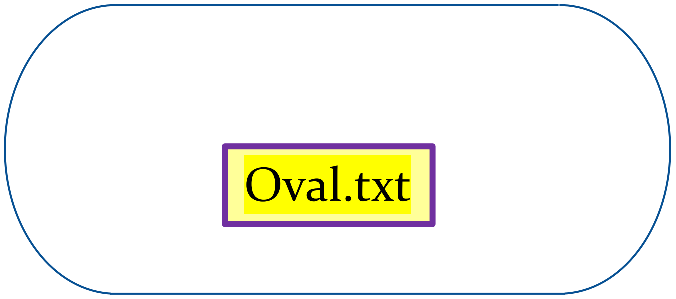
- A railway track and a model of a train (Lab02)
- Models of a railway station and a tunnel.
- Train stopping at the track in each lap.
- Scene view navigation (Camera movements).

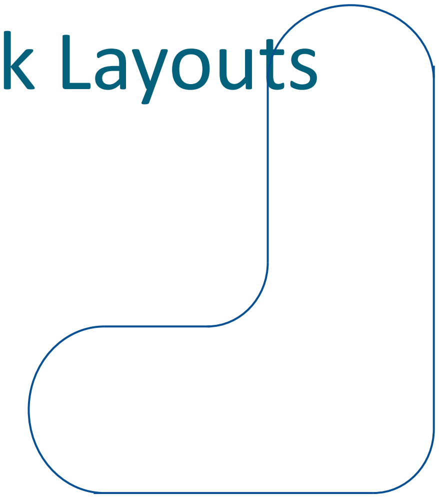
Extra Features (7 Marks)

- Non-circular loop tracks
- Better models for locomotive and wagons, Textures, Shape/animation features, Particle systems, Misalignment corrections.
- Animated scene objects: Barrier arm, vehicles, signalling lights
- View modes: Cab view, station view

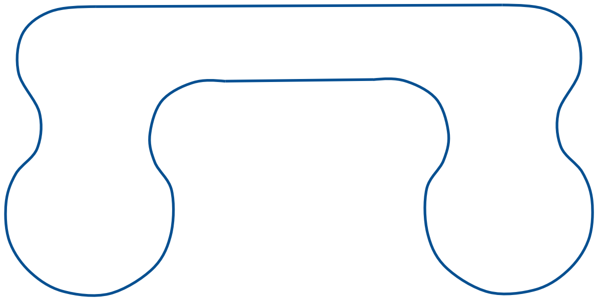
Common Non-Circular Track Layouts

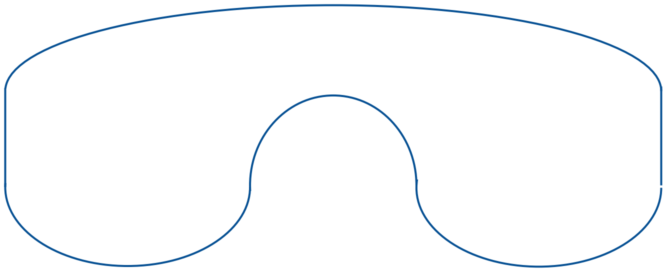
Oval



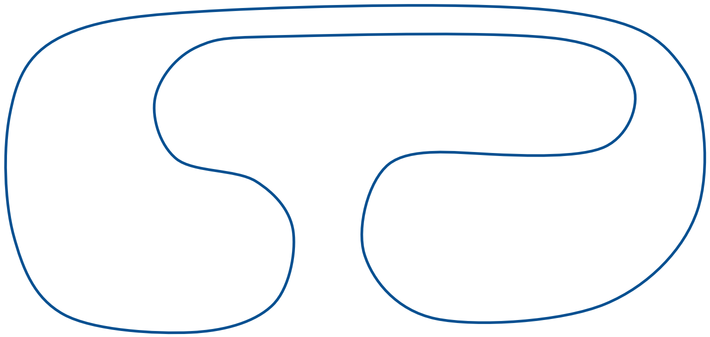


Dog Bone





Folded
Dog Bone

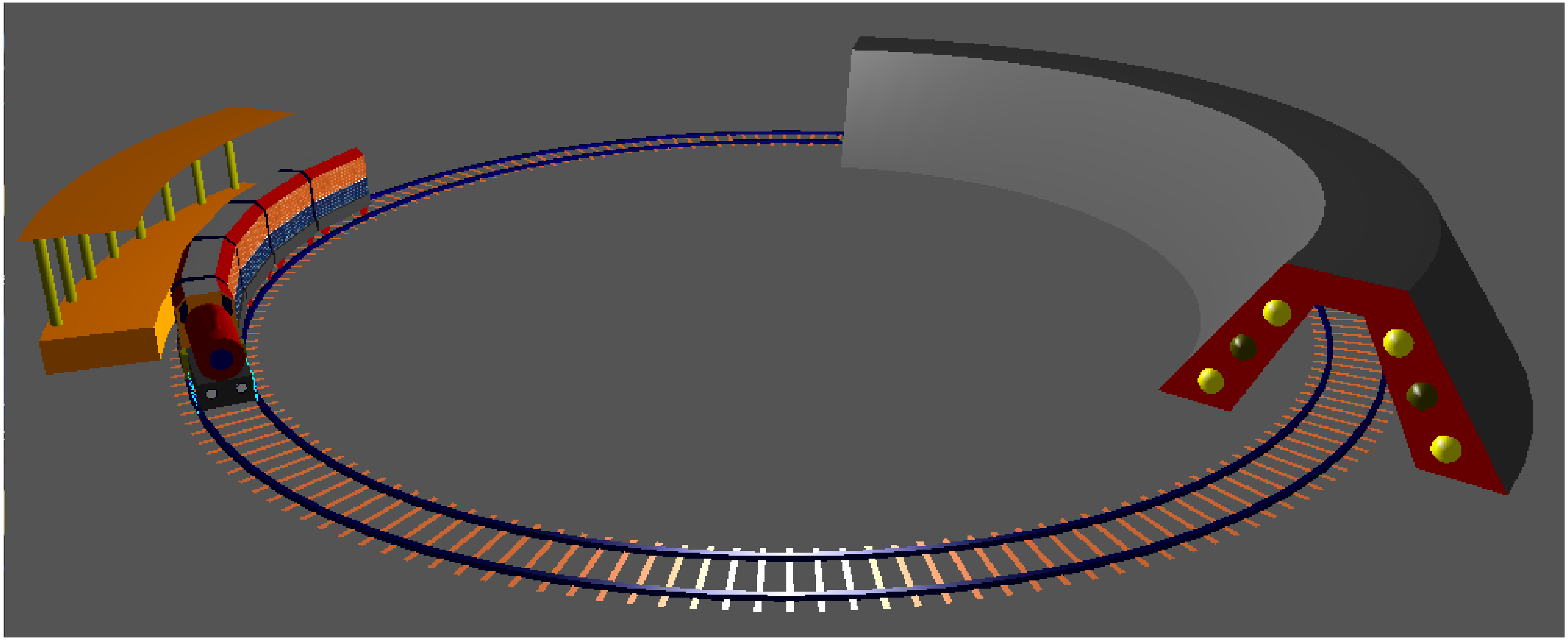


Avoid sharp bends!

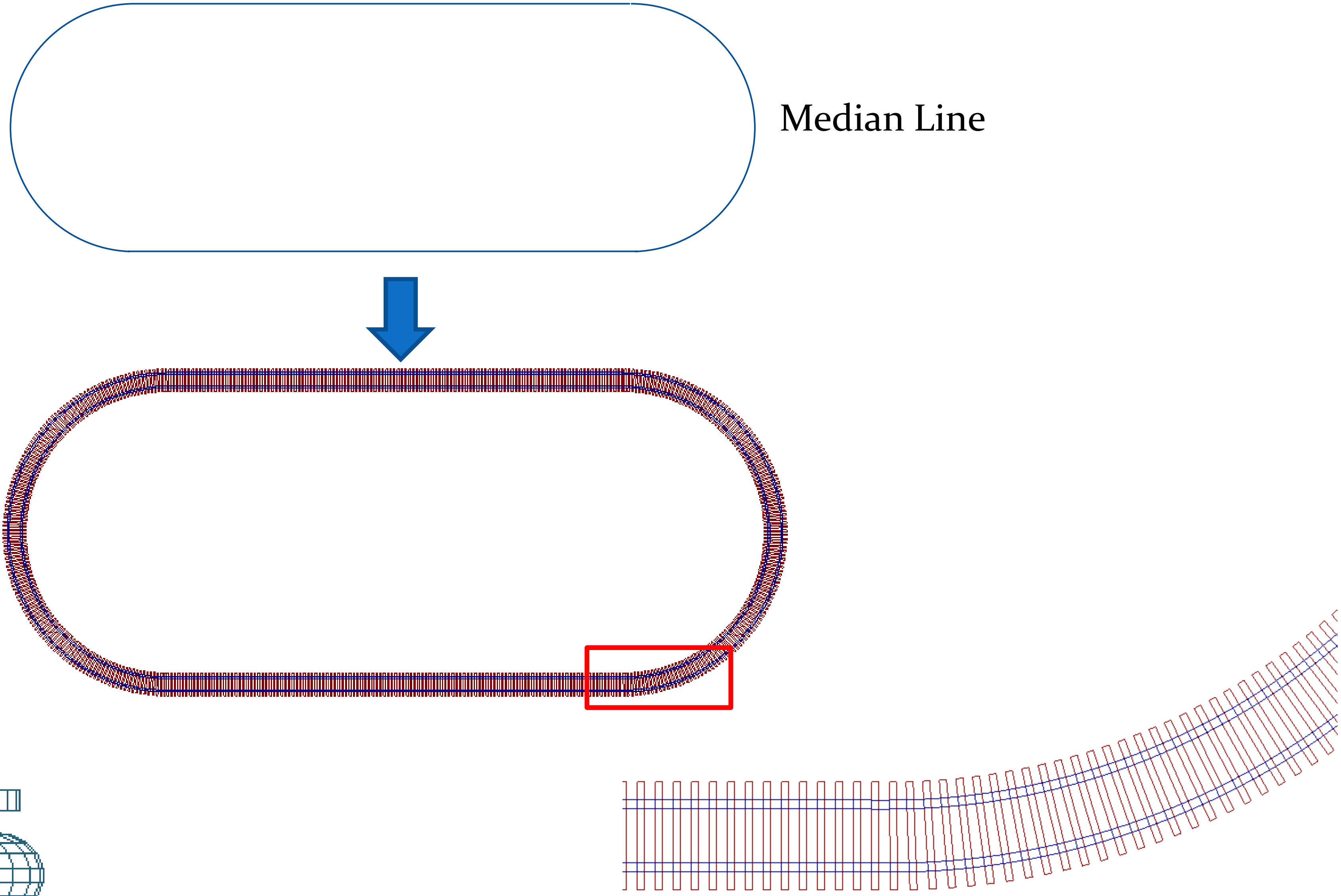
Track Options

- Circular (same as Lab 2): max 1 mark
- Oval (using the supplied file oval.txt): max 2 marks
- Your own design (please provide details in report): max 3 marks

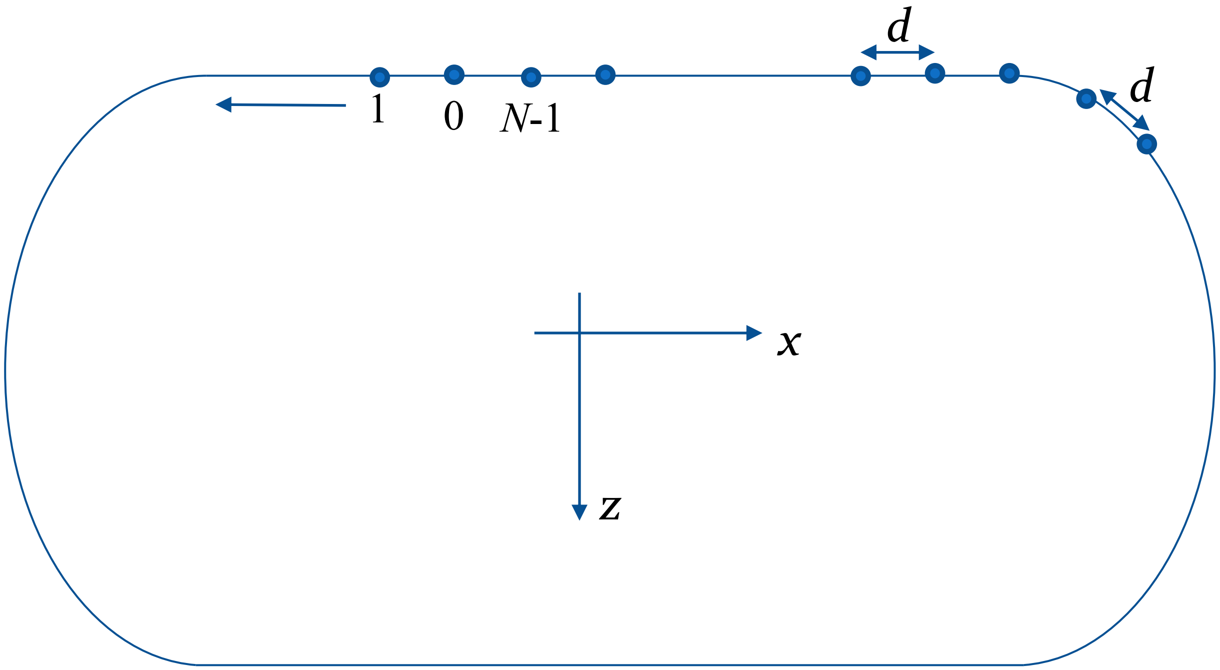
Circular Track (Example)



Step 1: Non-Circular Track Construction

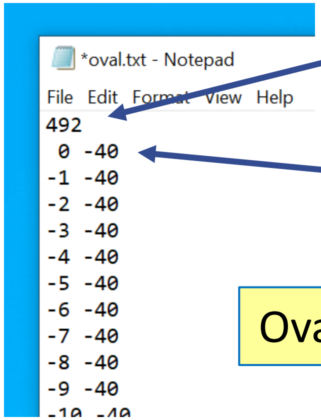


Median Line



$$x[i],\ \ z[i],$$
$$i = \ 0 \ \dots \ N\text{-}1$$

- Obtain the coordinates (*x*, *z*) of points along the median line at constant intervals *d*.



```
492
 0 -40
-1 -40
-2 -40
-3 -40
-4 -40
-5 -40
-6 -40
-7 -40
-8 -40
-9 -40
-10 -40
```

Total number of points

x[0] z[0]

Oval.txt

Useful tools

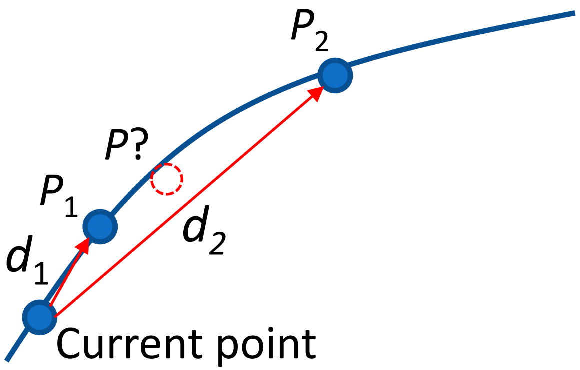
Virtual graph paper

- https://virtual-graph-paper.com/

Extraction of data points from curves

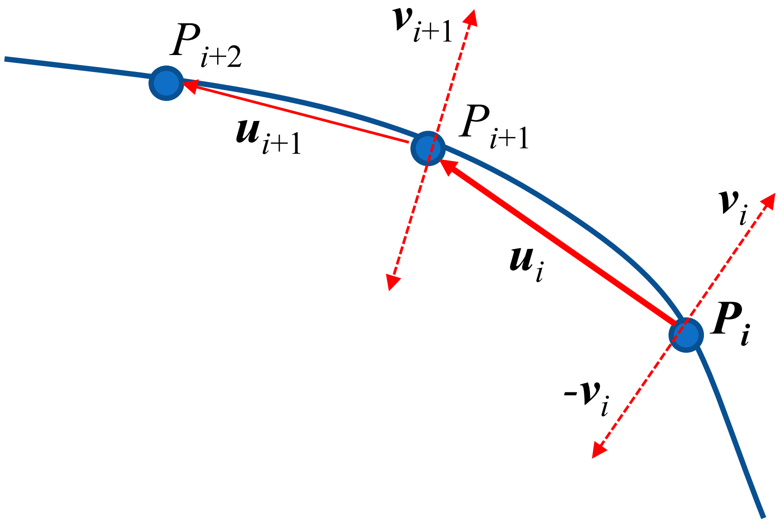
- https://automeris.io/WebPlotDigitizer/

- Computing equidistant points: We require a point at a distance d from the current point. The points are unevenly distributed with P_1 at distance d_1 and P_2 at distance d_2 ($d_1 < d < d_2$)



$$P = P_1 + \left(\frac{d - d_1}{d_2 - d_1}\right)(P_2 - P_1)$$

Median Line Vectors



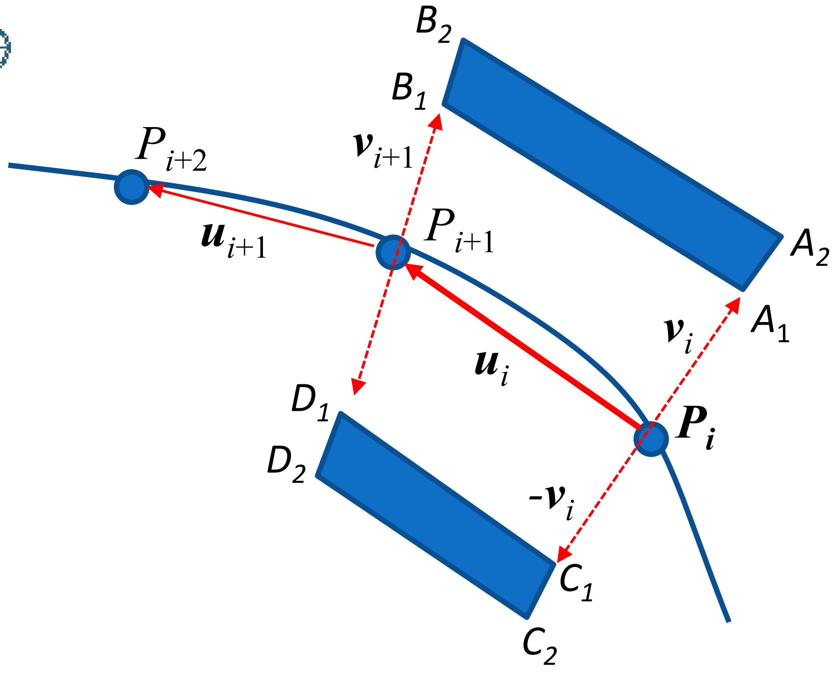
$P_i, \; P_{i+1}, \; P_{i+2}$ are three consecutive points along the median.

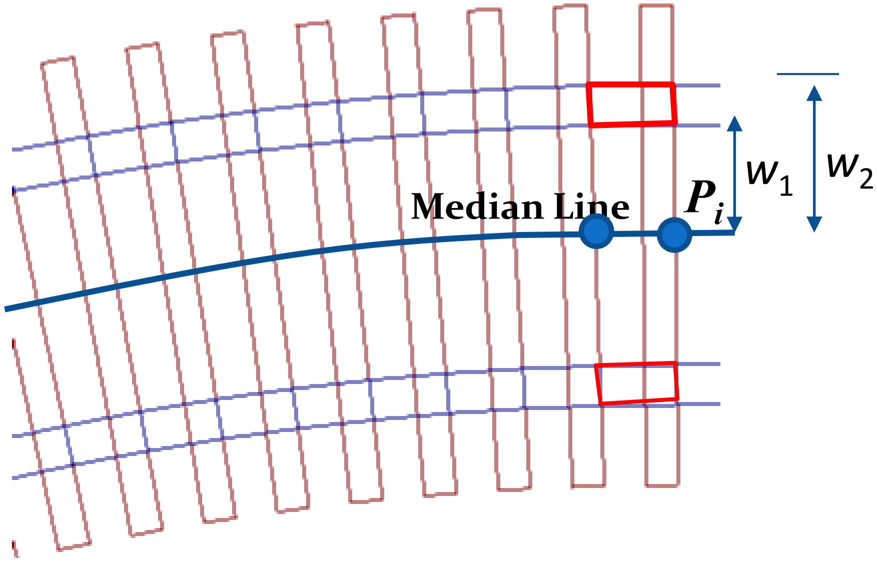
$\boldsymbol{u}_i$ is a **unit** vector from P_i to P_{i+1} .

$\boldsymbol{v}_i$ is a **unit** vector $\perp$ to $\boldsymbol{u}_i$.

If $\boldsymbol{u}_i = (u_x, u_z)$, then $\boldsymbol{v}_i = (u_z, -u_x)$.

Track Segments

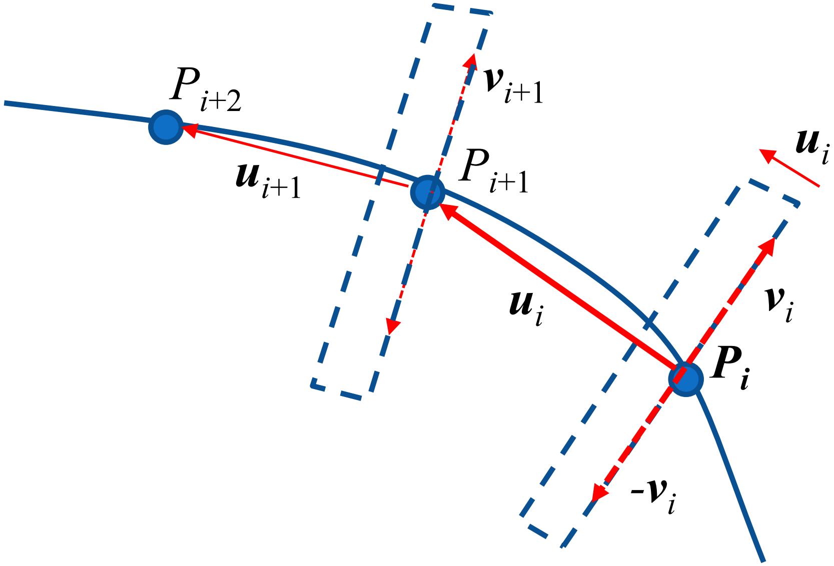


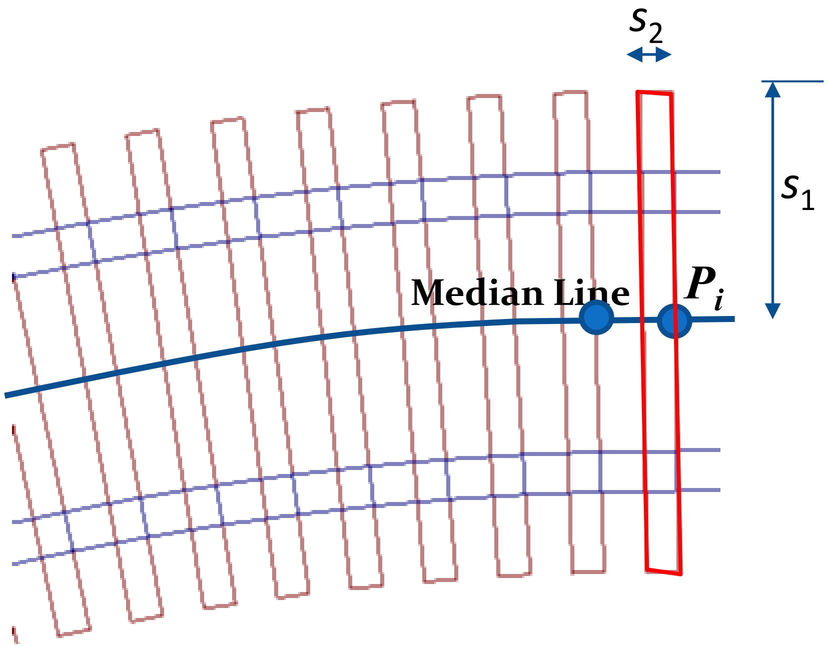


$A_1 = P_i + \mathbf{v}_i\, w_1$; $\qquad A_2 = P_i + \mathbf{v}_i\, w_2$

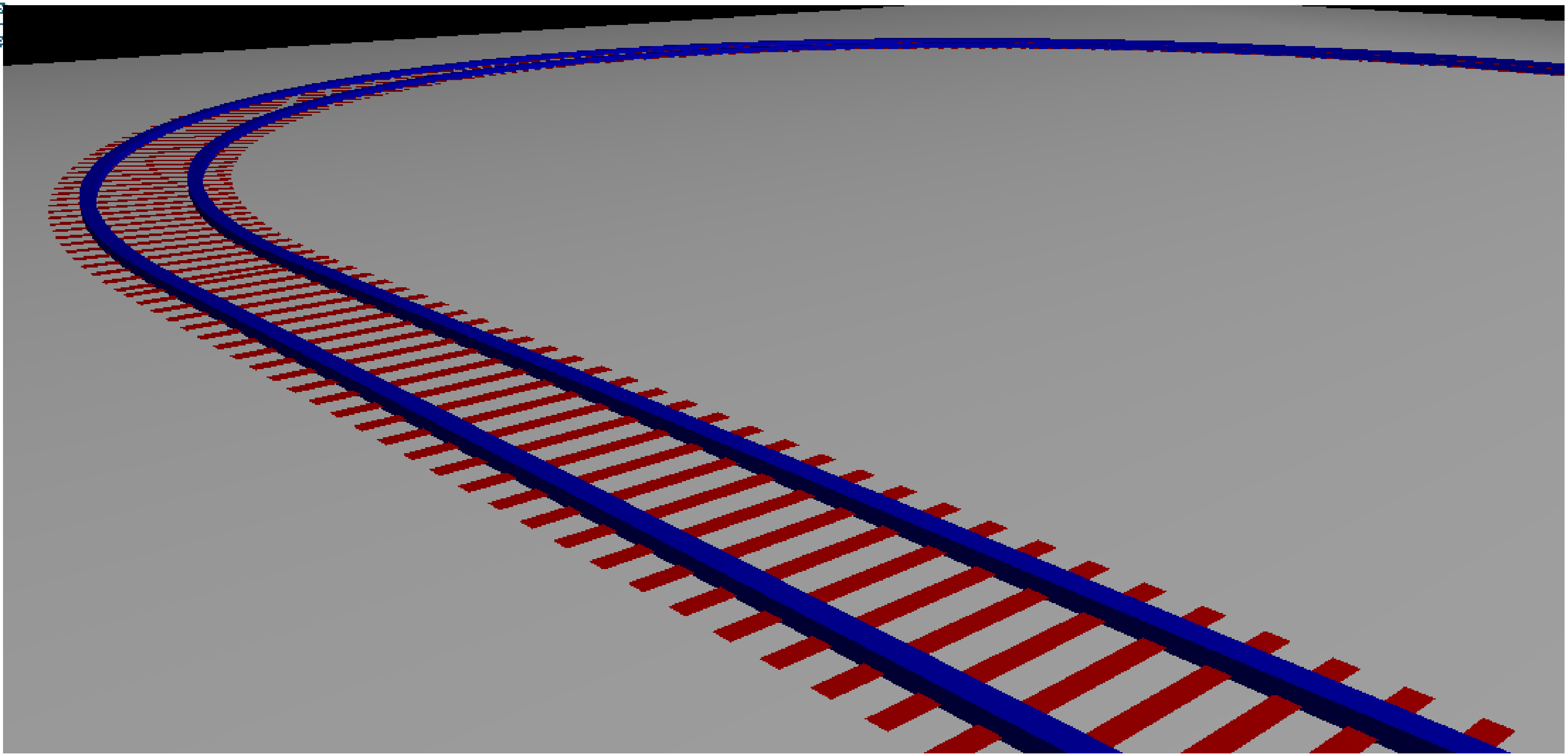
$B_1 = P_{i+1} + \mathbf{v}_{i+1}\, w_1$; $\qquad B_2 = P_{i+1} + \mathbf{v}_{i+1}\, w_2$ etc.

Railway Sleepers



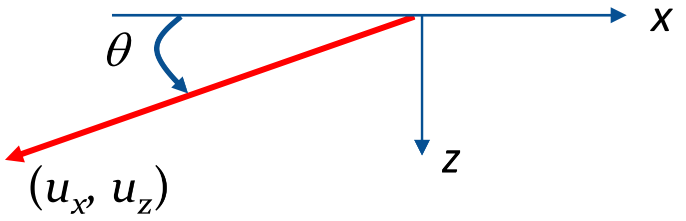


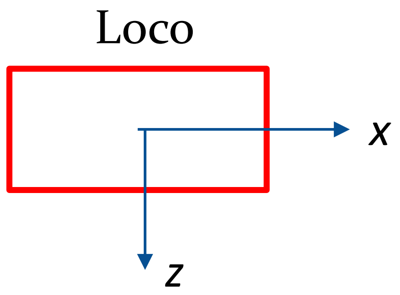
Non-Circular Tracks



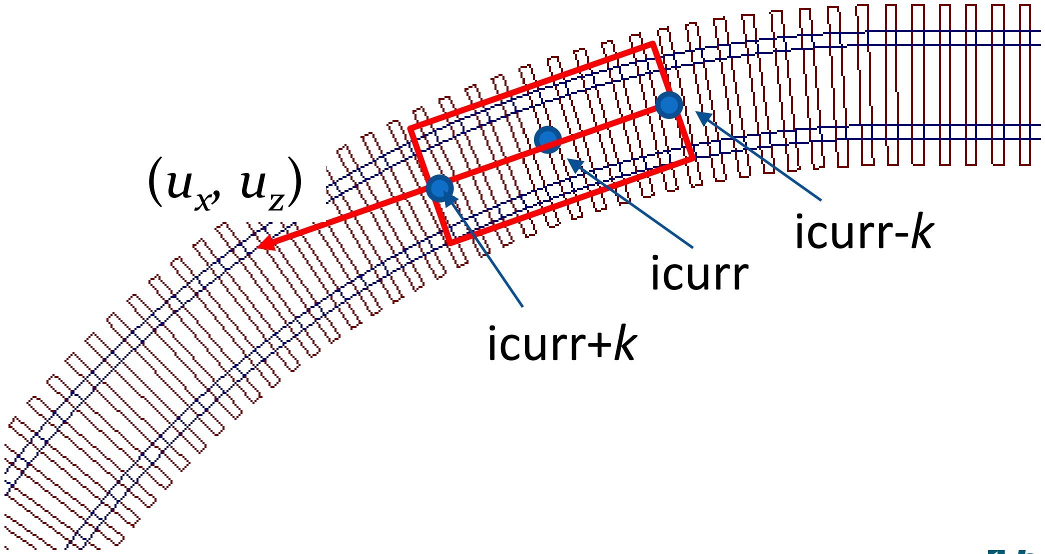
Positioning Models on the Track

- “**icurr**” is the index of the current position of the locomotive. Rotate the model by angle θ about the *y*-axis and translate it to (*x*[icurr], y_{track}, *z*[icurr]).

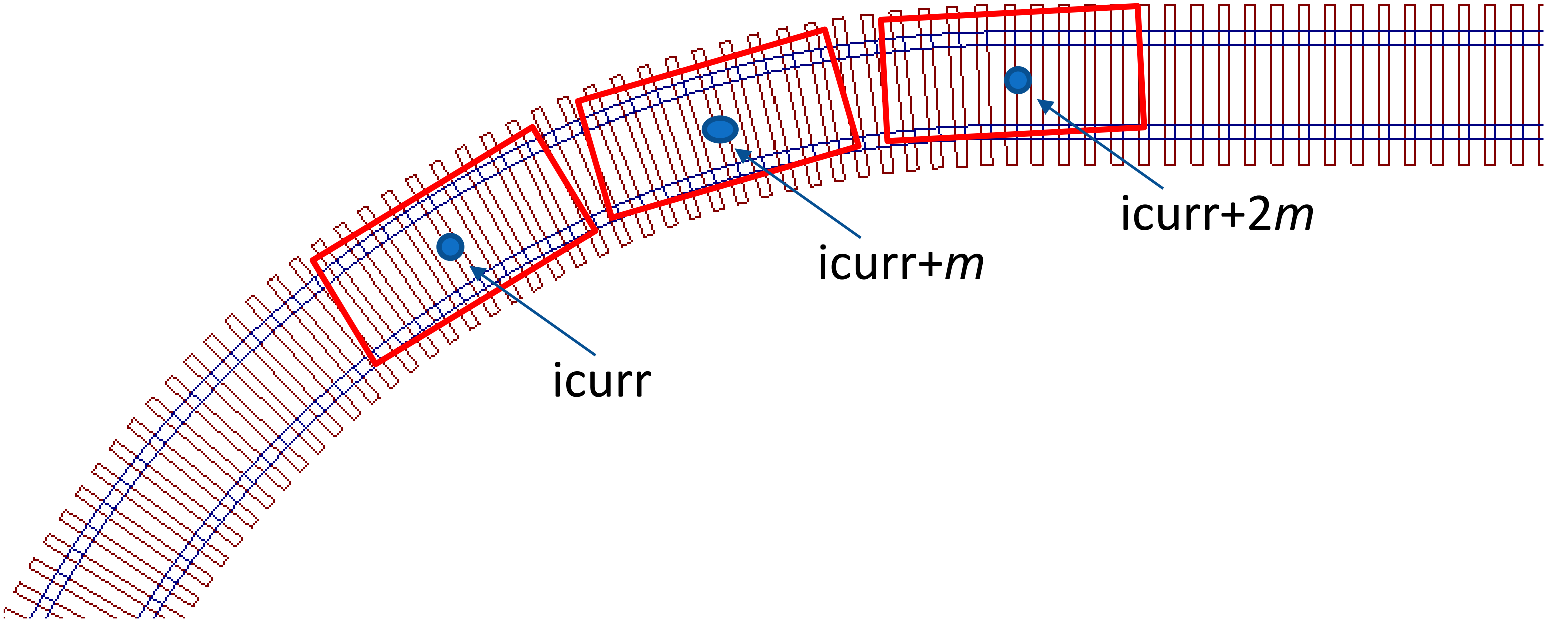




$$\theta = \text{atan2}(u_z\,,\, -u_x)$$



Positioning Models on the Track

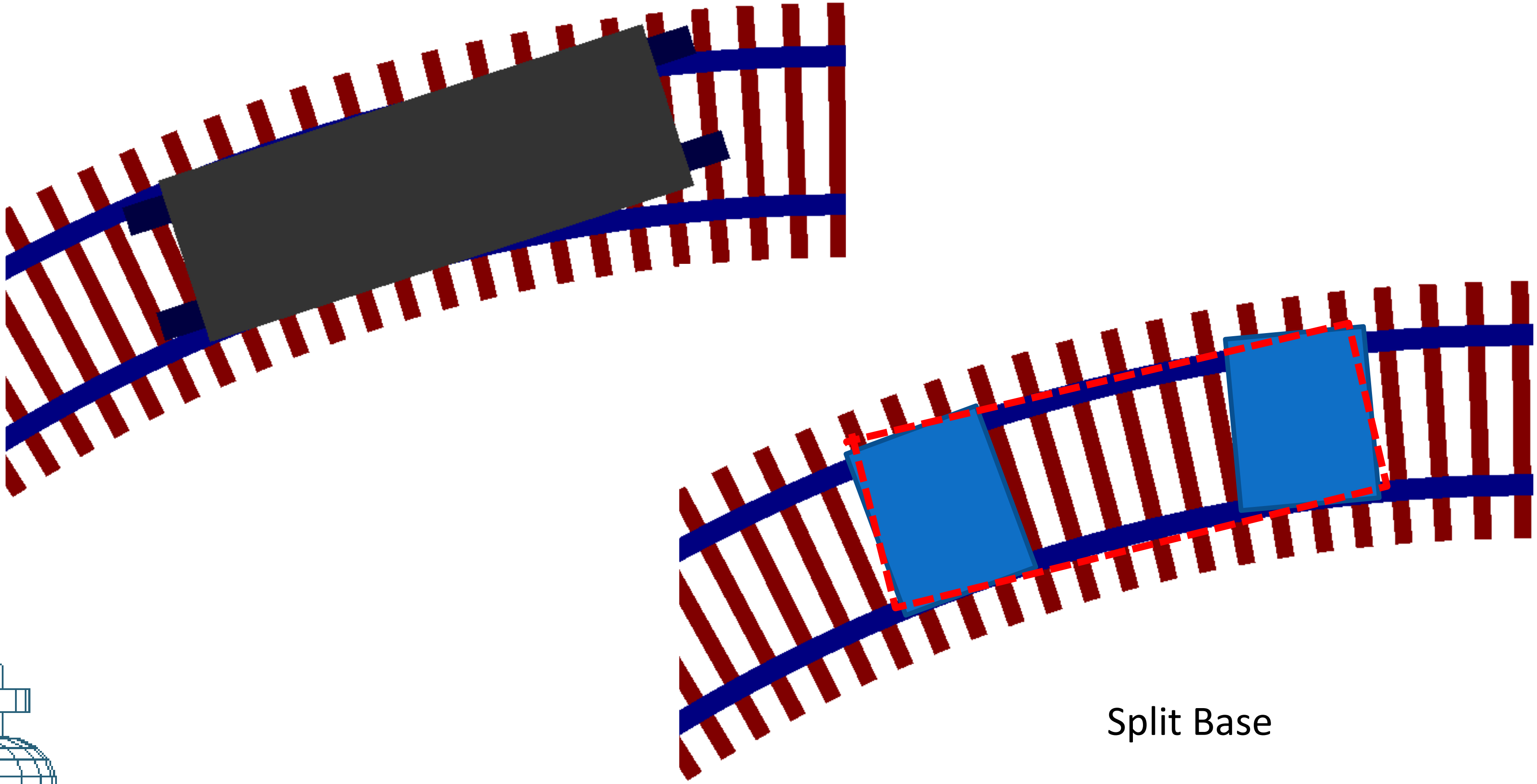


Animating the Train Model

- Simply increment “icurr” !!
- Note that the index wraps around to 0.
- Try to get a smooth animation
  - The train should not appear to be moving in discrete steps
  - If necessary, interpolate between consecutive positions and orientations.

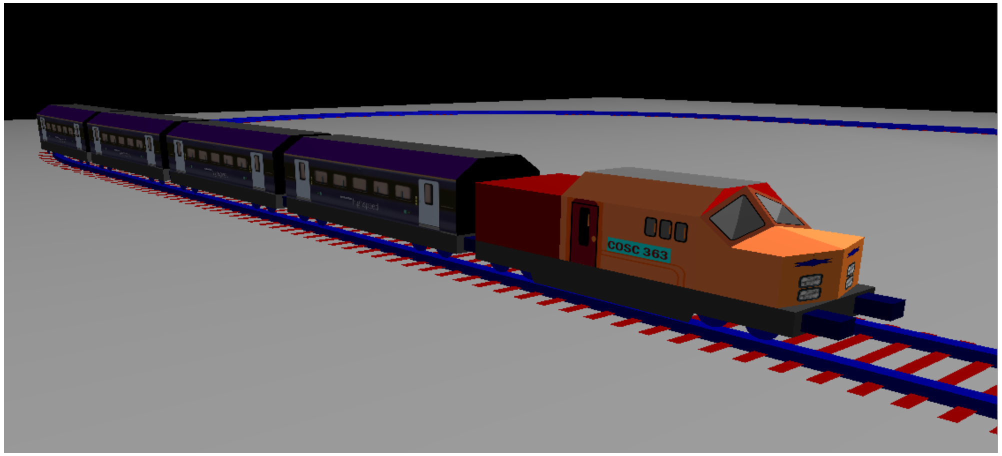
Alignment Errors

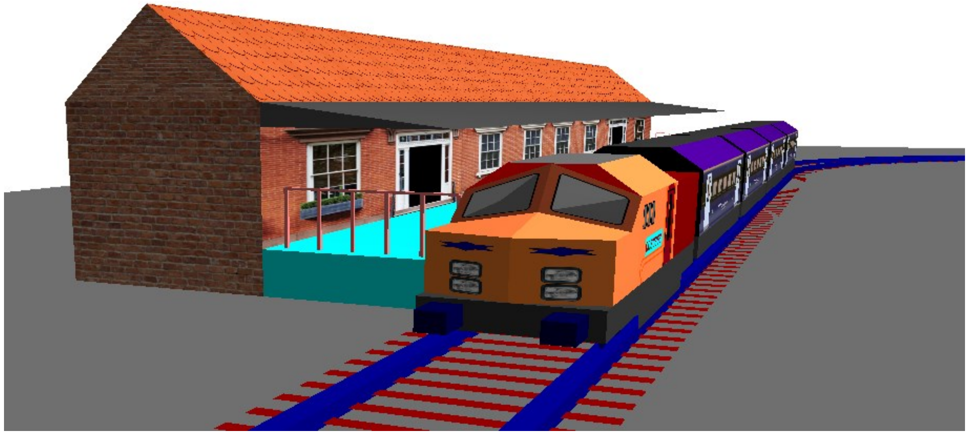
- You will not lose marks for small, not easily noticeable alignment issues.
- You will get extra marks for correcting alignment errors!



Texture Mapping

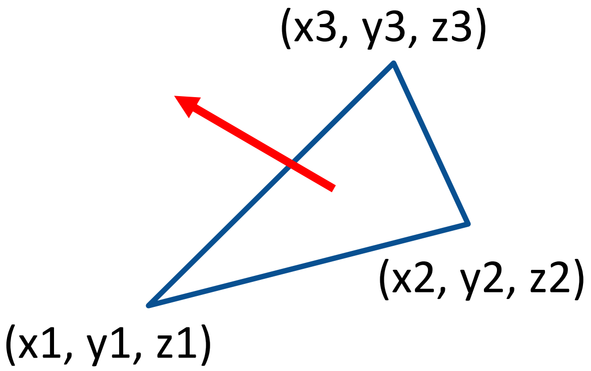
- The process of mapping textures to models will be discussed in next lecture (Mar 17) and this week’s lab (Lab 3).
- Note: Textures cannot be mapped to glutSolidCube.



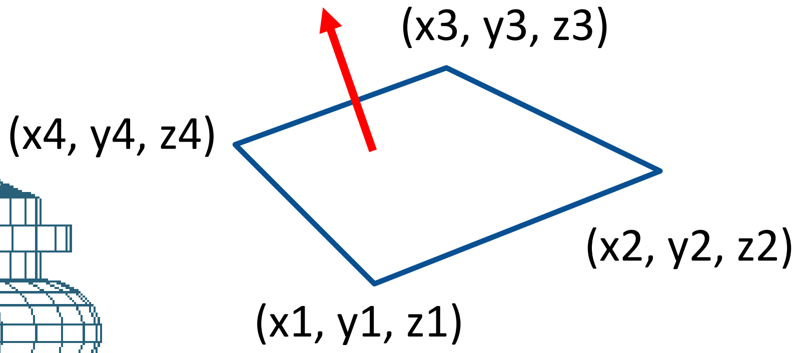


Object Modelling

- The lecture on Mar 21 will cover methods for object modelling and particle system generation.
- When creating primitives with arbitrary orientation, you will require a function to compute the normal vector.



```
glBegin(GL_TRIANGLES);
  normal(x1,y1,z1, x2,y2,z2, x3,y3,z3);
  glVertex3f(x1, y1, z1);
  glVertex3f(x2, y2, z2);
  glVertex3f(x3, y3, z3);
glEnd();
```



```
glBegin(GL_QUADS);
  normal(x1,y1,z1, x2,y2,z2, x3,y3,z3);
  glVertex3f(x1, y1, z1);
  glVertex3f(x2, y2, z2);
  glVertex3f(x3, y3, z3);
  glVertex3f(x4, y4, z4);
glEnd();
```

Normal Vector

- See also lecture slide [3]-19

```
void normal( float x1, float y1, float z1,
             float x2, float y2, float z2,
             float x3, float y3, float z3 )
{
   float nx, ny, nz;
   nx = y1*(z2-z3) + y2*(z3-z1) + y3*(z1-z2);
   ny = z1*(x2-x3) + z2*(x3-x1) + z3*(x1-x2);
   nz = x1*(y2-y3) + x2*(y3-y1) + x3*(y1-y2);
   glNormal3f(nx, ny, nz);
}
```

Faversham Express

- This is just a name given to this assignment! You are not required to incorporate any features of this train simulator route in your assignment.

London - Faversham High Speed with Javelin Class 395

https://www.youtube.com/watch?v=ciCt_cfW_Lw

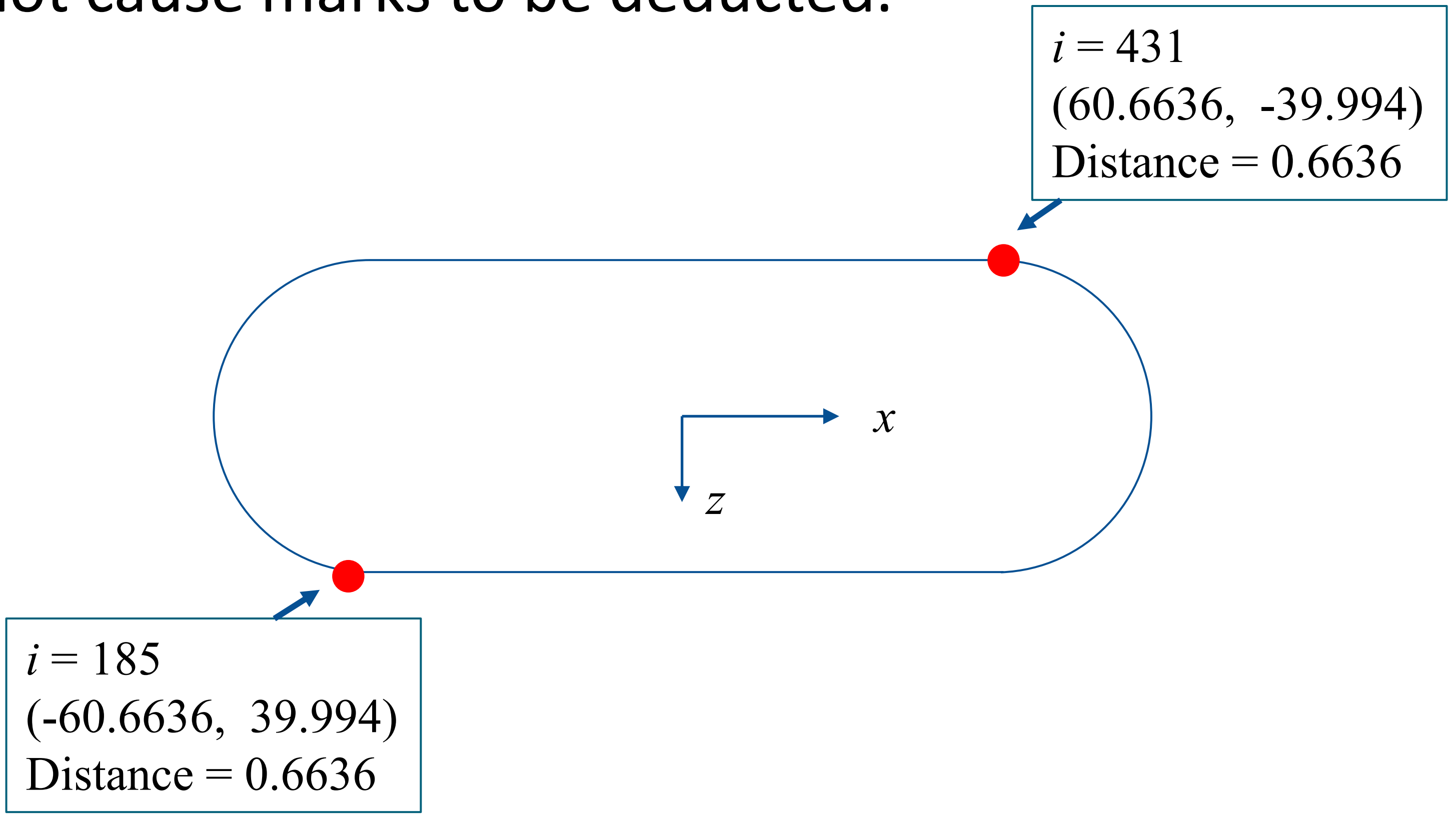
28 Mar

Median Line

- It is not always possible to sample/digitize a closed curve into a set of equidistant points
- Please make sure that the range of variation in distance between consecutive points is not very large. A nearly uniform distribution of points is required for smooth animation.
- Variations in the distance between consecutive points will not cause any major issues with the modelling of tracks (remember to convert vector $\boldsymbol{u}_i$ on Slide 11 to a unit vector)
- The distance between points need not be 1 unit.

Oval.txt

- There are two points in the data file where the distance between the current point and the next is not equal to 1.
- Slight discontinuities in the animation at these points will not cause marks to be deducted.



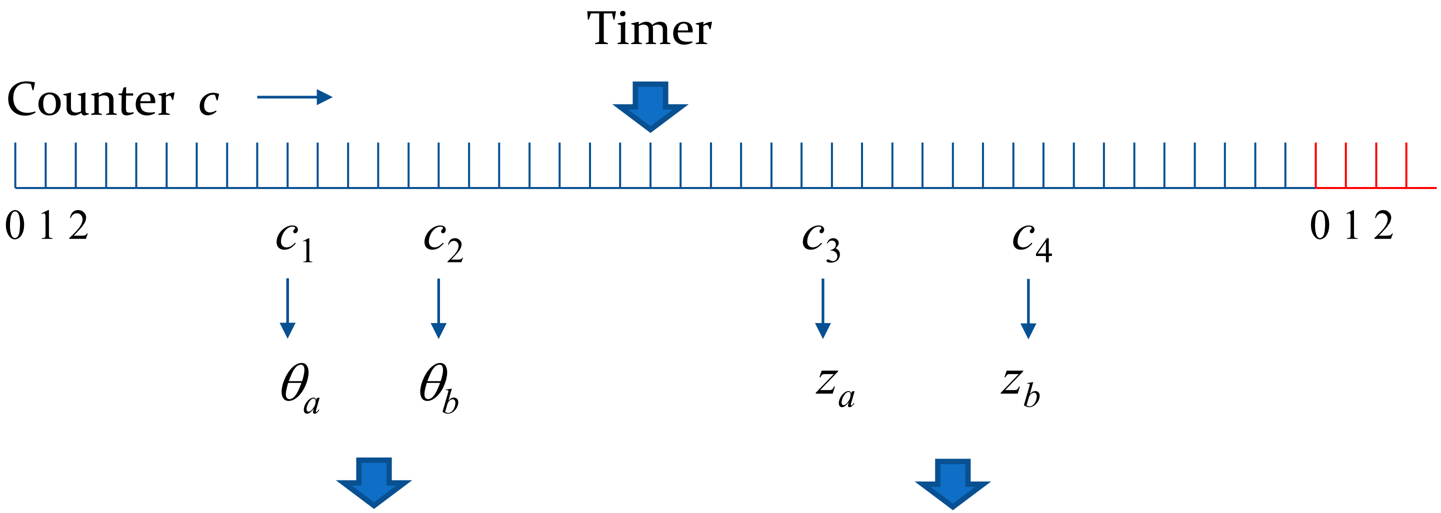
$i = 431$
(60.6636, -39.994)
Distance = 0.6636

$i = 185$
(-60.6636, 39.994)
Distance = 0.6636

Timer callback

- Please do not generate two or more independent (parallel) sequence of timer events.
- Suggested method:
  - Use only a single timer event sequence
  - Use only one timer callback function
  - Define a global `int counter` variable, incremented each time inside the timer callback function
  - Define start and end points of animation sequences based on the values of the counter variable.
  - The counter variable must be reset to 0 when the train completes one lap.

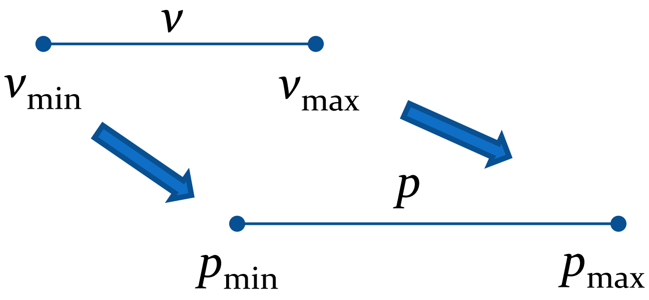
Controlling Multiple Animations



$$\frac{\theta - \theta_a}{\theta_b - \theta_a} = \frac{c - c_1}{c_2 - c_1}$$

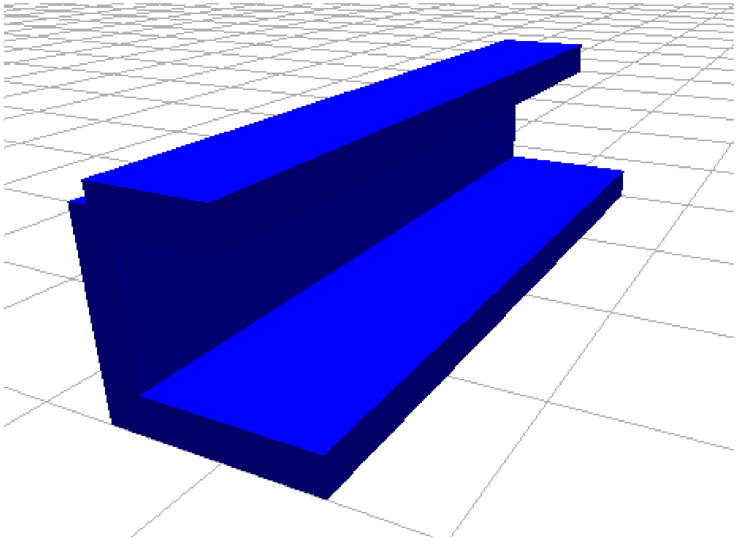
$$\frac{z - z_a}{z_b - z_a} = \frac{c - c_3}{c_4 - c_3}$$

Linear mapping:

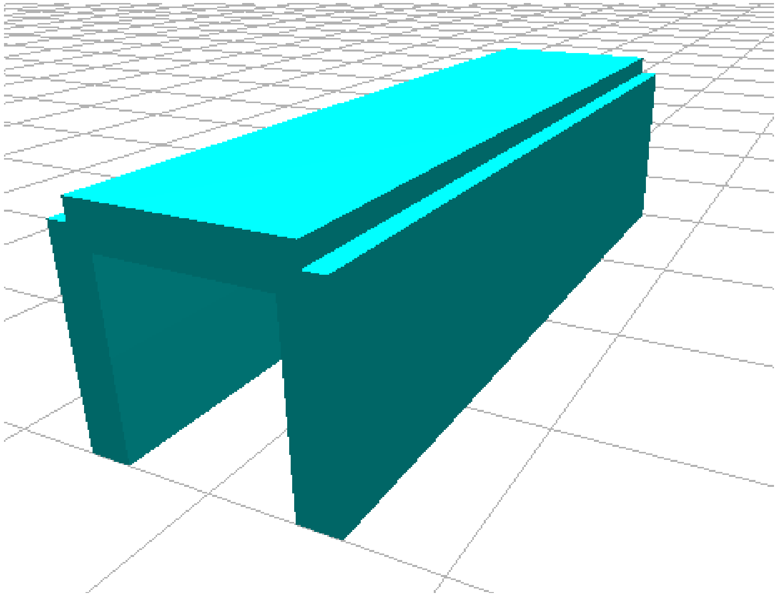


$$\boxed{\frac{v - v_{min}}{v_{max} - v_{min}} = \frac{p - p_{min}}{p_{max} - p_{min}}}$$

Simplistic Models



Railway Station?



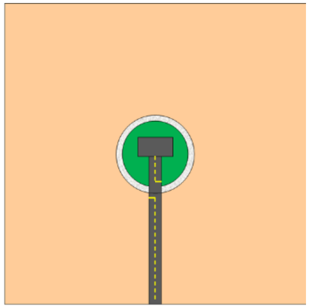
Tunnel?

- Please try to improve the quality of rendering of models constructed using a simple combination of very few primitives or basic shapes.

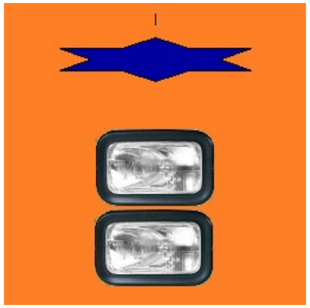
Textures

- May be obtained from easy-to-find images
- May be generated using paint/draw software
- Please do not use copyrighted/watermarked images

Loco



Floor



Lights

Light