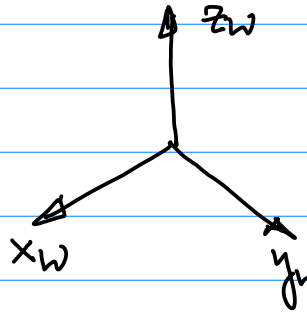
Nov. 17 (Thu).

Midterm Key and Discussion.

Normal Vector Calculation.



$\vec{n}(n_x, n_y, n_z)$ for x_w-y_w plane

Fig.1

$\vec{n} = (0,0,1)$ or $\vec{n} = (0,0,C)$

Generalized Case for Normal Vector

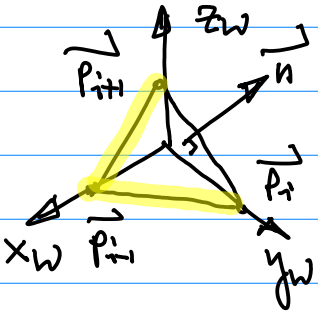


Fig.2

2 vectors $\vec{P}_i - \vec{P}_{i+1}$ and $\vec{P}_{i+1} - \vec{P}_{i-1}$

$$\vec{n}(n_x, n_y, n_z) = (\vec{P}_i - \vec{P}_{i-1}) \times (\vec{P}_{i+1} - \vec{P}_{i-1})$$

Semester Project

1. Formal Presentation and Demo At the end of the Semester.
2. Team Coordinator and Team members need to get together Set Tasks
3. Dec. 7th (No Instruction Day).
   Presentation. 1:30 pm – 3:00pm
4. Requirements (Technical Requirements)
   See Handout (To Be Posted on CANVAS)

a. Rotating Squares on the Road

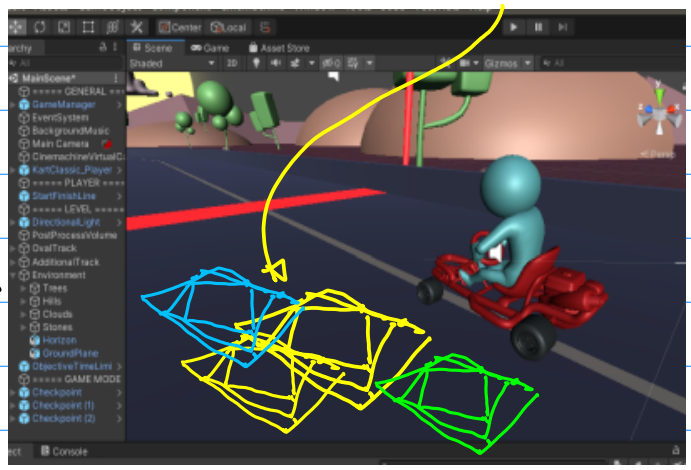


Fig.3

Or,

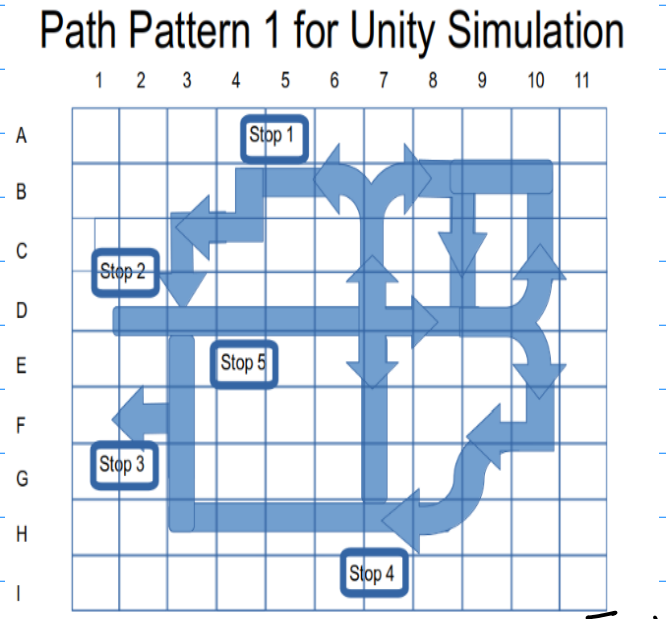
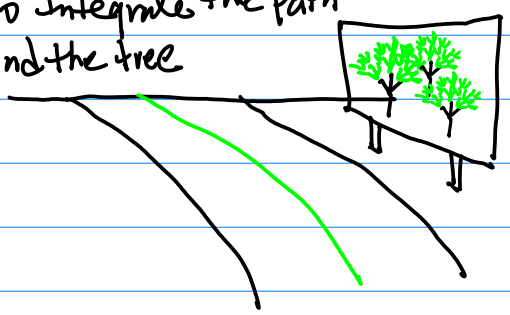


Fig.4

b. To Integrate the path and the tree



c. Video Display

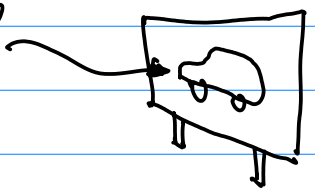


Fig5

1. Need Capability of Drawing A Line in Unity Enviroment → 3D Space #C Script.

https://docs.unity3d.com › ScriptReference › Debug.Dr...

Scripting API: Debug.DrawLine - Unity - Manual

Draws a **line** between specified start and end points. The **line** will be of the editor when the game is running and the gizmo **drawing** is ...

Color: Color of the line — End: Point in world

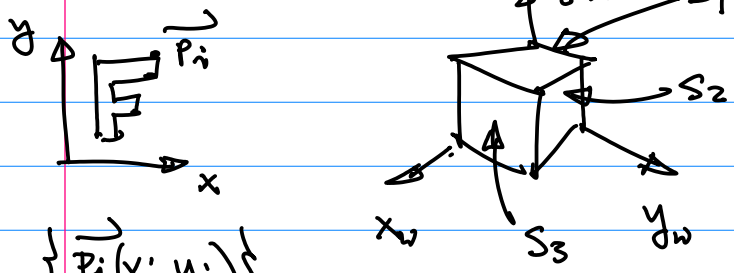
Duration: How long the line should be visible ... — DepthTest: Should t

Handles.DrawLine · Debug.DrawLine · Debug

https://docs.unity3d.com/ScriptReference/Debug.DrawLine.html

https://answers.unity.com/questions/8338/how-to-draw-a-line-using-script.html

2. LINEAR Decoration Algorithm.



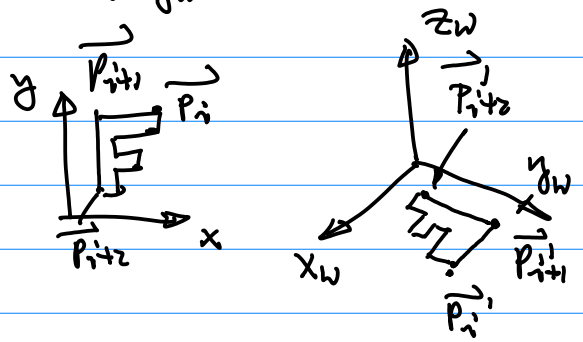
$\{\vec{P_i}(x_i, y_i)\}$ In 2D → Decorate 3D Cube

Decorate Surface S_1.

Step1. Define 2D Graphics Pattern $\{\vec{P_i}\}$ (like Letter "F")

Note: You have Squares, & Trees.

Step2. Move this 2D pattern into x_w-y_w-z_w World Coordinate



$$\begin{cases} x'_i = x_i & \checkmark \\ y'_i = y_i & \checkmark \\ z'_i = C \end{cases} \quad \cdots (1)$$

Decorate S_3.

Idea: Project S_3 plane to y_w-z_w.

Right Principle Define

y_w — Ind.

z_w — Func.

Match Ind → Ind (3d) (2d)

Func → Func (3d) (2d)

$$\begin{cases} y'_i = x_i \\ z'_i = y_i \\ x'_i = C \end{cases} \quad \cdots (2)$$

Decorate S_2

$$\begin{cases} z'_i = x_i \\ x'_i = y_i \\ y'_i = C \end{cases} \quad \cdots (3)$$

Group II Classes

Group II classes are those classes which meet TR, T, R, TWR, MTR, TRF, MTRF, MTWR, TWRF, RF, TF.

Regular Class Start Times	Final Examination Days	Final Examination Times
7:00 through 8:25 AM	Wednesday, December 8	7:15-9:30 AM
8:30 through 9:25 AM	Friday, December 10	7:15-9:30 AM
9:30 through 10:25 AM	Tuesday, December 14	9:45 AM-12:00 PM
10:30 through 11:25 AM	Thursday, December 9	9:45 AM-12:00 PM
11:30 AM through 12:25 PM	Monday, December 13	9:45 AM-12:00 PM
12:30 through 1:25 PM	Wednesday, December 8	12:15-2:30 PM
1:30 through 2:25 PM	Friday, December 10	12:15-2:30 PM
2:30 through 3:25 PM	Tuesday, December 14	2:45-5:00 PM
3:30 through 4:25 PM*	Thursday, December 9	2:45-5:00 PM
4:30* through 5:25 PM*	Monday, December 13	2:45-5:00 PM

Dec. 2nd (Thursday)

1° Final Exam Schedule

Dec. 9 (Th) 2:45–5:00pm.

Same Format as the midterm plus Need to Run Programs, Then take Screen Shots. (jpg. png) → On-Line tool to convert to pdf

Integrate the pdf file of these photos with your paper to form One pdf file.

2° Team Presentation Requirement

Ref (Requirements for the project see github of the class)

2021F-7-homework6-unity-office-karting-ML-2021-10-7.pdf

Another Related Document for the Semester-End project

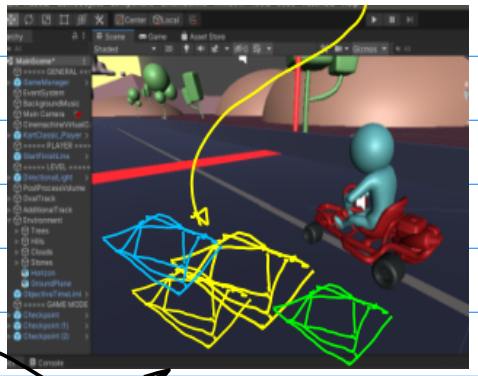
2021F-7c-semester-unity-office-karting-ML-2021-11-30.pdf

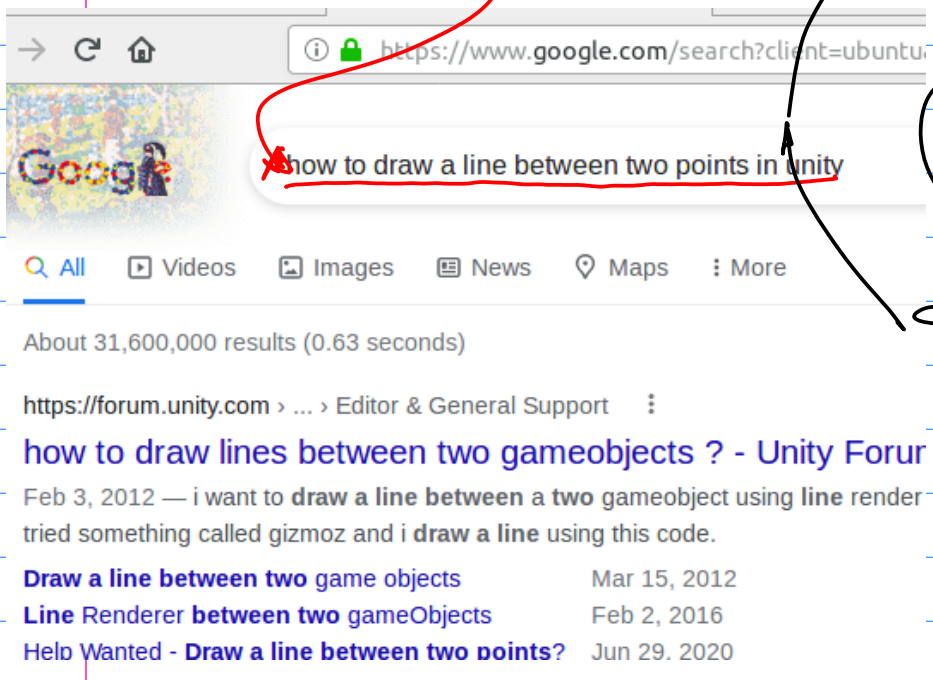
(1) Must integrate the screen saver program rotating squares and trees in the Unity environment;
(2) Must integrate customer coding for the shadow computation;
(3) The layout of the project must import the following driving path as shown in Figure 1.
(4) The implementation of the project is based on the integration of Roll-a-Ball game with import of the motor bike from the Karting game.
(5) with Video display on a virtual screen.

Note: Steps for this task:

Step1. Open & Run Unity "Karting" Game;

Step2. To Be Able to Draw a Line(s) in the Unity Environment.





$\vec{P}_i^{j+1} = \vec{P}_i^j + \lambda(\vec{P}_{i+1}^j - \vec{P}_i^j)$

(See previous discussion)

$\vec{P}_i^{j+1}(x_i^{j+1}, y_i^{j+1})$ etc.

Decoration Algorithm on x_w-y_w plane, Eqn

$$\begin{cases} x_i' = x_i \\ y_i' = y_i \\ z_i' = c \quad (\text{Let } c=0) \end{cases}$$

3D or 2D Lines

If you want a line in 3D space, try creating a LineI
http://rockonflash.wordpress.com/2010/04/17/hov

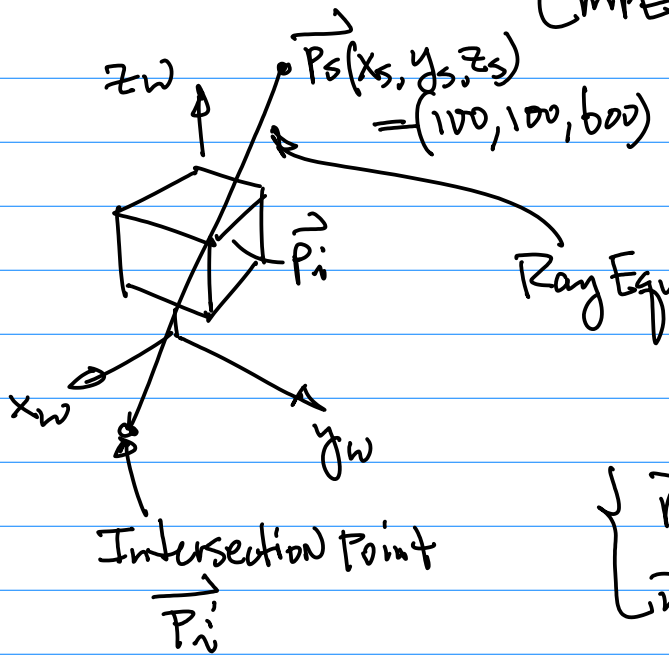
docs here: http://docs.unity3d.com/Documentatic LineRenderer.html

For a 2D line (onGUI), try:

Step3. Linear Decoration Algorithm. (See Notes, pp38)

Note: Requirements (2). Shadow Computation.

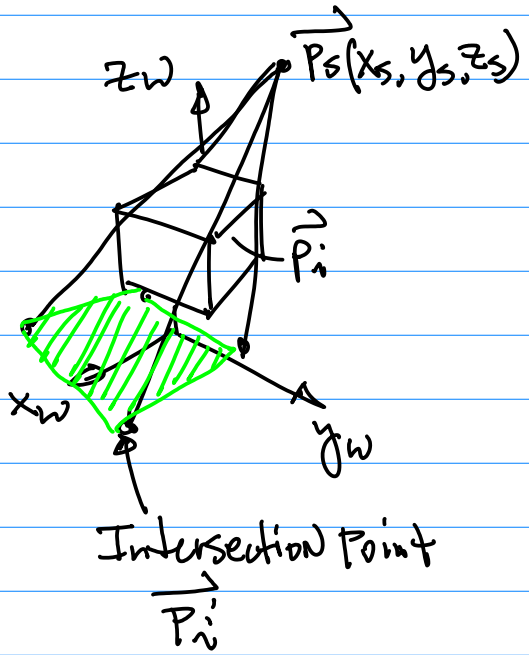
A Simplest Shadow Computation, such as the one with a Cube is fine. find λ of the Ray Equation, then use the Ray equation with this λ to find intersection point.



Ray Equation. $\vec{R} = \vec{P_i} + \lambda(\vec{P_s} - \vec{P_i})$

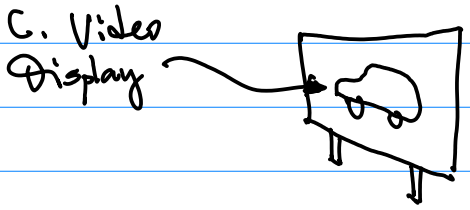
Find λ

$$\begin{cases} \vec{R} = \vec{P_i} + \lambda(\vec{P_s} - \vec{P_i}) \\ \vec{n} \cdot (\vec{v} - \vec{a}) = 0 \end{cases} \rightarrow \vec{n}(0,0,1)$$



Keep track Each intersection point, then plot polygon to Connect them, as Shown in the green

Note: play a video on a plane



Note: Requirement of Importing Video Clips to Unity.

http://gyanendushekhar.com/2020/03/15/play-video-in-unity-3d/

http://codesaying.com › playing-video-in-unity

Playing Video in Unity - Code Saying

Jul 6, 2020 — **Unity** stores such **imported video** files as Vi an **imported video** file, which the **Video** Player component

http://gyanendushekhar.com › 2020/03/15 › play-video...

Play video in unity 3D - Gyanendu Shekha

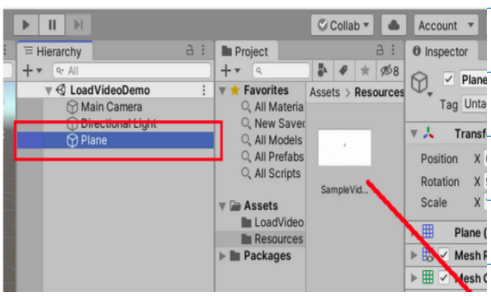
Mar 15, 2020 — Play **video in unity** 3D · Play **video** at run Player game object **in** the scene. (Game Object -> **Video** ->

Play video on a plane

Step 1: Create a Plane primitive game object. (Game Object -> 3D Object -> Plane)

Step 2: Add VideoPlayer component to it.

Step 3: Drag and drop video file to Video Clip Input field in the editor



Script to play the video clip.

```
using UnityEngine;
using UnityEngine.Video;

public class PlayRuntime : MonoBehaviour
{
    private VideoPlayer MyVideoPlayer;

    private void Start()
    {
        MyVideoPlayer = GetComponent<VideoPlayer>();
        // play video player
        MyVideoPlayer.Play();
    }
}
```

Submission Note:

1° Indicate Team Coordinator, And Each team member By Providing First, Last Name & SID

What to submit:
1. one submission for each team;
(1.1) include the team coordinator first name, last name and SID, then each team member's first name, last name and SID;
(1.2) submit the separate program code used in the project;
(1.3)save the project, use the following project name:
firstName-lastName-last4digitsSJSUid-unity-karting-path-2021-final-mm-dd; be sure to test out your saved project can be opened and included all the asset so it can run and can be tested by others.

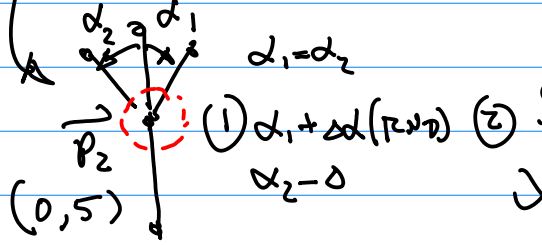
please be sure to provide Explanation/Description for the task assigned & Completed by Each individual.

2° PPT should be included.

For Patric/Anmol/Chris Team.

Post-Processing $T^{-1} = \begin{pmatrix} 1 & 0 & -\Delta x \\ 0 & 1 & -\Delta y \\ 0 & 0 & 1 \end{pmatrix}$

Rotation



$\alpha_1 = \alpha_2$

(1) $\alpha_1 + \Delta\alpha$ (RND) (2) $\lambda = 0.8$

$\alpha_2 - 0$

$\lambda + \Delta\lambda$ (RND)

$$\begin{pmatrix} x_n' \\ y_n' \\ 1 \end{pmatrix} = T^{-1} R T \begin{pmatrix} x_n \\ y_n \\ 1 \end{pmatrix}$$

Pre-processing $T = \begin{pmatrix} 1 & 0 & \Delta x \\ 0 & 1 & \Delta y \\ 0 & 0 & 1 \end{pmatrix}$, $\Delta x = 0$, $\Delta y = -5$