

CE318 Lecture 2 Notes

Vectors

The vector: (2, 5), means:

- 2 units per time period in the x axis
- 5 units per time period in the y axis
- $\sqrt{2^2 + 5^2} = \sqrt{29}$

Unit vectors are useful when you only want to know about the direction

- i.e. looking at an enemy
- Normalize method is useful for this

Transforms

Smooth Damp can be used to create a gradual position change towards a point

Transform.Translate is useful for moving a game object to a certain position

- Uses relative (local) axes by default (this can be changed)

Lerp (Linear Interpolation) is used for moving or changing values over a period of time

- i.e. Start fast from the starting point and gradually get slower towards the end point
- i.e. Button animation
- i.e. Fade to back screen

`lerpValue = Mathf.Lerp(Vector3 minValue, Vector3 maxValue, float interpolationPoint)`

Pitch is rotation around the x axis Yaw is rotation around the y axis Roll is rotation around the z axis

Negative localScale can be used to flip a game object

Transform.Translate and Transform.Rotate can be given an additional argument which denotes whether to change the transform globally (Space.Self) or relative (Space.Self) to the parent game object

Physics

If a game object has a collider, physics functions should be used rather than modifying its transform directly

Quaternions

All rotations in Unity are done by Quaternions Individual Quaternions rotations (i.e. x, y, z) should not be changed Quaternions are not effected by the Gimbal Lock Problem

Quaternion.LookRotation creates a rotation towards a specified forwards and up direction Quaternions.Slerp is slerp for Quaternions

Player Input

Positive and Negative Buttons Exercise

For an aircraft:

- Horizontal
 - Negative: Rotate left in the z axis Move left
 - Positive: Rotate right in the z axis Move right
- Vertical
 - Negative: Dip downwards
 - Positive: Point upwards
- Thrust
 - Negative: Shift
 - Positive: Spacebar
- Fire
 - Negative: (left click)
 - Positive: (Right click)
- Scroll wheel:
 - Negative: Zoom out (Scroll back)
 - Positive: Zoom in (Scroll forwards) Window movement (shake, etc.)

You will get extra marks for your final game if it includes a tutorial

Input Manager

Axis Options (Basic):

- Name
- Positive Button
- Description Name
- Negative Button
- Description Negative Name

Edit -> Project Settings -> Input

Key Input

`Input.GetAxis(string name)`: To check for a specific axis `Input.GetKey(string key)`: To check for a specific key

- Whilst the key is being held down `Input.GetButton()` `Input.GetKeyDown(KeyCode keycode)`
- Only on the first frame triggered `Input.GetButtonDown()` `Input.GetKeyUp(KeyCode keycode)`
- Only on the first frame triggered
- Key codes can be gather via predefined values (i.e. `KeyCode.A`, `KeyCode.Space`) `Input.GetButtonUp()`

`Input.GetButton` is recommended over `Input.GetKey`, as it allows access to input controls specified in the Input Manager `Input.GetKey` can only return true or false `Input.GetAxis` can return any value between -1 and 1 `Input.GetAxisRaw` can only return -1, 0 or 1

Axis Options (Advanced):

- Gravity
 - This effects how quickly the axis value will return to 0 once it has been let go
- Sensitivity:
 - This effects how quickly the axis value will go to 1 or -1 once it has been pressed
- Dead
 - For joysticks, can create a "dead zone", where input is not registered, useful for ignoring slight joystick movement
 - Can make value zero if both the positive and negative values are being held
- Snap:
- Invert: Positive is Negative and Negative is the previous positive
 - Useful for dizziness
 - Useful for making plane controls seem "natural" when rotated passed a certain amount

Mouse Input

Useful for detecting clicks on a collider or a GUI element

OnMouseDown()

- Called on the frame a mouse button is click on a collider or GUI element `OnMouseDown()`
- Called on the frame a mouse button is released on a collider or a GUI element `OnMouseUp()`
- Called on the frame a mouse enters a collider or GUI element `OnMouseEnter()`
- Called on the frame a mouse exits a collider or GUI element `OnMouseExit()`
- Called every frame a mouse is over a collider or GUI element `OnMouseOver()`
- Called every frame a mouse is over a collider or GUI element `OnMouseDrag()`
- Called every frame after a mouse has been click on a collider or GUI element and the mouse is still being held down `OnMouseDownAsButton()`
- Called on the frame a mouse is released on the same collider or GUI element as it was pressed on