Game Development

# User input

To get the direction to another entity, take away the position of the current object from the object you are trying to get the direction to, then normalize the value.

Input class

Used to get input from user.

Methods include:

* GetKey - returns a bool depending on if a certain key is getting pressed.

## Transform

Contains the position, rotation, and scale of a gameobject.

Fields include:

* Forward - The direction the transform is facing

## Time

Fields include:

* DeltaTime- time in between frames.

# Collision

Each shape has a mesh, made up of triangles, which is too complex to be used for collision usually.

Different types of collider for each shape, which is much simpler than the 3D mesh for the object. The collider uses simple maths to detect whether a collision is taking place.

Two functions for detecting collision:

* OnTriggerEnter - For a non-blocking collision, basically just moving through something. One of the colliders must have a rigid body component
* OnTriggerExit - Opposite of above
* OnCollisionEnter - For a blocking collision, like bumping into something.

Turn on IsKinematic to remove physics from rigidbody to just use it for the collider.

Tags can be used to ensure something different happens depending on the object colliding with.

Collider does not have to have the same size and position as the mesh renderer. Can use multiple colliders for one object

# Graphics

### Accessing other components at runtime

Can access other components and access any public field and variable.

### MipMap

Makes smaller resolution version of the same image so there is a suitable version depending on how far away an image is, saves the pc from runtime calculations.

### Compression

Compressing images in unity will massively decrease the size of an image without impacting quality severely.

### Tiling

Uses tiles to repeat a material.

### Height Map

Allows for a texture to be used to change how light interacts with a texture, allowing for it to look 3D

### Sprite Renderer

Will combine sprites in the same object. This allows for dealing with the graphics of each component separately.

# Sound

### Audio Source

Can be added to an object to output a noise. Properties include volume, loop, and spatial blend which can be set to 3d to make sure the sound changes based on the space. A falloff can be changes to outline when the noise can be heard, and how loud it is.

# UI

You can use a canvas to store UI elements in. These elements can be anchored so that they will be in the same place regardless of the user’s screen size. An event system is required to handle button presses, tab selecting, etc. They are usually automatically added when a UI component is created.

Buttons can have a script attached to them, then the button has access to all the functions and public fields in the script to call or edit on click. You can also pass a variable as a parameter into a function.