Unreal 5 Notes

**Quick Shortcuts**

1. Press ‘END’ after selecting OBJECT to snap it to nearest Object.
2. Select OBJECT and Press and hold ‘ALT’ and Drag with mouse to the direction to clone OBJECT.
3. Use Shift+RM Click to select multiple Objects.
4. Go to QUICK ADD TO PROJECT and click on Visual Effects then select the Options suitable for SKY i.e Atmosphere, Cloud, Fog, etc, then click on LIGHTS and then Select the appropriate Option as per the Scenario.
5. Always add the appropriate Content Packs from the ADD then Add Feature or Content Packs such as Third Person Pack, Vehicle Pack, etc.
6. To make a bouncing of light we have to set render to Lumine in Project Settings.
7. Go to Window and then select ‘WORLD SETTINGS’ to set the default settings for world/map/level.
8. To change Keyboard and Mouse settings go to Project Settings and then Click on ’Input’.
9. Click the Right Mouse Button and Scroll the Mouse wheel up/down to speed up or slow down the movement speed.
10. Select/Right Click a Tab from other Window drag and drop it into main window to add the tab in main window.
11. We can Apply Physics to Objects by selecting them and the setting the Simulate property to True by checking the checkbox in the Physics Tab in Properties.
12. We can perform Ejecting by holding F8, ejecting is a process in which we move out of our player body (like a soul moves out of a body). Ejecting works when we are in the main MAP.
13. We should add Player Start to indicate the Spawn Point.
14. Actor Panel can be used to create a simple level, It is available in the Quick Add to Project Section in the top left.
15. To scoop out the contents of a shape we can use Subtractive form of it.
16. If we want to form a single object out of multiple objects then simply drag and drop that mesh into the parent mesh’s static mesh component this will nest that object.
17. We have to select a single object by double clicking it and then edit it for controlled collision from the collisions drop down list, then we have to create a Blueprint of that object and save it so that we can use it again. Also apply physics and mass as required.
18. After creating a variable compile it, we can ctrl+drag to get it and alt+drag to set it.
19. We can drag and select multiple nodes in a Blueprint and then click ‘C’ to add them in a function. Or you can drag and select all the nodes and then right click and select ‘Collapse to Function’.
20. Open level by name/reference can be used to reload the level also use get current level name.
21. Go inside the project folder the right click on the .uproject and the select switch unreal engine version and then select the version u want.
22. In the blueprints to the custom object in the right side details/properties search auto then in pawn section change the values as per the requirements.
23. Use ctrl+drag to select and drag a line from one node to other.
24. We can change our Code Editor by clicking on edit(present in the top left menu) then select Editor Preference then go to Source Code option and then select your preferred Code Editor.
25. To create a C++ Class, Go to Terminal(top left Menu) then select the Run Build Task (Ctrl+Shift+B). Game Engine must be closed.
26. Use Paste Here from the Right Click Menu in then go to Edit option or use the keyboard shortcut.
27. The Tick function is like a recurring function.
28. In UPROPERTY we can pass another argument called category=”name”, this will create a category in the property outliner.
29. We also have VisibleAnywhere which allows us to see the property anywhere, but does not allow us to edit anywhere.
30. So in order for the playing character to be able to handle the collisions properly when being idle, we have to keep the character moving in small steps when the character Is idle, use moveupdatecomponent and set the X and Y to some small value then again to the same for another moveupdatecomponent but in minus value so the character moves back to its initial position. E.g moveupdatecomponent x=1,y=1,z=0-> moveupdatecomponent x=-1,y=-1,z=0 and keep the sweep ticked and do not forget the getactorrotation.
31. Go to the blueprints dropdown and then go to projects settings->Game Mode then you can either Create or Select a game mode such as Game Mode Base. And make sure if you select a game mode in projects settings then set the game mode to not overridden in world override.
32. In order to use play from here you must first set the player start and link it to the model that will spawn when we hit play and also follow the above step 34.
33. To display message in output log use the ulog i.e UE\_LOG and then click on the extending option provided by CAPTAINs vs code extension for displaying a sting don’t forget to add \*before the vatiable while displaying.
34. **Pure Function** is a with function with no side effects (which cannot be seen on the screen) but it has return value in short its used for calculation. We can make a function pure by ticking the pure checkbox after selecting a function.
35. Member Function are functions that are stored within the class and called by objects. To do double click on a mesh/object in content drawer then click on open full blueprint editor.
36. Self node points to the current instance.

**Blueprints**

Event Graph = place/Canvas where we place different Blueprint Nodes(Premade Functionalities).  
each Blueprint Node consists of Pins or Execution Pins(Input pins tell when to run this node , Output Pin tells what to do after) that we can connect to each other

Right Click to open Blueprints Action List.

Select an Object in the Game then open the level blueprint right click and create a reference node(indicates where the data of the object is stored in form of address) These nodes have special pins called data pins which are used to Input and output data (left pin= Input and right pin= output).

Event Begin Play = Event the takes place when game start, more like **when** a particular functionality should take place.

Print String = It display Text In-Game.

We can use a button node to attach different events to it, just right click and select the button u want and then connect it as per your needs.

On right clicking if you want to see only the nodes which are related to your currently used node then tick on Context Sensitive.

Jump Physics

Force and Impulse   
Force = Mass \* Acceleration

Impulse = Mass \* Velocity Change

**Data Types**

Integer(int32) – green\*blue

Float – Light green

String - pink

Boolean - red

Struct – Vector(X,Y,Z), Rotator(X,Y,Z), Transform(location,rotation,scale)

**Condition**

Branch Node

Equal, Greater, etc

**Player Related Nodes**

Get Player Pawn

Get Actor Location

Get Control Rotation

Get Actor Forward Vector

**Vectors**

Get Forward Vector

**C++**

Int -> int32

UCLASS() before a class

A<Class Name> -> Actor Class

While declaring a variable use UPROPERTY(EditAnywhere) to make the variable visible.

Use the blocks symbol button at bottom Right corner for Recompiling and Live Coding.

BeginPlay() function is executed when the game starts.

FVector – F specifies that it is a Struct, we use FVector constructor to assign default value.

**Binary Space Partitioning (BSP)**

**Delta time (very important to match game frames)**

**Simple setup for UI**

1. Create a Map LVL from File in the Maps/other folder.
2. Create a UI Folder
3. In UI folder right click and the select User Interface -> Widget Blueprint -> User Widget.
4. New File will be created rename it if needed and then double click it to open and for more convenience Dock it to main app.
5. In search palette search for canvas and drag and drop it in Hierarchy and then add all other widgets in it as needed.
6. The go to Graph section from upper right corner.
7. get player controller -> SET show mouse cursor and Event Constructor -> SET show mouse cursor and SET show mouse cursor -> set input mode UI only and get player controller -> set input mode UI. (Tick show mouse cursor)
8. **For Start** from button -> open level(by name) then add the name of level in it. Get player controller -> set show mouse cursor, open level (by name) -> set show mouse cursor -> set input mode game only & get player controller -> set input mode game only.
9. **For Quit** from button -> Quit Game.
10. **For Linking Menu** to View Port open game level blueprint, event begin play -> create main menu widget and enter menu name -> add to viewport both pin connect.
11. **For Options Menu** click on button then select on click. For screen modes create a varriable with datatype ewindowmode then drag drop var with get mode -> to int -> subtract (val 1) -> clamp(min 0 max 2 depends on modes) -> to byte -> drag drop screen var and use set mode and then finally on click -> screen var set mode.
12. **Same process for resolution (int datatype)** resindex var -> add/sub -> clamp (min 0 max 4 depending on res nos) -> set resindex var and on click -> get resindex var. resindex var -> switch on int (no of pins = total res nos), set from bot res- and + -> switch on int. create res var with int point datatype. Set res (pin right click split struct pin) and create 4 copies and then link them to switch on int and then set all the resolution x and resolution y values (for res).
13. **Same process for graphics/Quality (int datatype) like resolution but do not use switch on int.**
14. **For Vsync use Boolean datatype and then on click -> set vsync var(no tick),on click -> set var (with tick).**
15. **For Apply, on click -> get game user setting -> set full screen mode. Screenmode var -> set full screen mode, get use game settings -> set overall scalability level and screen fullscreen mode -> set overall scalability level. Get qualityindex -> set overall scalability level.**
16. **To store and load the previous saved settings right click and select event construct -> get game user settings -> load settings ->set screenmode -> set qualityindex -> set vsync -> set res and then use get and is fun of all and connect them to the set vars. Set res -> break int point then use equal = then -> branch -> set resindex (for each 0 to 4) and then link each false of branch to next branch.**
17. **To change text select the text then from content then bind then create new binding then get var (convert if needed then connect to return node).**
18. **Got to main menu options button then click on on click then on click -> remove from parent -> create widget(add name) ->add to viewport**

**Learn in Detail**

Blueprint Classes and Instances

What is static mesh component

Vector Arithmetics

Collision