Advanced Games

Scripting

Why?

- Game engine code <u>has</u> to be fast and optimal.
- Game engines are big, bulky, and complex.

- Gameplay code is *less* optimal
  - Usually just loads of logic checks and globals
  - Can lead to horrendous spaghetti
  - I built a goddam beautiful rendering system and I don’t want this shoddy gameplay logic near my engine code, and neither should you.

google image for “gameplay logic”

```
61                  break;
62              case State.OutOfAmmoCooldown:
63                  // Wait for turret to cool down
64                  _CooldownTimer -= Time.deltaTime;
65                  if (_CooldownTimer <= 0.0f)
66                  {
67                      // Ready to go!
68                      _CurrentState = State.WaitingForTarget;
69                      _Ammo = 10;
70                      _ShotTimer = 0.0f;
71                  }
72                  break;
```

An example script

```
1.  local Player = game.Players.LocalPlayer
2.  local Character = Player.Character or Player.CharacterAdded:Wait()
3.  local UserInputService = game:GetService("UserInputService"):GetMouse()
4.  local ThrusterList = {BackThrusterOne,BackThrusterTwo,LeftThruster,RightThruster}
5.  local DataTable = {game.Players.LocalPlayer,,false,Player:GetMouse()}
6.
7.  UserInputService.InputBegan:connect(function(Input, processed)
8.  if Character.Humanoid.SeatPart ~= nil then
9.  if processed == false then
10. if Input.KeyCode == Enum.KeyCode.W then
11.     ShipModel.Thrusters.BackThrusterOne.ParticleEmitter.Enabled = true
12.     ShipModel.Thrusters.BackThrusterTwo.ParticleEmitter.Enabled = true
13. elseif Input.KeyCode == Enum.KeyCode.A then
14.     ShipModel.Thrusters.LeftThruster.ParticleEmitter.Enabled = true
15. elseif Input.KeyCode == Enum.KeyCode.S then
16.     ShipModel.Thrusters.FrontThruster.ParticleEmitter.Enabled = true
17. elseif Input.KeyCode == Enum.KeyCode.D then
18.     ShipModel.Thrusters.RightThruster.ParticleEmitter.Enabled = true
19.
```

It’s not just about segmentation

Why Script?

- Scripting languages used in games since **Quake C**
- Primary benefits of script:
  - Takes pressure off **engineering team**
  - Code becomes data—**rapid iteration**
  - Empowers **content creators**
  - Key enabler of **mod community**

Why Script?

- Scripting languages used in games since **Quake C**
- Primary benefits of script:
  - Takes pressure off **engineering team**
  - Code becomes data—**rapid iteration**
  - Empowers **content creators**
  - Key enabler of **mod community**

It’s not just about segmentation

- Iteration
  - Hot-reloading gameplay
- Ease-of-use

Other benefits

- Useful to model flexible AI and gameplay
  - character personalities, behaviors
  - mission creation, dialogs, level design

- Scripts are satellites to the core engine
  - can be written in a different language (more accessible to non programmers)
  - <u>run in safe environment</u>
    - (faults are not backfiring to the main application)

- can be coded by less technical people
  - AI designers, content producers, gameplay designers, interns


JS


UDK



Lua

Ok, but how?

You need two things:

1. A runtime script interpreter/compiler

   this actually run the scripts, will be in it’s own thread

2. Bindings from your C++ code to the script

   which functions are exposed to lua

Example C++/Lua binding

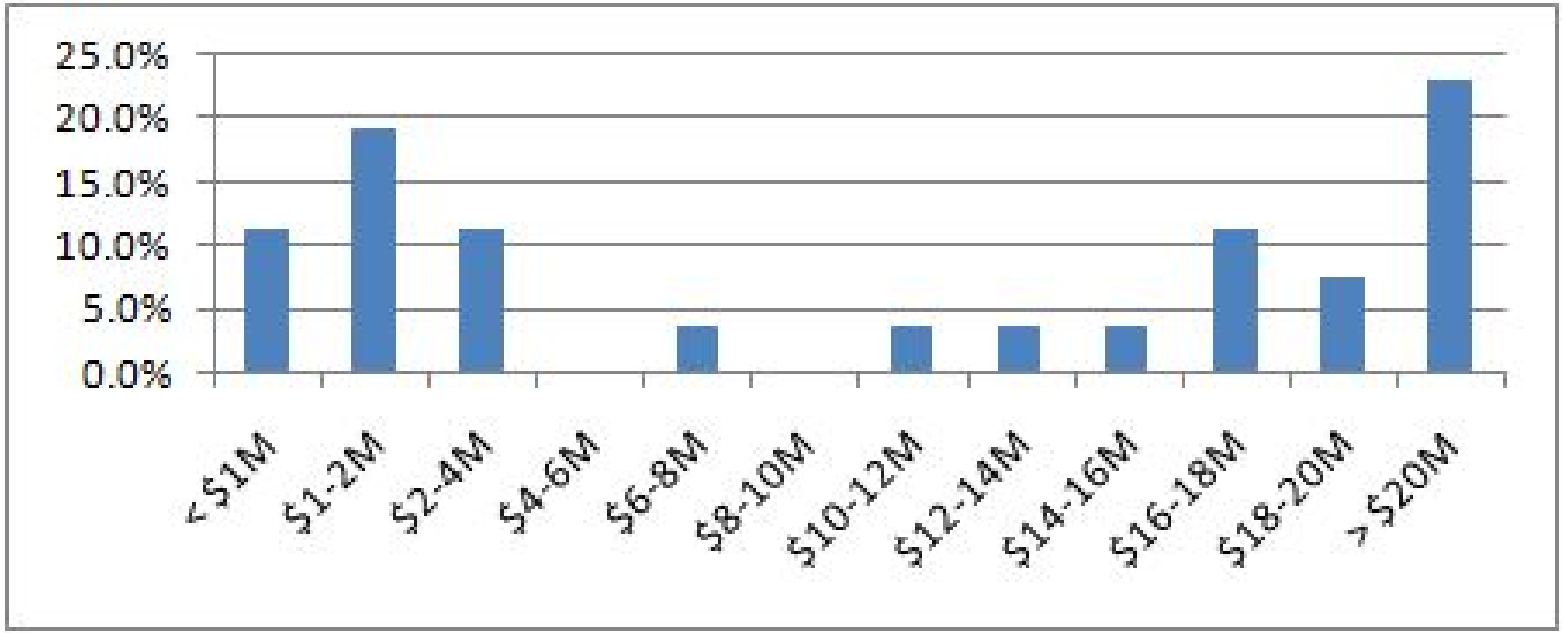
CPP file

```cpp
#include “selene.h”
int my_multiply(int a, int b) {
    return (a*b);
}
// Register the function to the Lua global "c_multiply"
sel::State state;
state["c_multiply"] = &my_multiply;
```

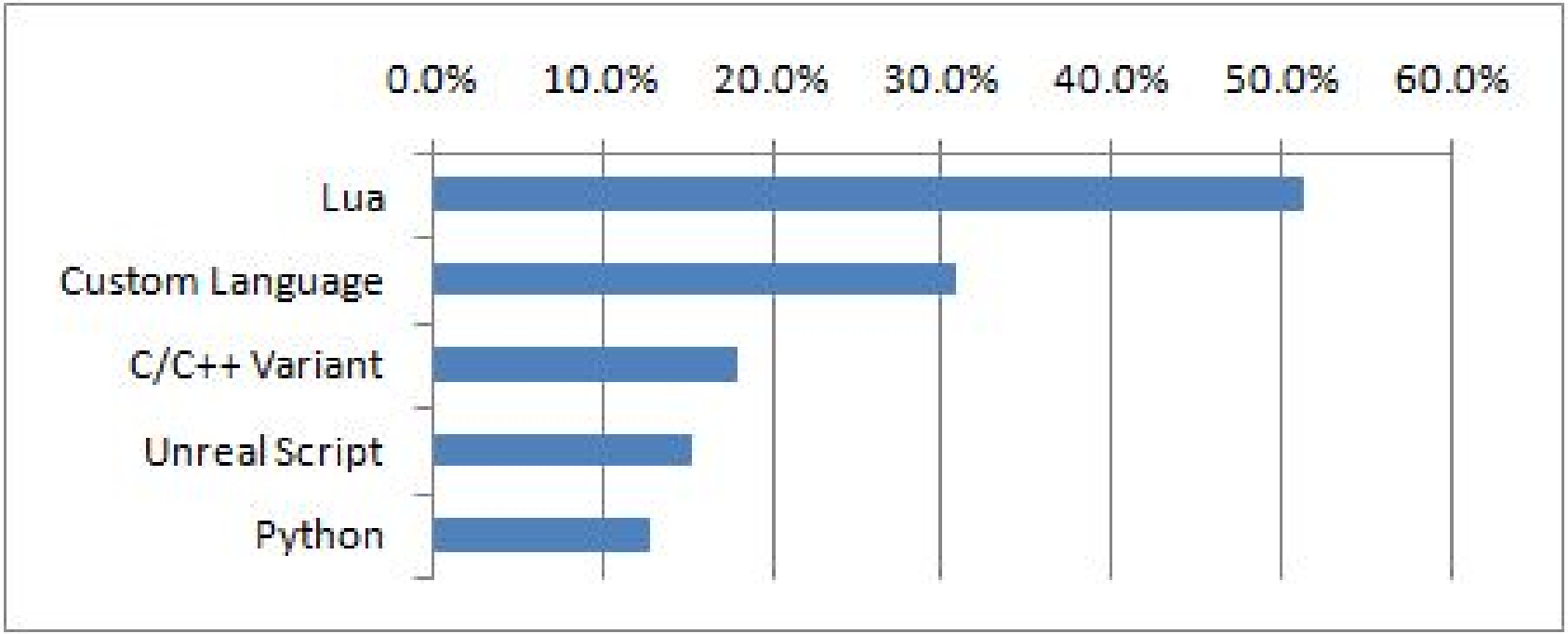
Lua file

```lua
print(c_multiply(2,4))
```

Who’s Using Scripting



What are they using?



It’s not all High level

The coolest programming language I have ever seen..

Game Oriented Assembly Lisp

“GOAL does not run in an interpreter, but instead is compiled directly into PlayStation 2 machine code for execution”

Used on:

- Jak and Daxter
- Crash Bandicoot

  (Game Oriented Object Lisp (GOOL))

http://all-things-andy-gavin.com/2011/03/12/making-crash-bandicoot-gool-part-9/

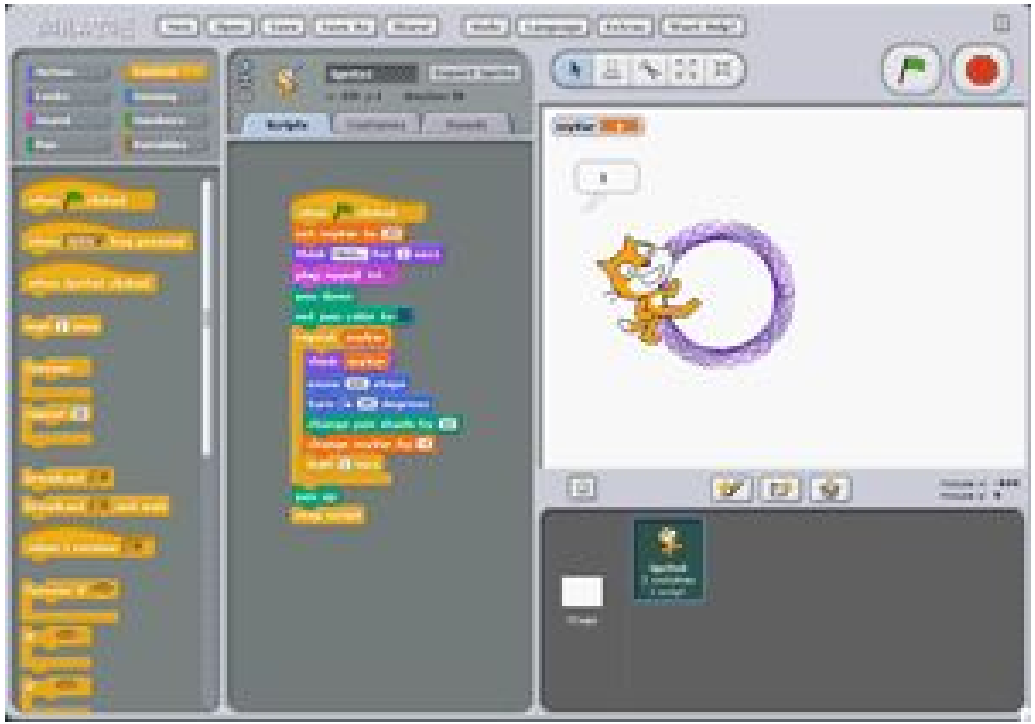
The Making of Jak & Daxter

The Making of The Last of Us

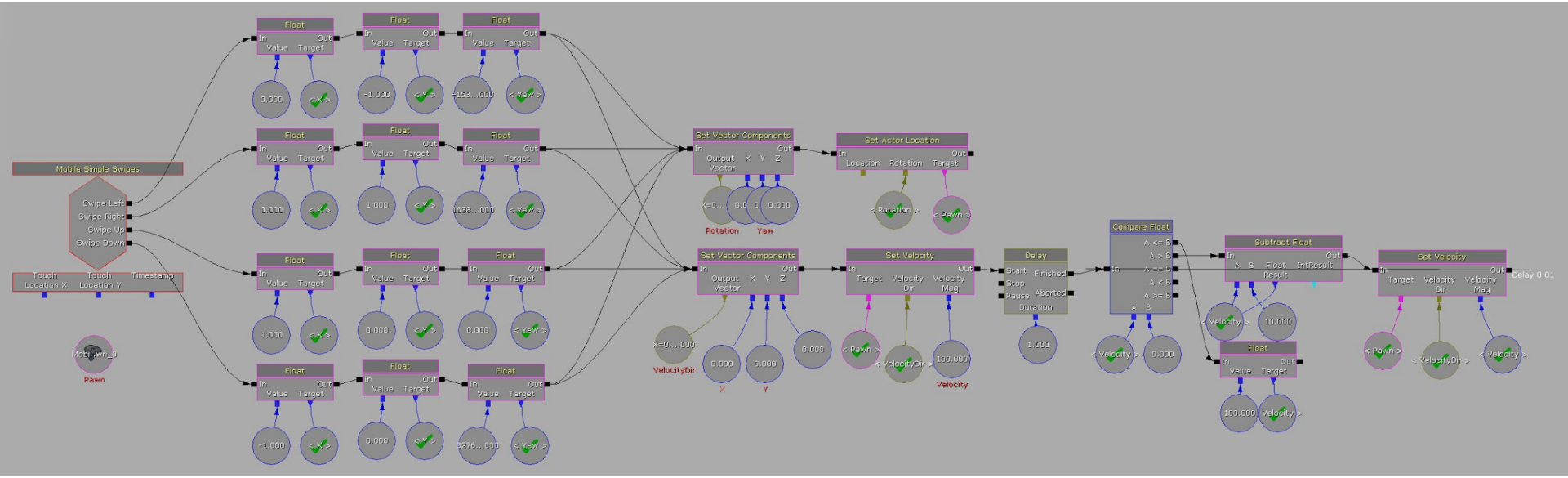
Visual Scripting

Babies first program

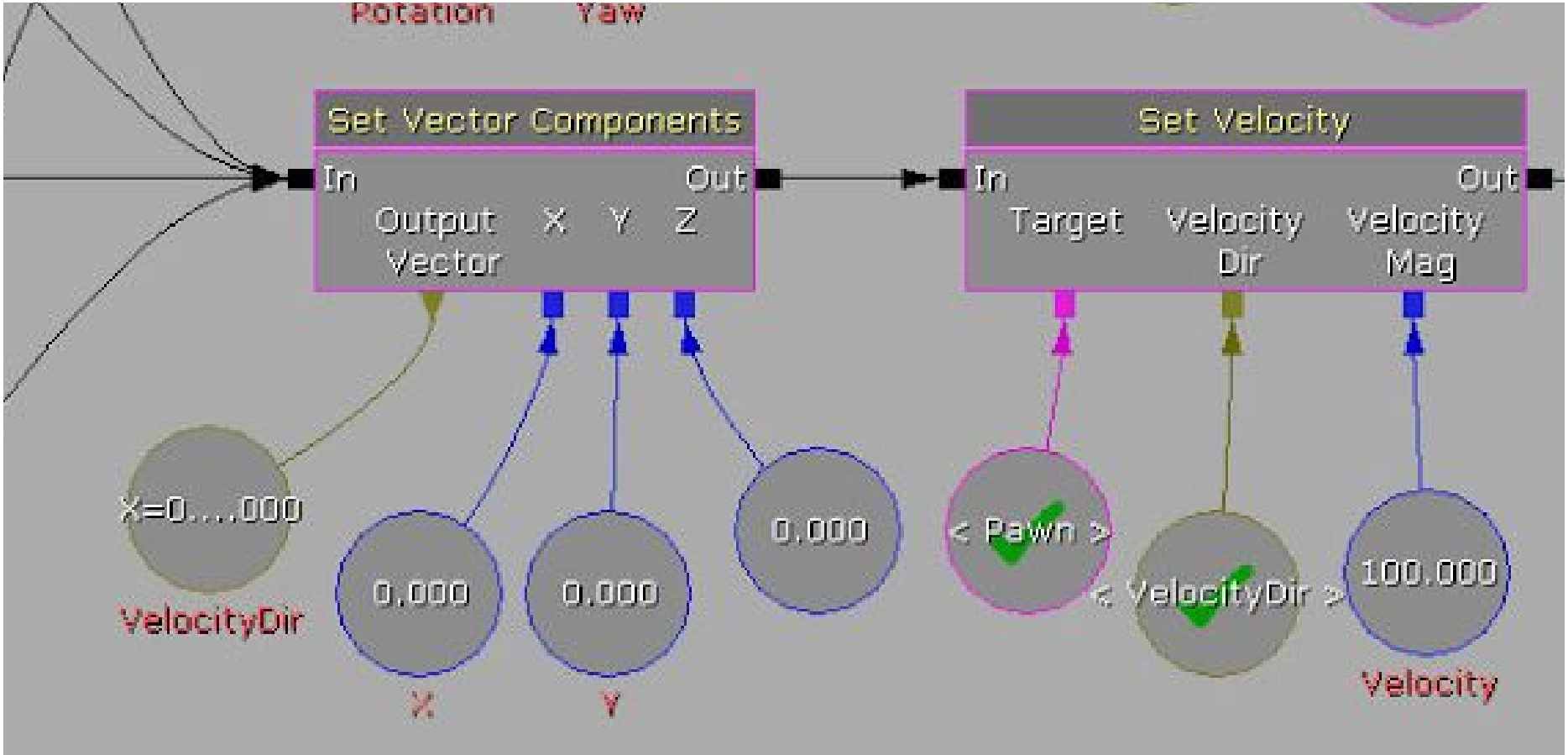
```
when [flag] clicked
set collected to 0
set Score to 0
point in direction 90
set volume to 100 %
forever if <collected < 100>
    if <key space pressed?>
        move 15 steps
        if <x position > 150>
            set x to -150
```



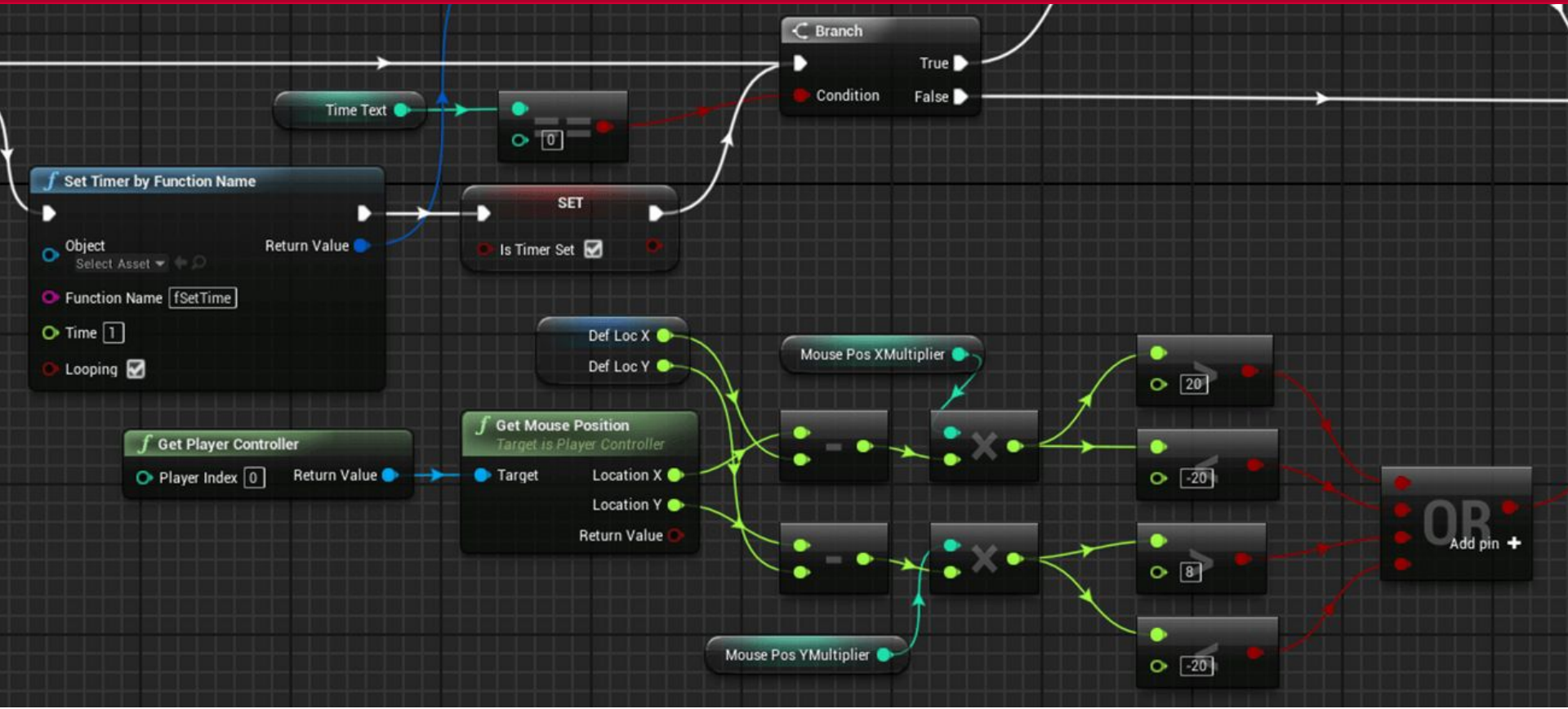
Kismet


Mobile Simple Swipes
Swipe Left
Swipe Right
Swipe Up
Swipe Down
Touch Location X
Touch Location Y
Timestamp
Mobi_wn_0
Pawn
Float
In
Out
Value
Target
0.000
-1.000
1.000
-1638...000
1638...000
-3276...000
3276...000
Set Vector Components
Output Vector
X
Y
Z
Potation
Yaw
VelocityDir
Set Actor Location
Location
Rotation
Set Velocity
Velocity Dir
Velocity Mag
100.000
Delay
Start
Stop
Pause
Finished
Aborted
Duration
Compare Float
A <= B
A > B
A == B
A < B
A >= B
Subtract Float
IntResult
Result
10.000
Delay 0.01

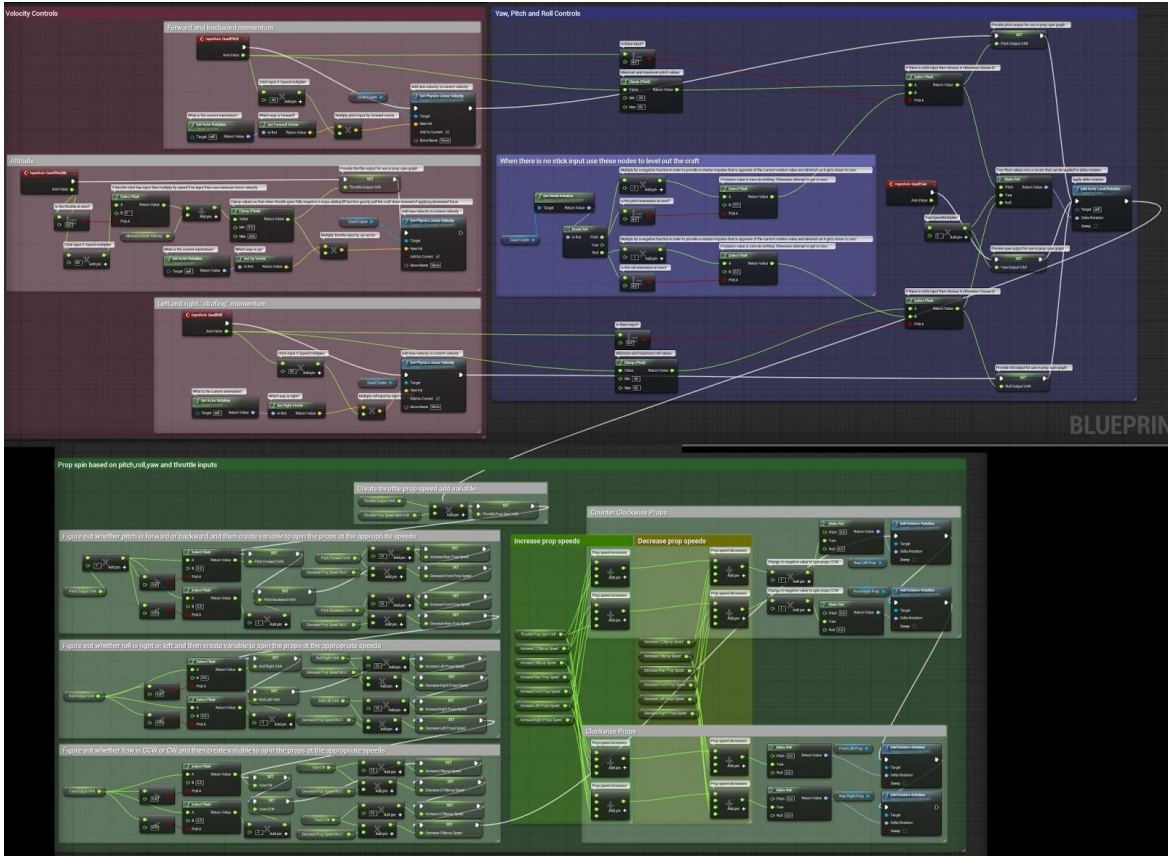
Kismet


Rotation
Yaw
Set Vector Components
In
Out
Output Vector
X
Y
Z
Set Velocity
In
Out
Target
Velocity Dir
Velocity Mag
X=0....000
VelocityDir
0.000
X
0.000
Y
0.000
< Pawn >
< VelocityDir >
100.000
Velocity

Ue4 Blueprints



UE4 blueprints



MIT

APP INVENTOR

MIT App Inventor 2

ai2.appinventor.mit.edu/#6239121076912128

MIT App Inventor 2 Beta

Project ▾ Connect ▾ Build ▾ Help ▾

My Projects Guide Report an Issue

HelloPurr

Screen1 ▾ Add Screen ... Remove Screen

Designer Blocks

Palette

User Interface

- Button
- CheckBox
- DatePicker
- Image
- Label
- ListPicker
- ListView
- Notifier
- PasswordTextBox
- Slider
- Spinner
- TextBox
- TimePicker
- WebViewer

Layout

Media

Drawing and Animation

Sensors

Social

Storage

Viewer

Display hidden components in Viewer

9:48

Screen1

Non-visible components

Sound1

Components

- Screen1
  - Button1
  - Sound1

Rename Delete

Media

kitty.png

meow.mp3

Upload File ...

Properties

Screen1

AboutScreen

AlignHorizontal: Left

AlignVertical: Top

BackgroundColor: White

BackgroundImage: None...

CloseScreenAnimation: Default

Icon: None...

OpenScreenAnimation: Default

ScreenOrientation: Unspecified

Scrollable: ☑

Title: Screen1

VersionCode

PaintPot2

Screen1 | Add Screen ... | Remove Screen | Designer | Blocks

Blocks

- Built-in
  - Control
  - Logic
  - Math
  - Text
  - Lists
  - Colors
  - Variables
  - Procedures
- Screen1
  - HorizontalArrangement1
    - RedButton
    - BlueButton
    - GreenButton
  - DrawingCanvas
  - HorizontalArrangement2
    - ButtonWipe
    - ButtonBig
    - ButtonSmall

Rename | Delete

Media

kitty.png

Upload File ...

Viewer

```
when RedButton .Click
do set DrawingCanvas . PaintColor to [red]

when BlueButton .Click
do set DrawingCanvas . PaintColor to [blue]

when GreenButton .Click
do set DrawingCanvas . PaintColor to [green]

when ButtonWipe .Click
do call DrawingCanvas .Clear

initialize global small to 2

initialize global big to 8

initialize global dotsize to 2

when ButtonBig .Click
do set global dotsize to get global big

when DrawingCanvas .Touched
  x y touchedAnySprite
do ? call DrawingCanvas .DrawCircle
       centerX  get x
       centerY  get y
       radius   get global dotsize
       fill     true

when DrawingCanvas .Dragged
  startX startY prevX prevY currentX currentY draggedAnySprite
do call DrawingCanvas .DrawLine
       x1  get prevX
       y1  get prevY
       x2  get currentX
       y2  get currentY

when ButtonSmall .Click
do set global dotsize to get global small
```

0 0

Show Warnings