

**乗り物は
パラメータゲーにも
向いている!**



[Twitter @vins_cluster](https://twitter.com/vins_cluster)



vinsです。Logicとか多用したワールドが好きです
GameJAM 2020冬では「カンヅメRPG」で
大賞もらいました

クルマやヘリに乗って
広大なワールドで
遊ぶのは面白いが.....



乗り物機能は
パラメータゲージを
やるのにも
向いている!



HP025/045 MP010/010 03
LV01 EXP000000/00020

たたかう

正面に攻撃する。LVUPすると
威力がアップすることもある。

アイテム / X 長押しで降りる

敵を全滅させて球に触れると部屋を出ます

vins

Advance to this sphere to exit this room.

LV01
HP045/045

LV01
HP025/045

5

出口(EXIT)

敵を全滅させて球に触れると部屋を出ま
Destroy all enemys and advance to this sphere to exit this ro

**「乗り物何もわからん!?!」とかで
言ったとおり、乗り物があると
入力機能が充実する**

**●乗り物にConstraintでついてくる
ボタンとか**

パラメータゲーにおいては

入力が大事

→色々なコマンドを使えてこそ

パラメータゲーになる!

そして今回注目したいのは……

→ じれ

Space



Shift



X 長押し

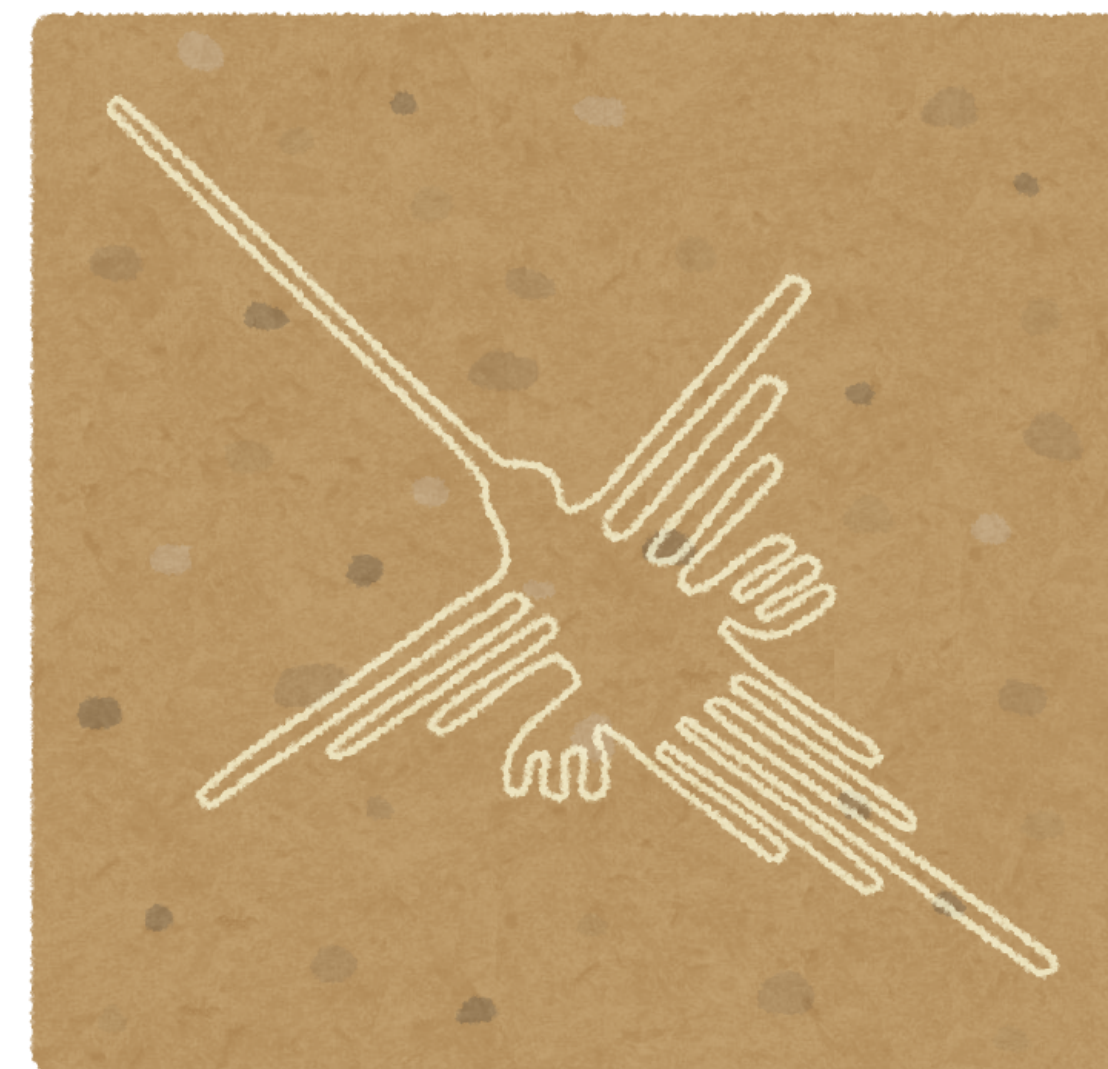


空を飛ぶ、ジャンプするなら

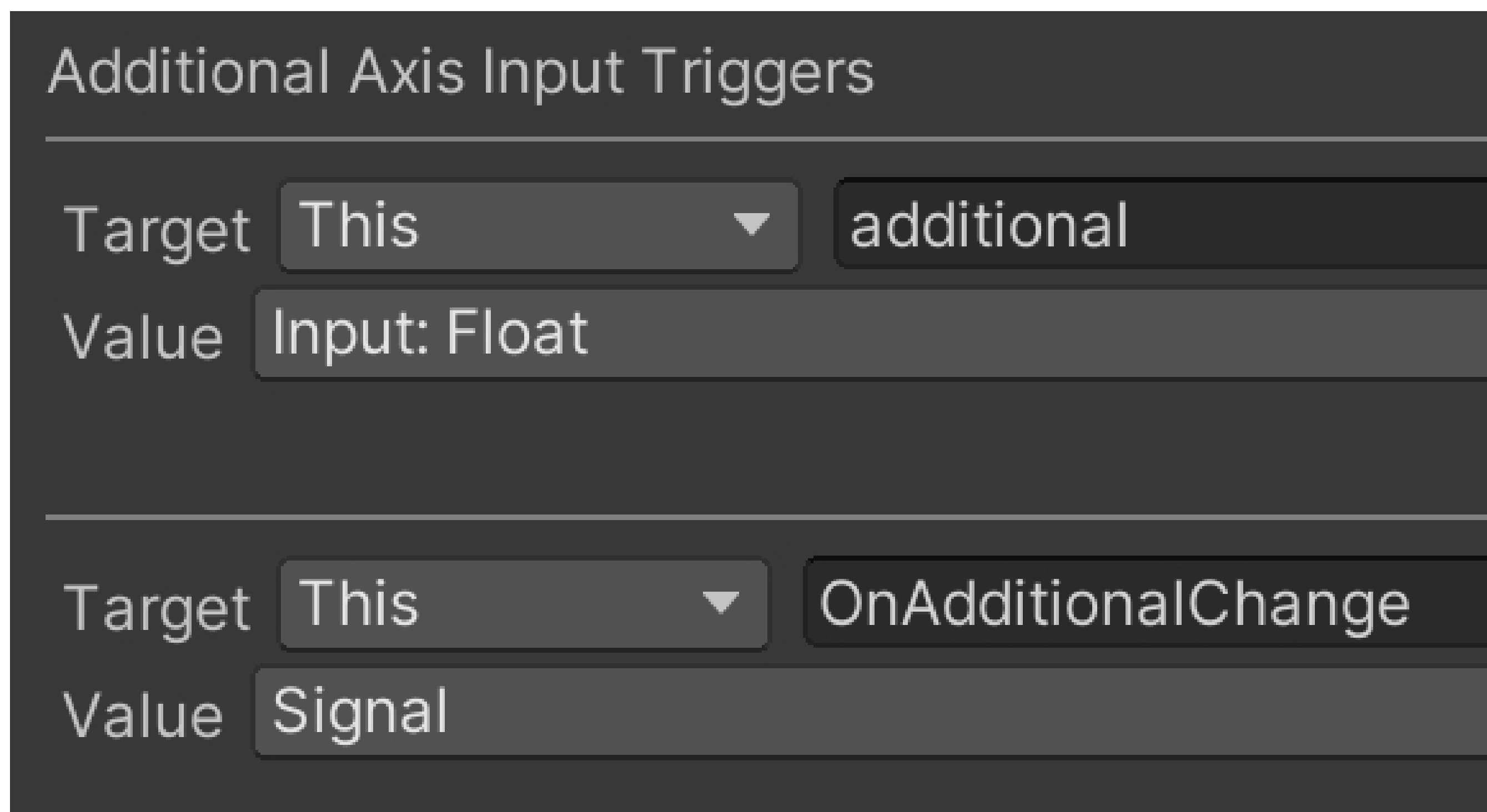
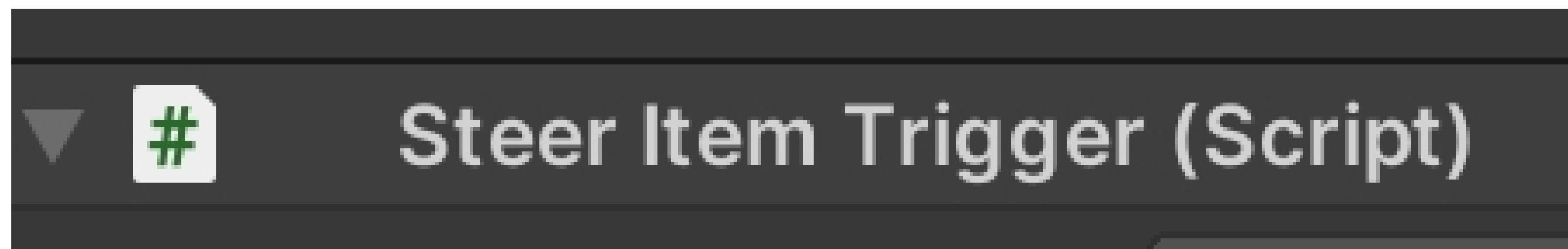
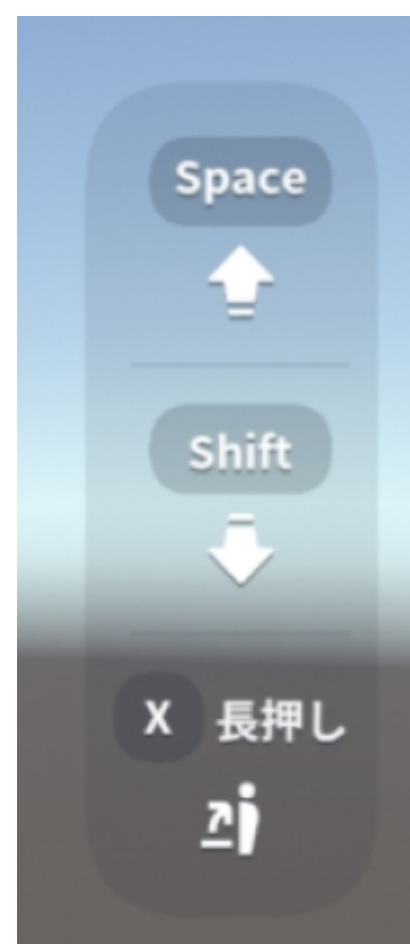
この上下は使うが……

→地上のゲームならなくてもいい

→パラメータゲーの入力に使ってみる!



SteerItemTriggerの AdditionalAxis〜から 上下入力をとれる (プラスなら上、 マイナスなら下)



この上下でコマンドを選択、
乗り物についてくる
ボタンで決定というのを
やってみる



決定ボタン

SteerItemTrigger



ItemLogicと PlayerLogicで 選択中の コマンドを変更

Item Logic (Script) configuration:


- Target: This
- Key: OnAdditionalChange
- Logic 1: This > OnAdditionalA (Signal) = GreaterThan (RoomState Float This additional) > Constant Float 0.3
- Logic 2: This > OnAdditionalB (Signal) = LessThan (RoomState Float This additional) > Constant Float -0.3

Item Logic (Script) configuration:

- Logic 1: Target This, Key OnAdditionalA, Owner SubSelectingUp, = Constant Bool checked
- Logic 2: Target This, Key OnAdditionalB, Owner SubSelectingDown, = Constant Bool checked

Player Logic (Script) configuration:

- Target: Player
- Key: SubSelectingDown
- Logic: Player > SubComNo = Add (RoomState Integer Player SubComNo) > Constant Integer 1



たたかう

正面に攻撃する。LVUPすると
威力がアップすることもある。

で、こここの部分をどうやって
実装するのか？

たたかう
ポーション
ファイア
フリーズ
サンダー
パンチ
キック
なげる

こんな感じの
テクスチャを用意

PlayerLocalUIに **RawImage**で 表示させる



▼ Raw Image

Texture

Color

Material

Raycast Target

Maskable

UV Rect X Y
W H

▶ Animator

▼ # Set Animator Value Gimmick (Script)

Target

Key

Parameter Type

Animator Parameter Name

▼ # Set Animator Value Gimmick (Script)

Target

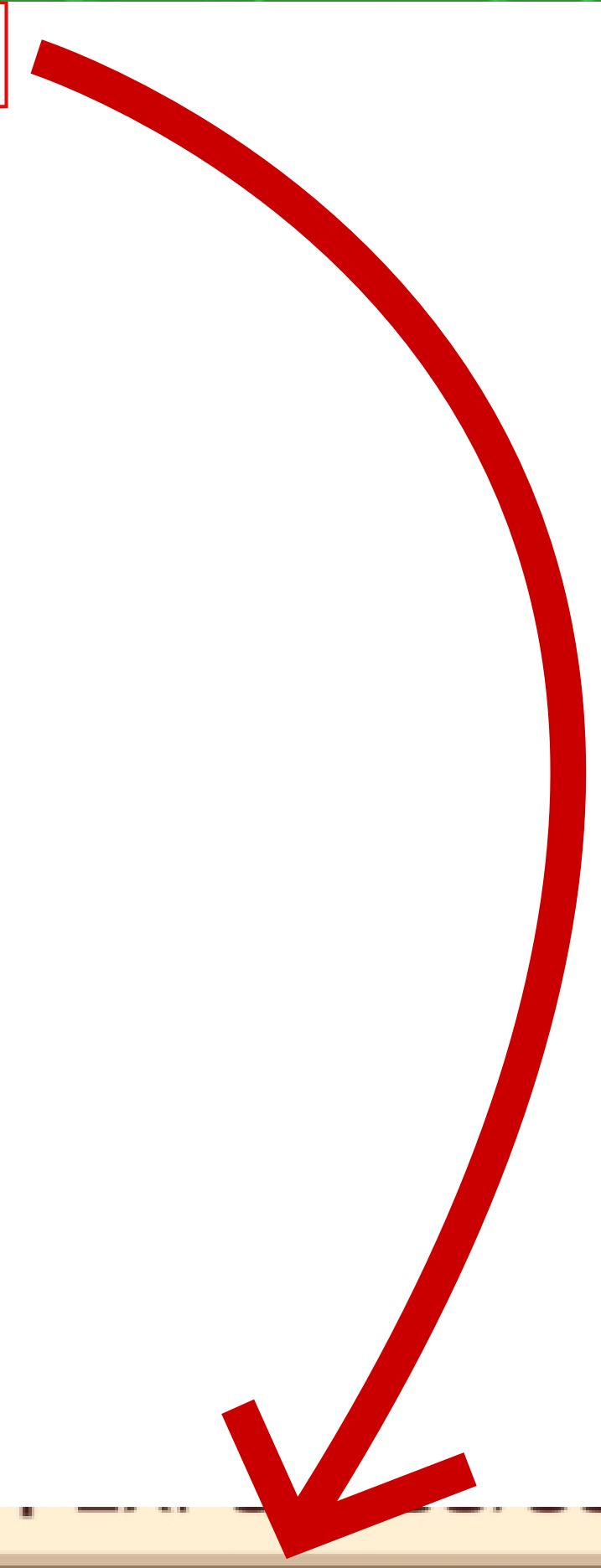
Key

Parameter Type

Animator Parameter Name

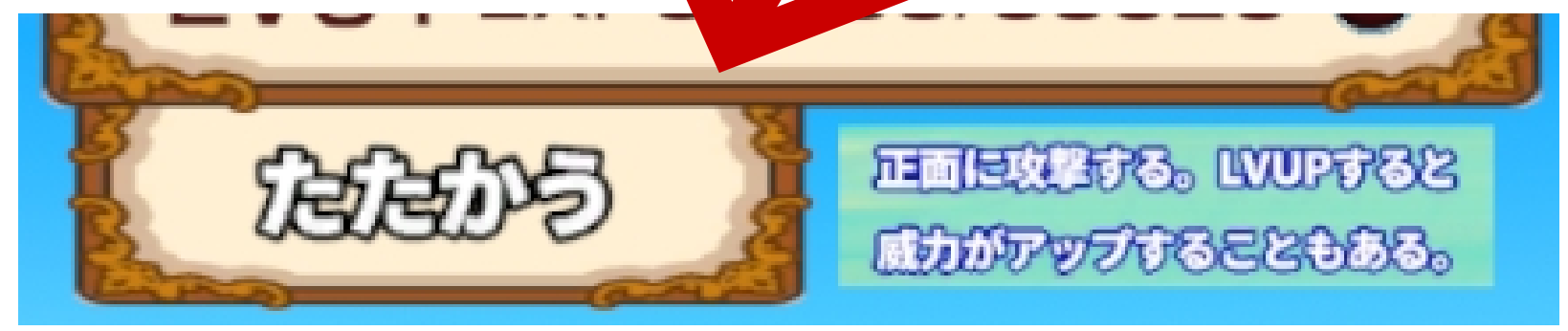


たたかう
ポーション
ファイア
フリーズ
サンダー
パンチ
キック
なげる



UVRectの設定でテクスチャの
一部だけ表示可能

ちなみにYは上下逆なので注意



PlayerLogicから送られてきた数字から
どうやってUVRectの数字をいじるのか？

**Player
Logic**



**RawImageの
UVRect**

SetAnimatorValueGimmickを使い……



✓ Set Animator Value Gimmick (Script)

Target: LocalPlayer

Key: ComX

Parameter Type: Float

Animator Parameter Name: ComX

この数字をいじりたい

X 0 Y 0.9375

W 0.25 H 0.0625

Set Native Size

AnimatorのBlendTree

使うことができます

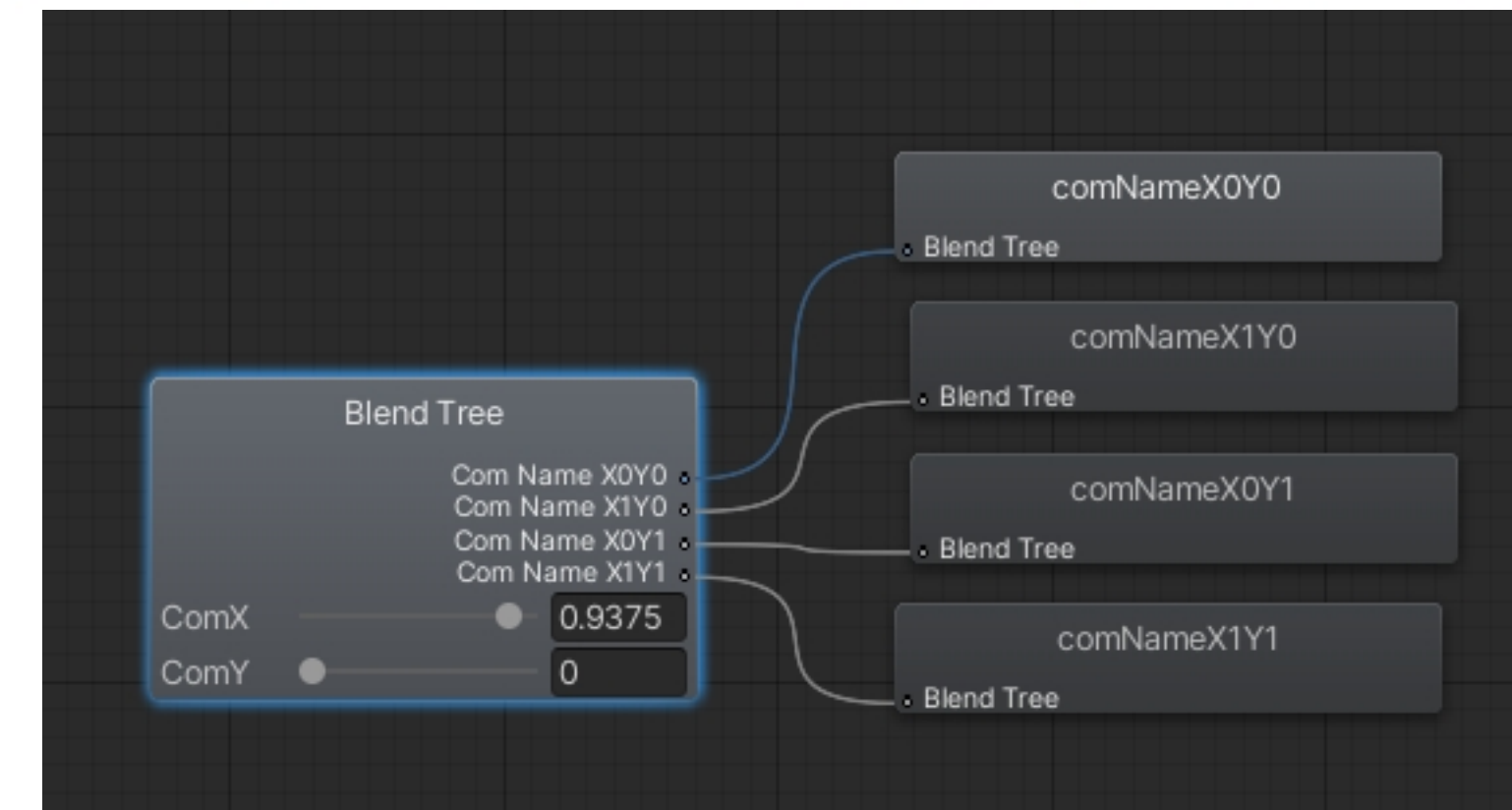
Xが0Yが0

Xが1Yが0

Xが0Yが1

Xが1Yが1

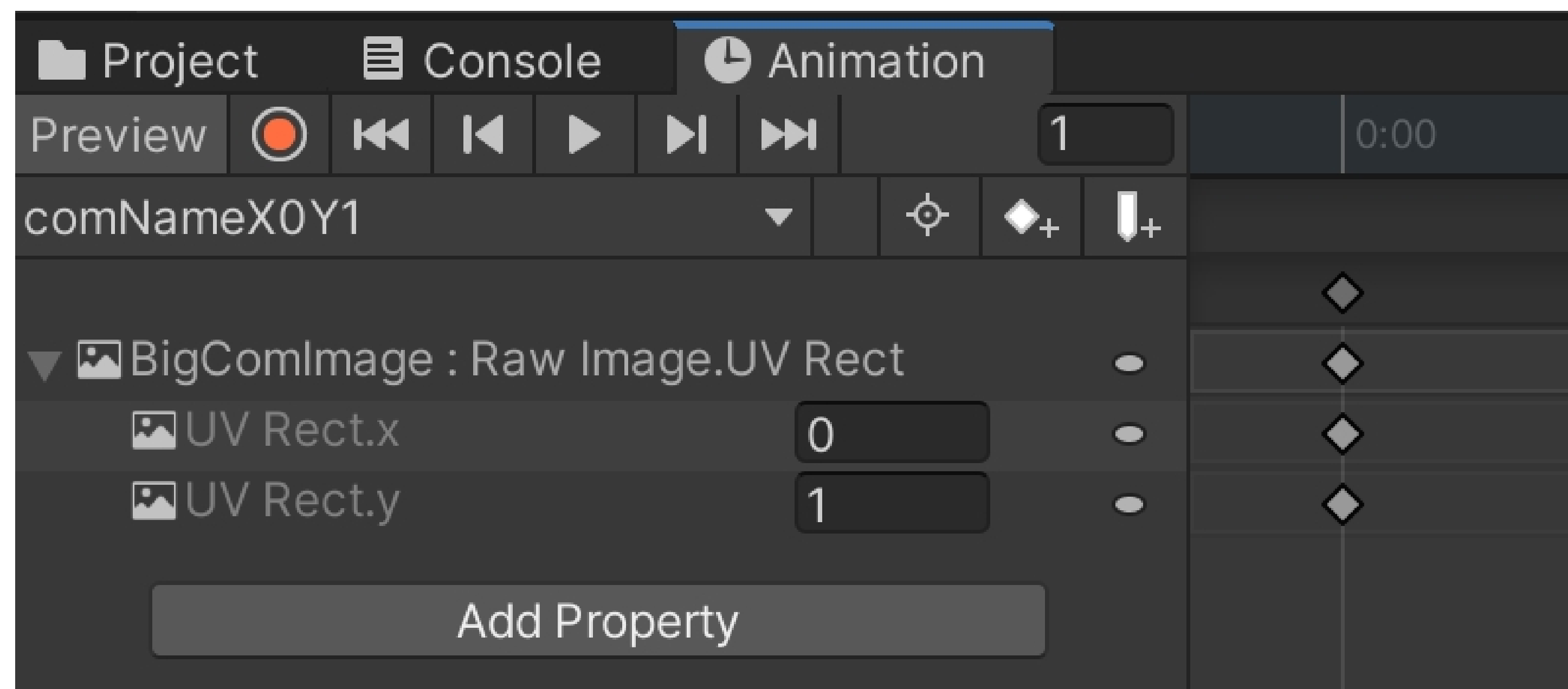
この4つのanimを
ComXとComYという
パラメータでBlend



Blend Type: 2D Simple Directional

Motion	Pos X	Pos Y
comNameX0Y0	0	0
comNameX1Y0	1	0
comNameX0Y1	0	1
comNameX1Y1	1	1

使ってるのは
UVRectを
変えるだけの
単純なanim
×4つ



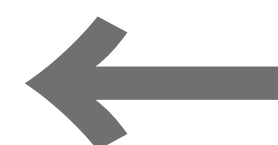
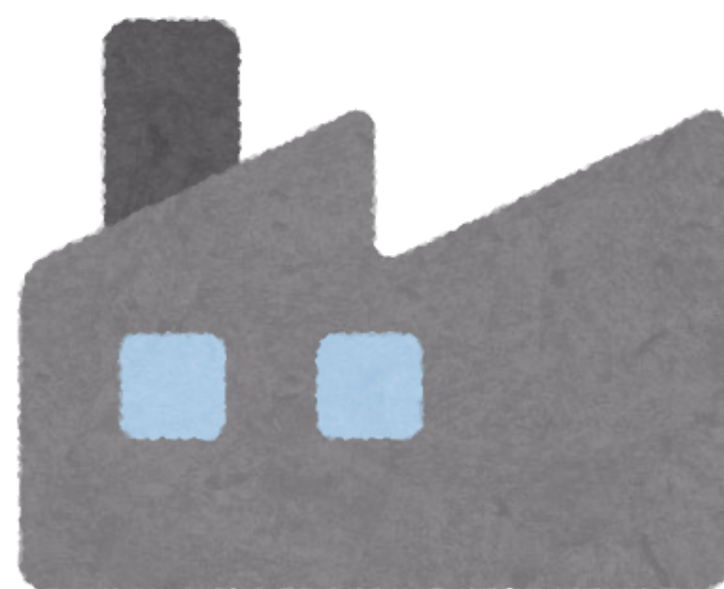
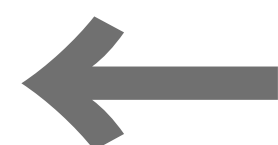
Player Logic



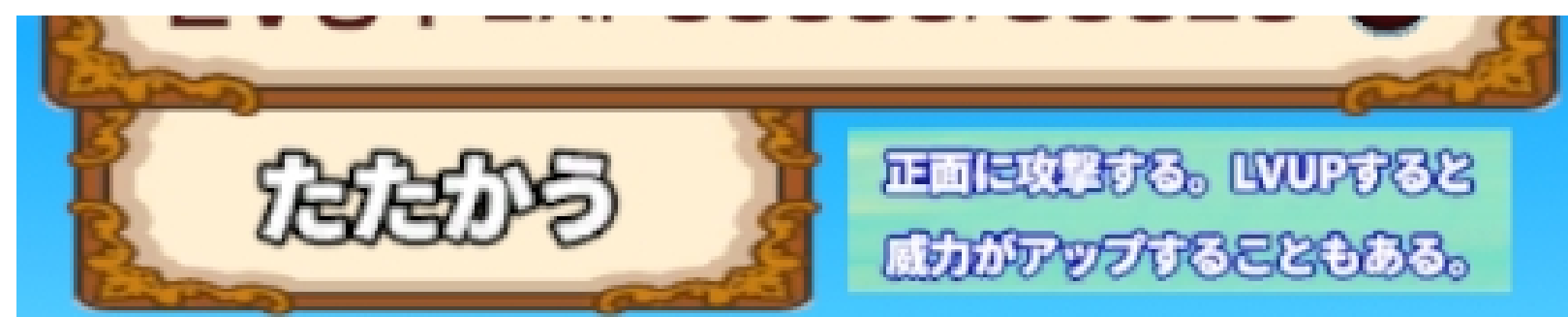
SetAnimator ValueGimmick



RawImageの UVRect



Animatorの BlendTree



うまく合成して

.....



とりあえず、**Animator**は
アニメさせるだけでなく
色々な数字をいじれることを
知ってもらえれば……

ちなみに最初の
PlayerLogicの部分は
こんな感じ

CurrentSkillNoの
数値に従って
ComXとComYを計算

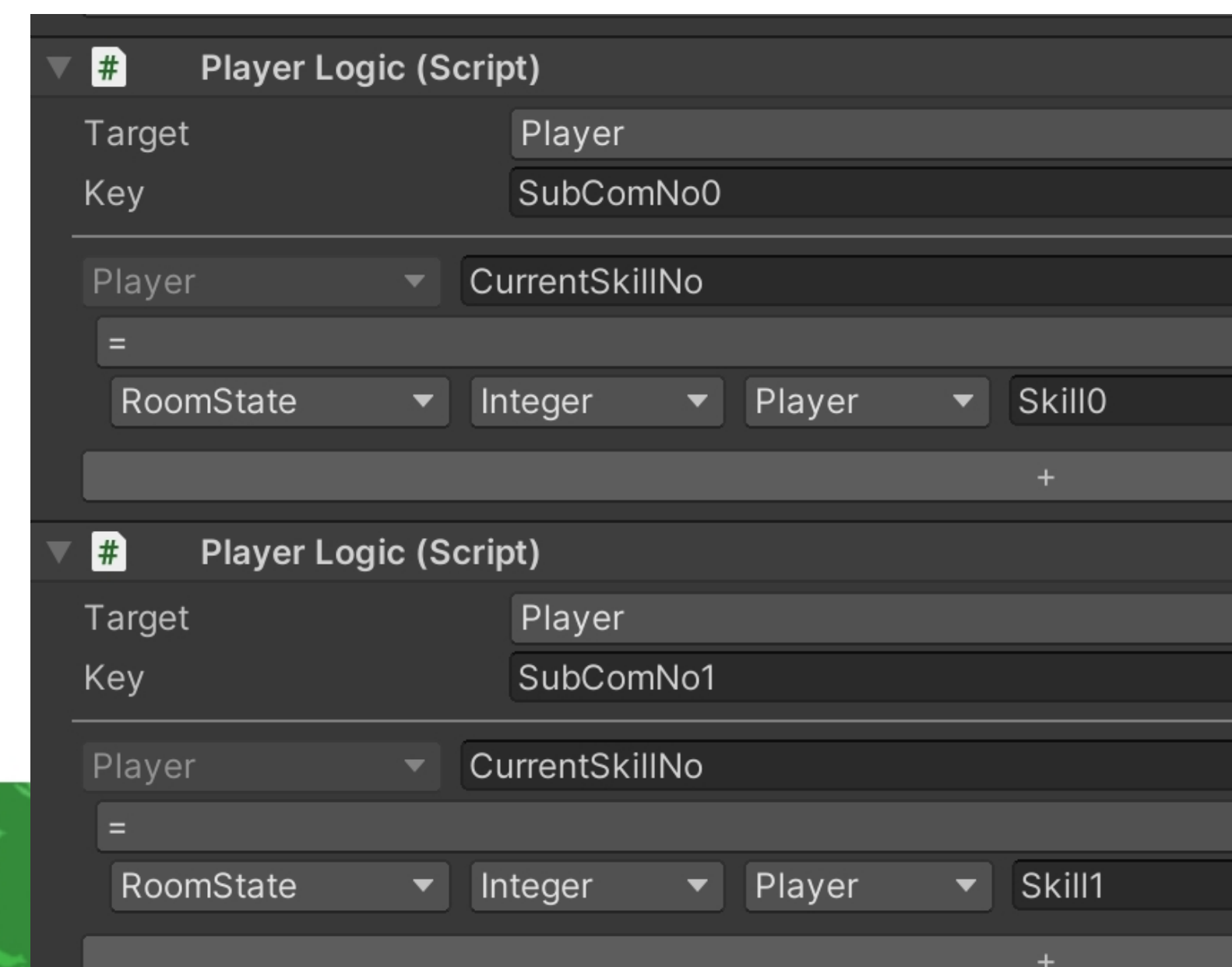
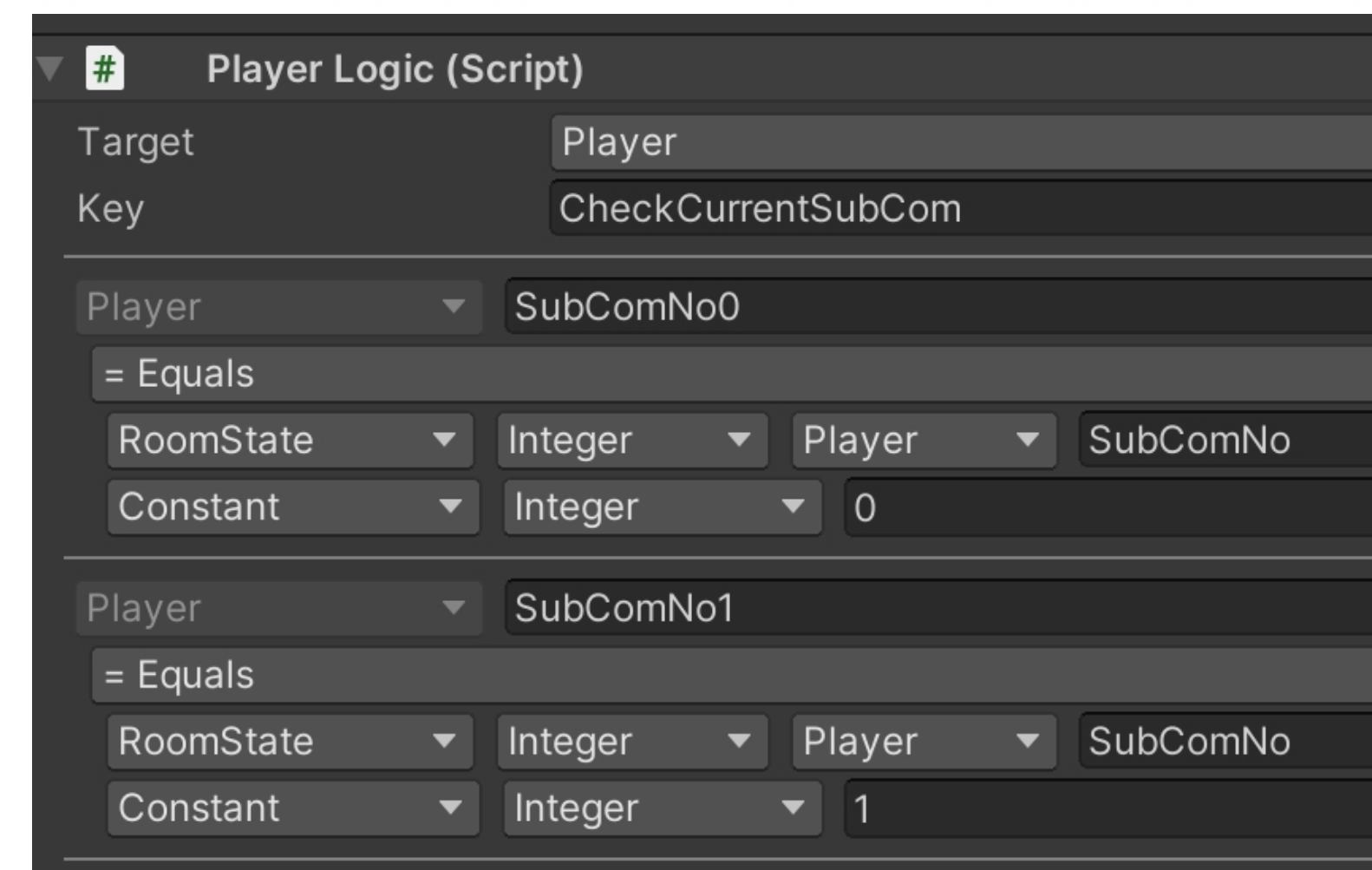
The screenshot shows the 'Player Logic (Script)' configuration window. It contains several logic bricks arranged in a sequence:

- Target:** Player
- Key:** ComXYCheckSub
- Player:** ComXBase (Integer)
- Operation:** = Divide
- RoomState:** Integer
- Player:** CurrentSkillNo
- Constant:** Integer 16
- Player:** ComX (Float)
- Operation:** = Divide
- RoomState:** Float
- Player:** ComXBase
- Constant:** Float 4
- Player:** ComYModulo (Integer)
- Operation:** = Modulo
- RoomState:** Integer
- Player:** CurrentSkillNo
- Constant:** Integer 16
- Player:** ComY (Float)
- Operation:** = Divide
- RoomState:** Float
- Player:** ComYModulo
- Constant:** Float 16
- Player:** ComY (Float)
- Operation:** = Subtract
- Constant:** Float 0.9375
- RoomState:** Float
- Player:** ComY

A plus sign (+) is visible at the bottom right of the window, indicating that the logic sequence can be extended.

**Skill0とかSkill1とかに
数字を入れておくことで、
コマンドの順番は固定じゃ
なくできる**

**レベルが上がるとスキルが
増えるとか、モンスターの
肉を食べるとスキル変わるとか……**



.....



細かいトコはだいぶ
はしよりましたが、
あとはコマンド実行する
ボタンのことについて





乗り物もItemなので、
ボタンは別のItemにして
ParentConstraintについてこさせる

勉強会「乗り物何もわからん!?!」参照

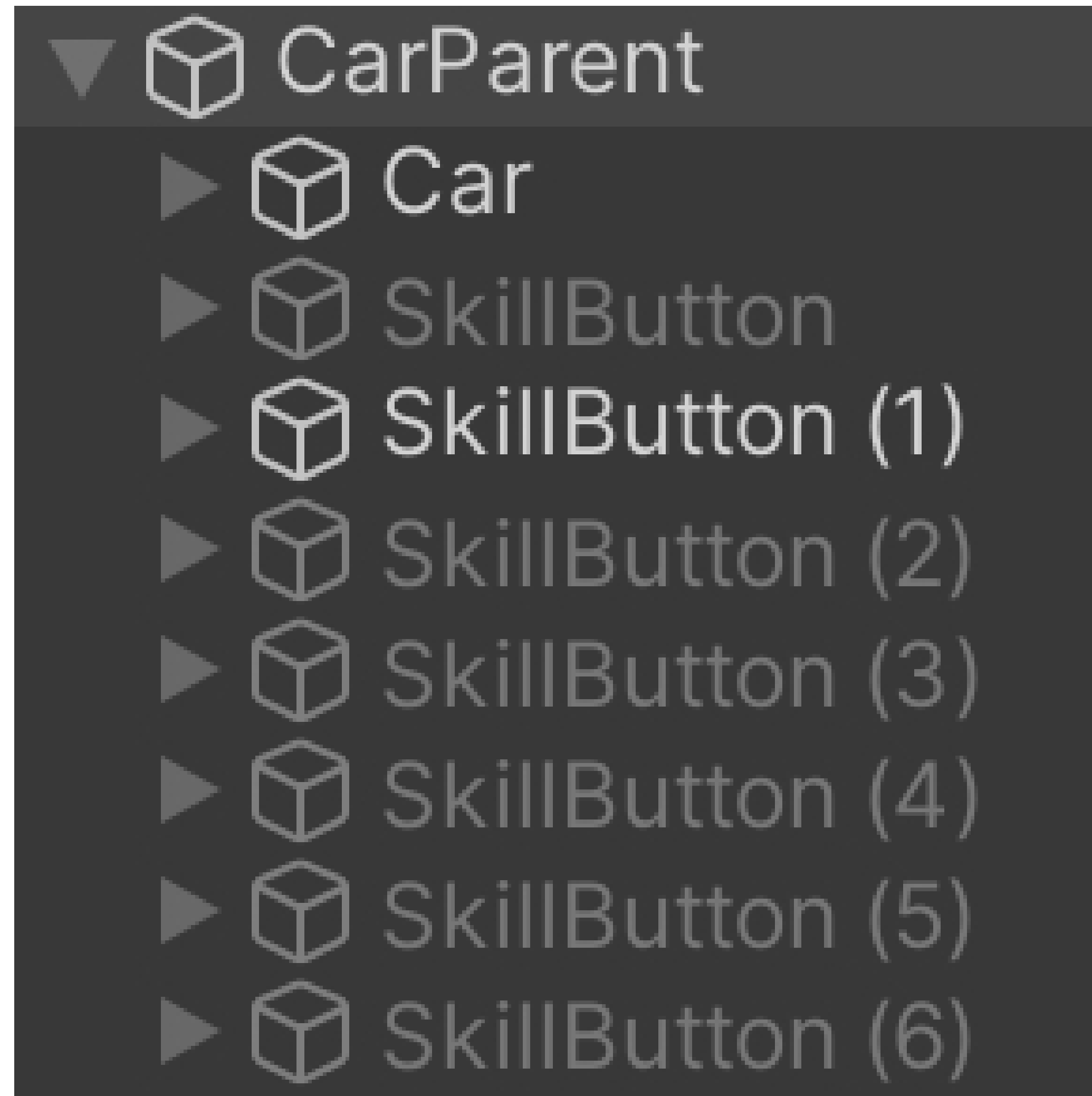
- **プレイヤーがダメージを受けた処理とかは乗り物側**
- **攻撃を出す処理とかはボタン側**

分けられるのは地味にメリット
(1つのItemにLogicが大量につくと大変!)

選択中のコマンドで
表示するボタンを
変更とかすると
もっと分けられるが……

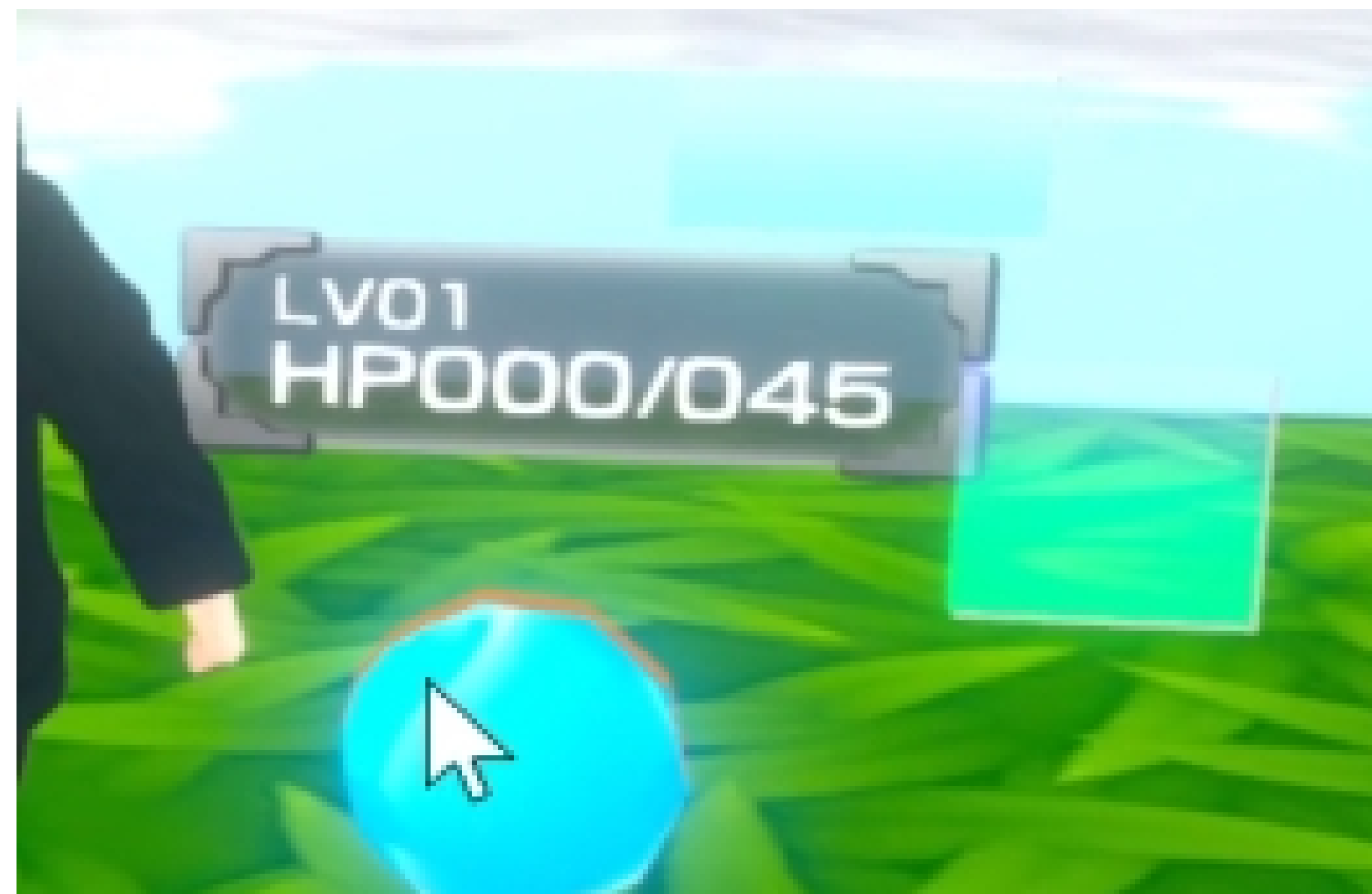


かえって
ややこしいかも？



**ボタンを表示する場所も
地味に悩ましい**

**1人称・3人称・VRで
適切なサイズと位置が
違ってくる……**



2つ付けてみた例

209

青の箱が乗り物なので、箱に乗って緑のエリアに入って下さい。
Unity上では緑のエリアに入るごとに一度数字が増えるのに、アップロードすると
緑のエリアにいる間ずっと数字が上がり続けてしまう.....



あと、どうも
乗り物乗ってるときに
当たり判定が
ヘンになってる？



細かい懸念はあるものの……
従来よりも色々な入力ができることで
表現が広がったのは事実

コマンド式アドベンチャーゲームとかも
いけるはず!

皆さんもパラメータゲージとか
作っていきましょう!

ありがとうございました!