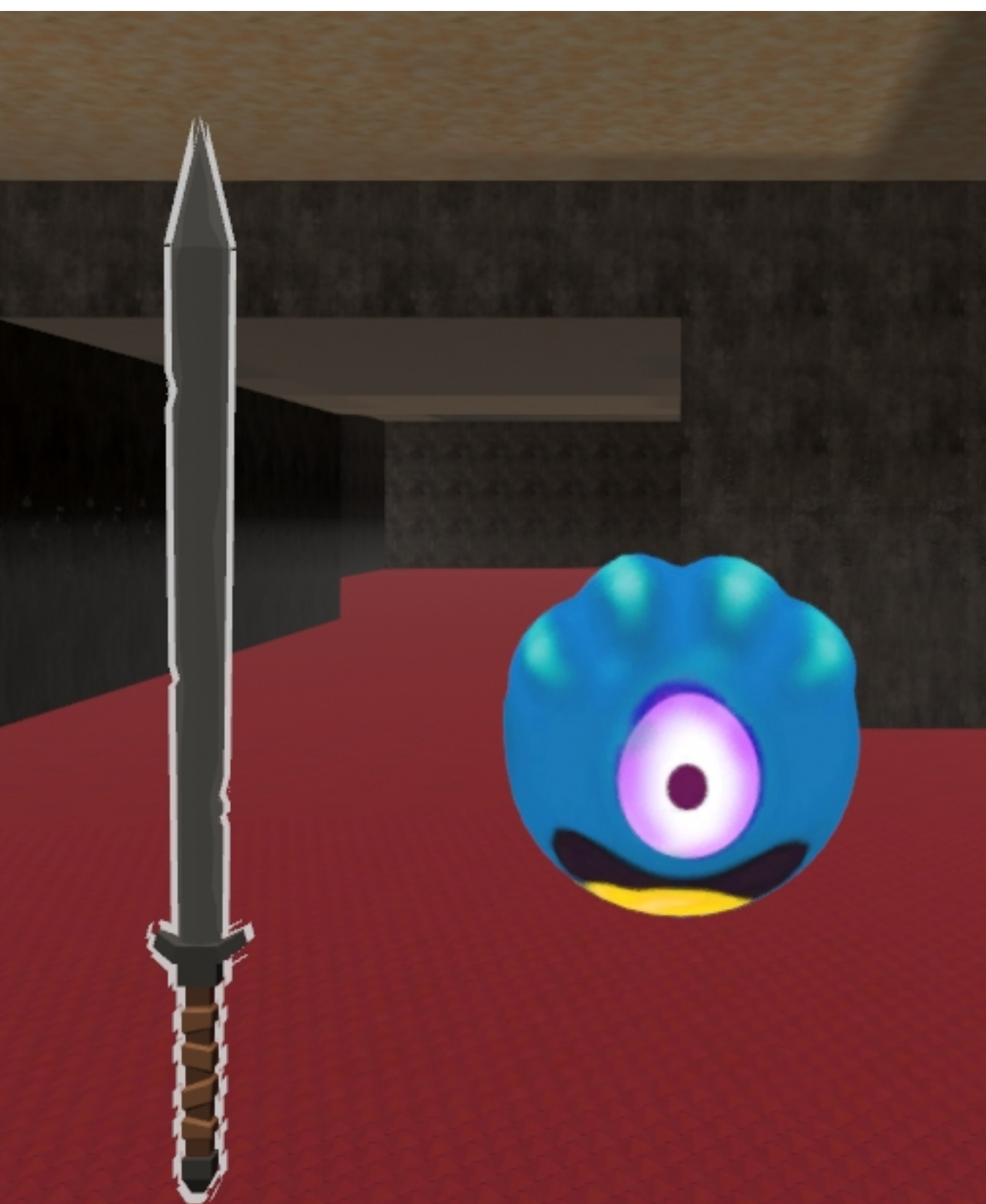


[Twitter @vins_cluster](https://twitter.com/vins_cluster)



Logicのよくあるパターン 剣で戦うRPGイメージで

よく出るLogicのパターン紹介
中級者の人にはもう実装していることが
多いと思いますが.....



掴まずにほっとくと 消えるアイテム

すでにこのネタで記事を
書かれた人もいますが……

OnGrab～で「掴んでいるフラグをON」

OnRelease～で「掴んでいるフラグをOFF」「カウント0」

フラグがOFFなら**ItemTimer**→**ItemLogic**で

時間カウント+1。 ↑これはOnCreate～で開始させる

時間カウントが10以上なら**DestroyItemGimmick**



ItemLogic呼ばなくても、
Valueに入りたい値を
突っ込んでおけば
その値が保存される。

The screenshot displays three Blueprints components in a vertical stack:

- On Create Item Trigger (Script):** Target is 'This', Value is 'Signal', and the target is set to 'TickTimer'.
- Item Timer (Script):** Target is 'This', Key is 'TickTimer', Delay Time Seconds is '1', and Size is '2'. It has two triggers: the first has Target 'This' and Value 'Signal' pointing to 'OnTimer'; the second has Target 'This' and Value 'Signal' pointing to 'TickTimer'.
- Item Logic (Script):** Target is 'Global', Key is 'OnTimer'. The logic is: 'This' (dropdown) -> 'GlabTickInc' (dropdown) -> 'Signal' (dropdown) -> '= Not' (dropdown) -> 'RoomState' (dropdown) -> 'This' (dropdown) -> 'Grabbing' (dropdown).

**TickTimerがループ。
OnTimerで、Logic呼んで、
GrabbingがOFFか
どうかチェックする。**

Item Logic (Script)

Target: This
Key: GrabTickInc

This: This GrabTick Integer

= Add

RoomState: RoomState This: This GrabTick

Constant: Constant Integer 1

This: This Destroy Signal

= GreaterThanOrEqual

RoomState: RoomState This: This GrabTick

Constant: Constant Integer 10

+ +

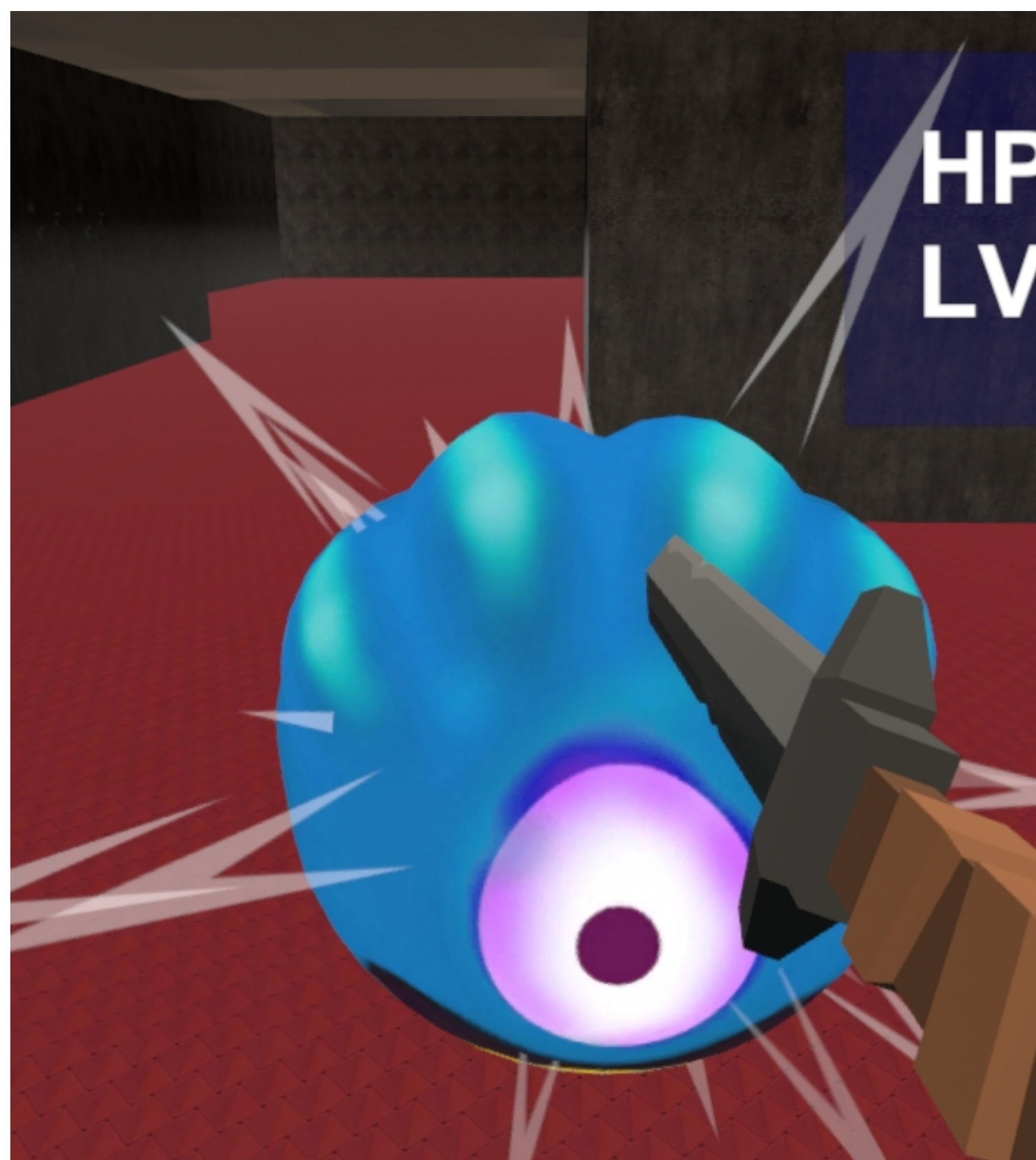
✓ Destroy Item Gimmick (Script)

Target: This
Key: Destroy

**GrabTickを1増やし、
10を越えてたら
DestroyItemGimmick。**

モノによってはDestroyするのではなく、
初期位置に**WarpltemGimmick**
しても良い。

(初期位置に重なる形でEmptyObjectを
置いておく)



剣が敵に当たると ダメージを与える

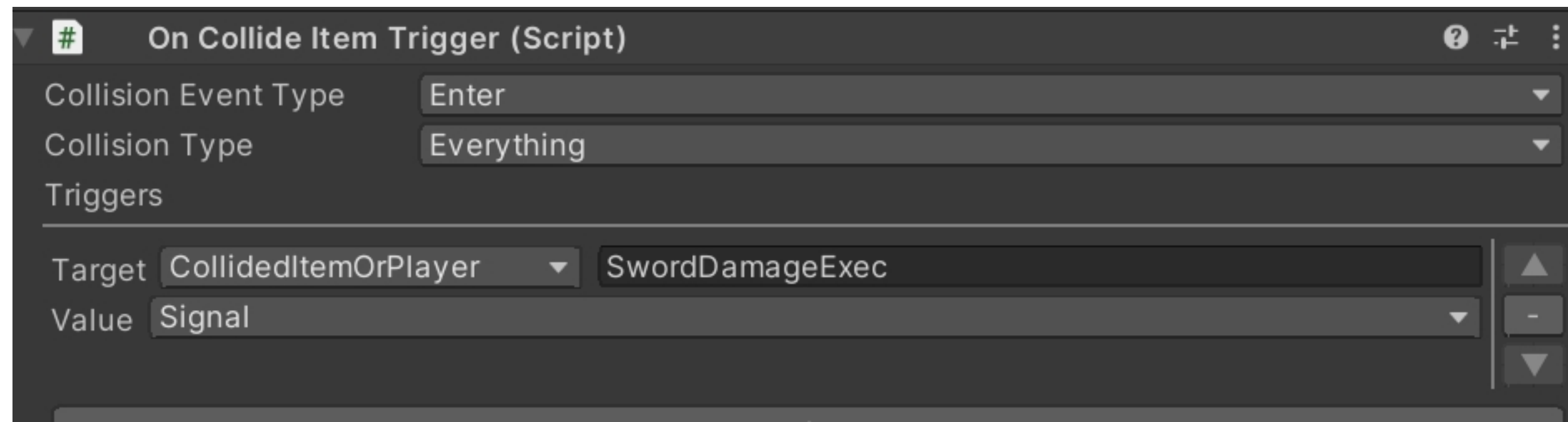
武器の種類がいくつがあっても
やりやすいパターン

フツーツーに、剣が敵に当たったら

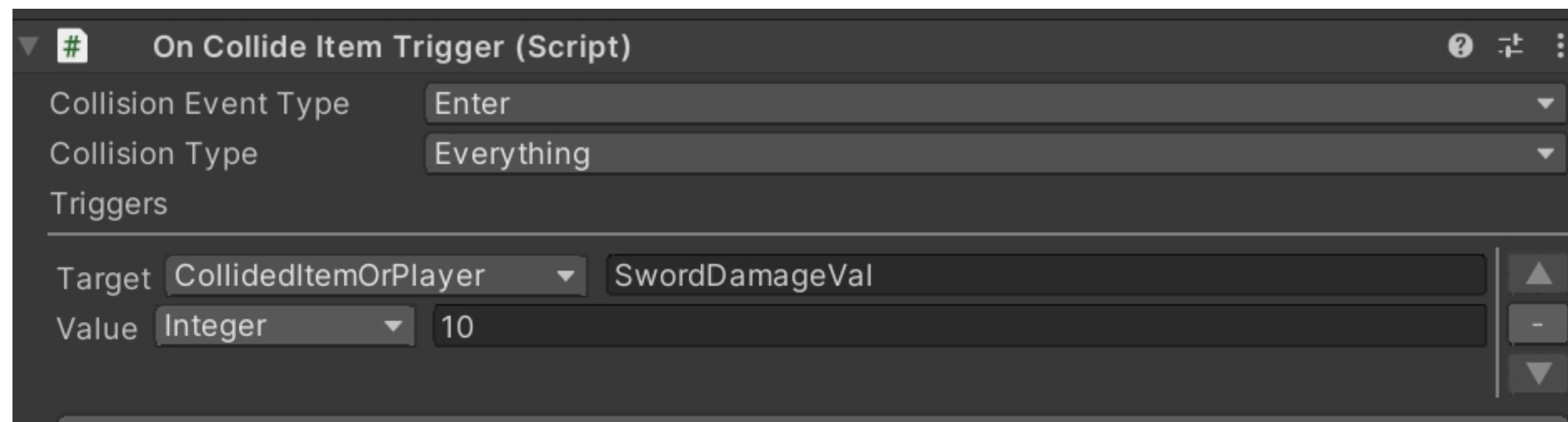
OnCollidedItemTrigger でよい

target は `CollidedItemOrPlayer`

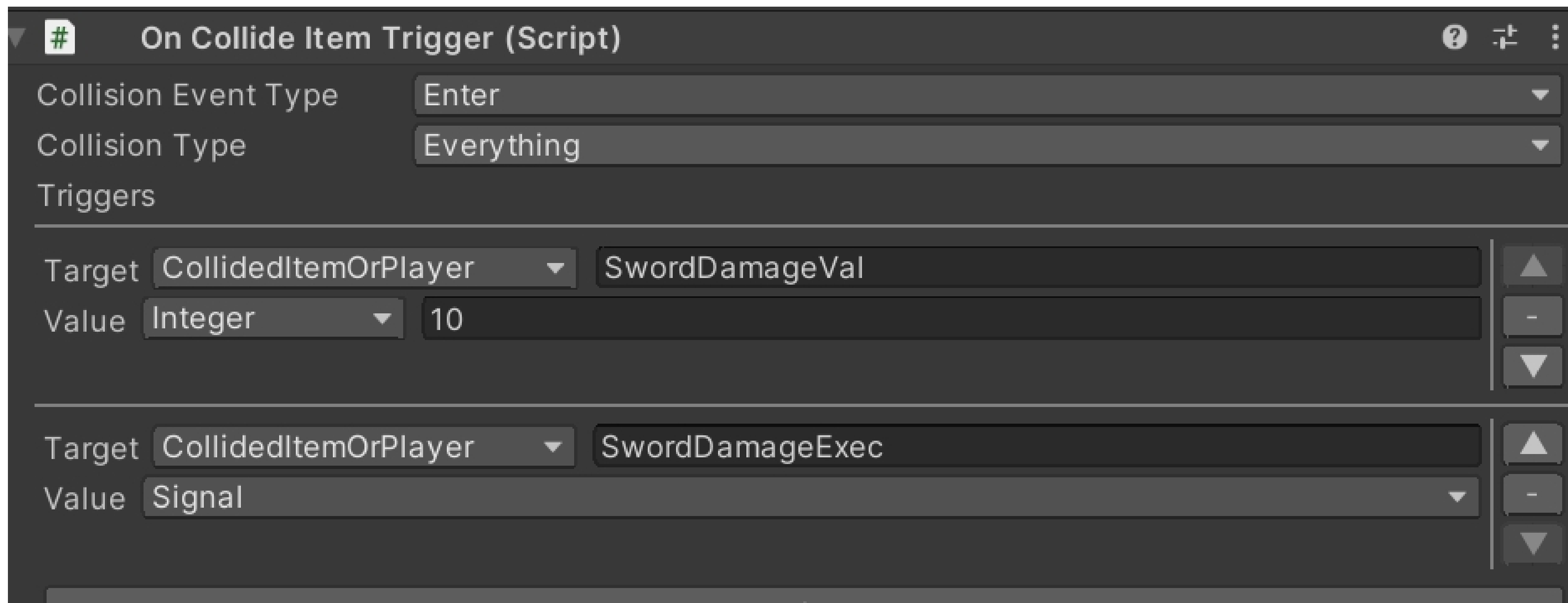
ただ、これだけだと……



Signalを送るだけ
→ **値が設定できない**



Integerを送るだけ
→ **Logic起動しない**



Integerで値を送り、SignalでLogic起動

Item Logic (Script)

Target: This

Key: SwordDamageExec

This HP Integer

= Subtract

RoomState This HP

RoomState This SwordDamageVal

This Hit Signal

=

Constant Bool

This KillExec Signal

= LessThanOrEqual

RoomState This HP

Constant Integer 0

+

Logicは
こんな
感じとか

残念ながら、今のclusterでは
Itemから**OnCollideTrigger**で
相手に固定値しか送れない

なので

PlayerのLVによって
持っている剣の攻撃力を
変えるとかができない……

ただ、**On10Damage**とか

On60DamageとかKey増やしまくるよりは

Integer送る→Signal送る

……の手法がマシ

敵は一度出現したら動かさない
ほうが無難かも？(同期問題)

当たったはずが当たってない!

はストレスになるので……



LVUPさせる

処理と、計算式の例

PlayerLocalUIの

基礎はスキップします

ゆるゆる勉強会#1で

ちょっと取り上げたんで

On Join Player Trigger (Script)

Triggers

Target Player OnJoin

Value Signal

Player Logic (Script)

Target Player

Key OnJoin

Player LV Integer

=

Constant Integer 1

Player MHP Integer

=

Constant Integer 20

Player EXP Integer

=

Constant Integer 0

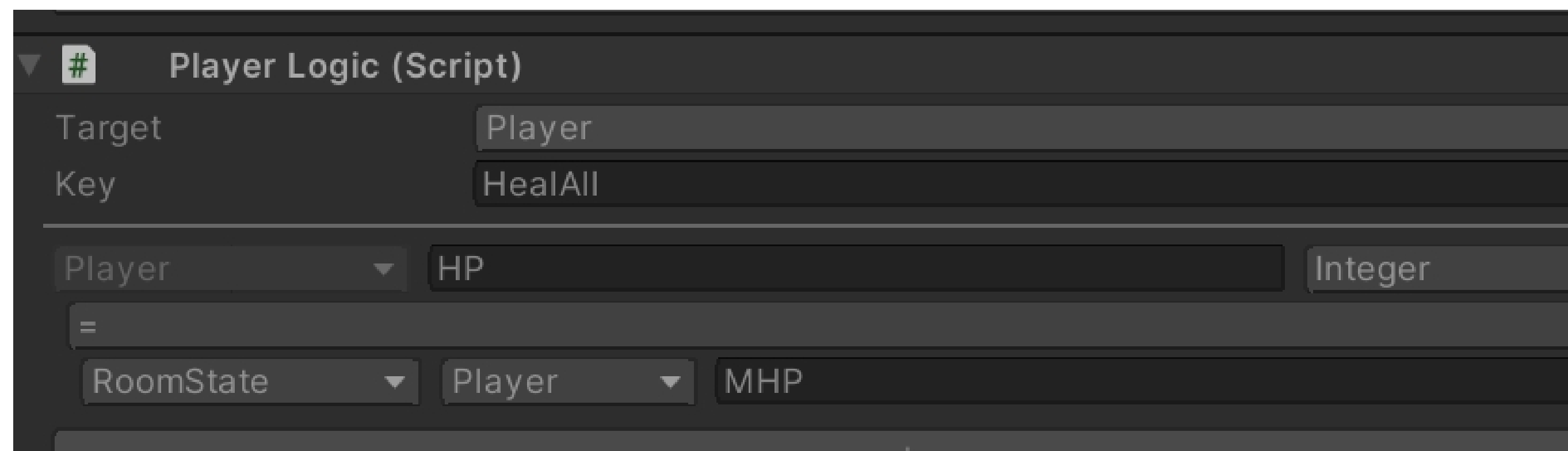
Player HealAll Signal

=

Constant Bool

とりあえず
初期化は
こんなので

回復は初期化の
とき以外も使うので
別Logicに



ちなみに**PlayerLogic**をつけるのは
適当なEmptyObjectでよい
(いくつかに分けることをオススメ)

Player Logic (Script)

Target: Player

Key: GotEXPCheck

Player EXP Integer

= Add

RoomState Player EXP

RoomState Player GotEXP

Player LVUpExec Signal

= GreaterThanOrEqualTo

RoomState Player EXP

RoomState Player NextLVUpVal

肝心の LVUP 処理

こいつが問題

Player Logic (Script) ?

Target: Player

Key: NextLVUpValCheck

Player: NextVal0 Integer

= Multiply

RoomState: Player LV

RoomState: Player LV

Player: NextVal1 Integer

= Multiply

RoomState: Player NextVal0

RoomState: Player NextVal0

Player: NextVal2 Integer

= Multiply

RoomState: Player NextVal1

Constant: Integer 10

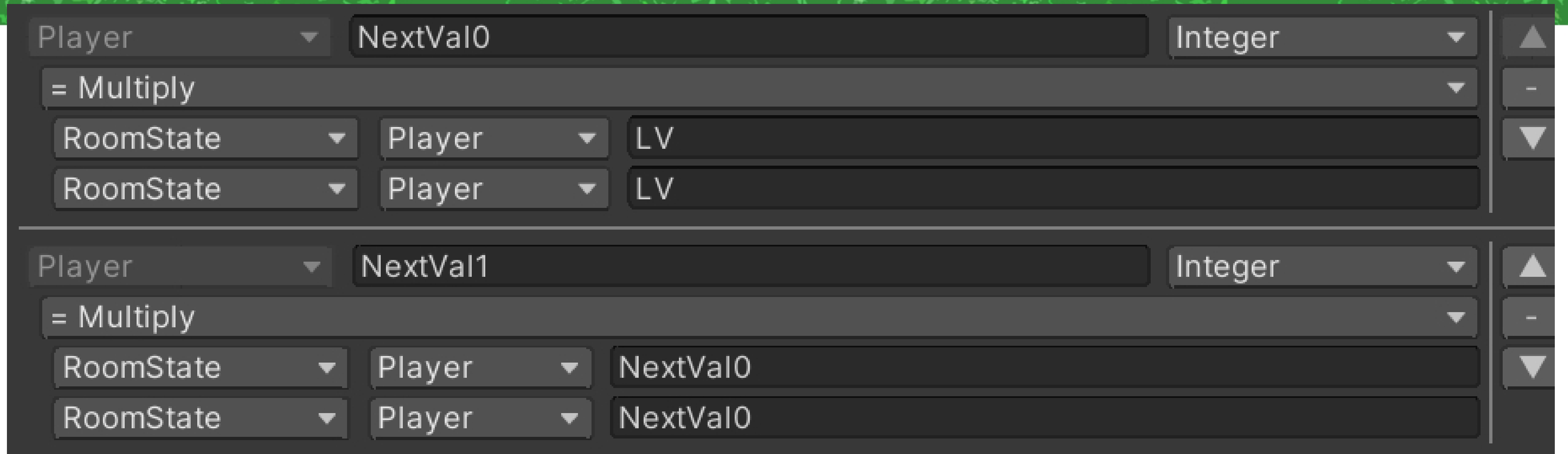
Player: NextLVUpVal Integer

= Add

RoomState: Player NextVal2

Constant: Integer 40

こんなの



前半 $(LV \times LV)^2$

LV1.....1

LV2.....16

LV3.....81

Player	NextVal2	Integer	▲
= Multiply ▼			
RoomState	Player	NextVal1	▼
Constant	Integer	10	

Player	NextLVUpVal	Integer	▲
= Add ▼			
RoomState	Player	NextVal2	▼
Constant	Integer	40	

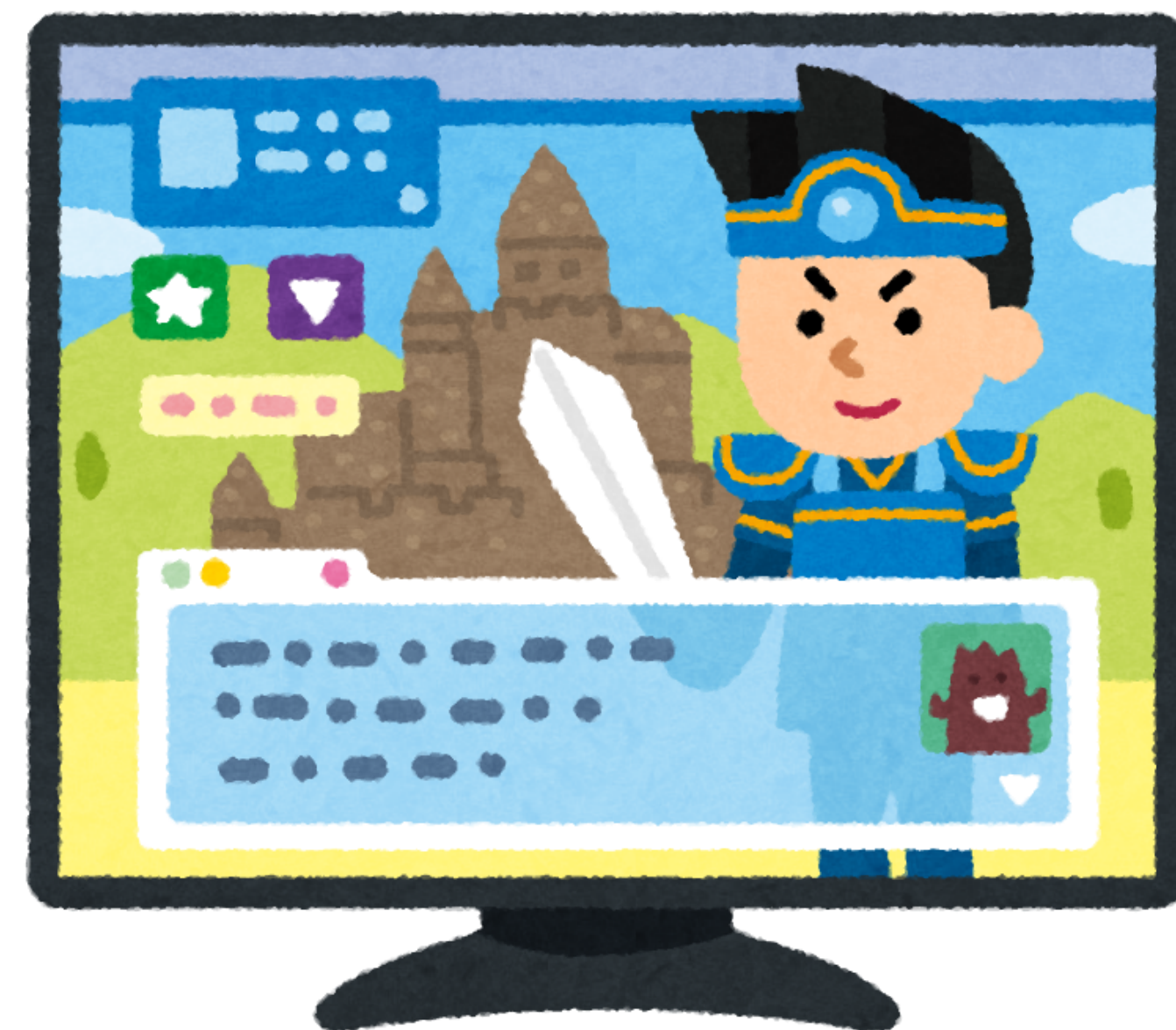
$$(LV \times LV)^2 \times 10 + 40$$

- LV1.....10+40**
- LV2.....160+40**
- LV3.....810+40**

LV1	10+40	=	00050
LV2	160+40	=	00200
LV3	810+40	=	00850
LV4	2560+40	=	02600
LV5	6250+40	=	06290
LV6	12960+40	=	13000

割と急激に
伸びていく

LVが上がったらわかりやすく
強くする。そして**次の稼ぎ場で**
稼ぐほうが効率良くなるようにする

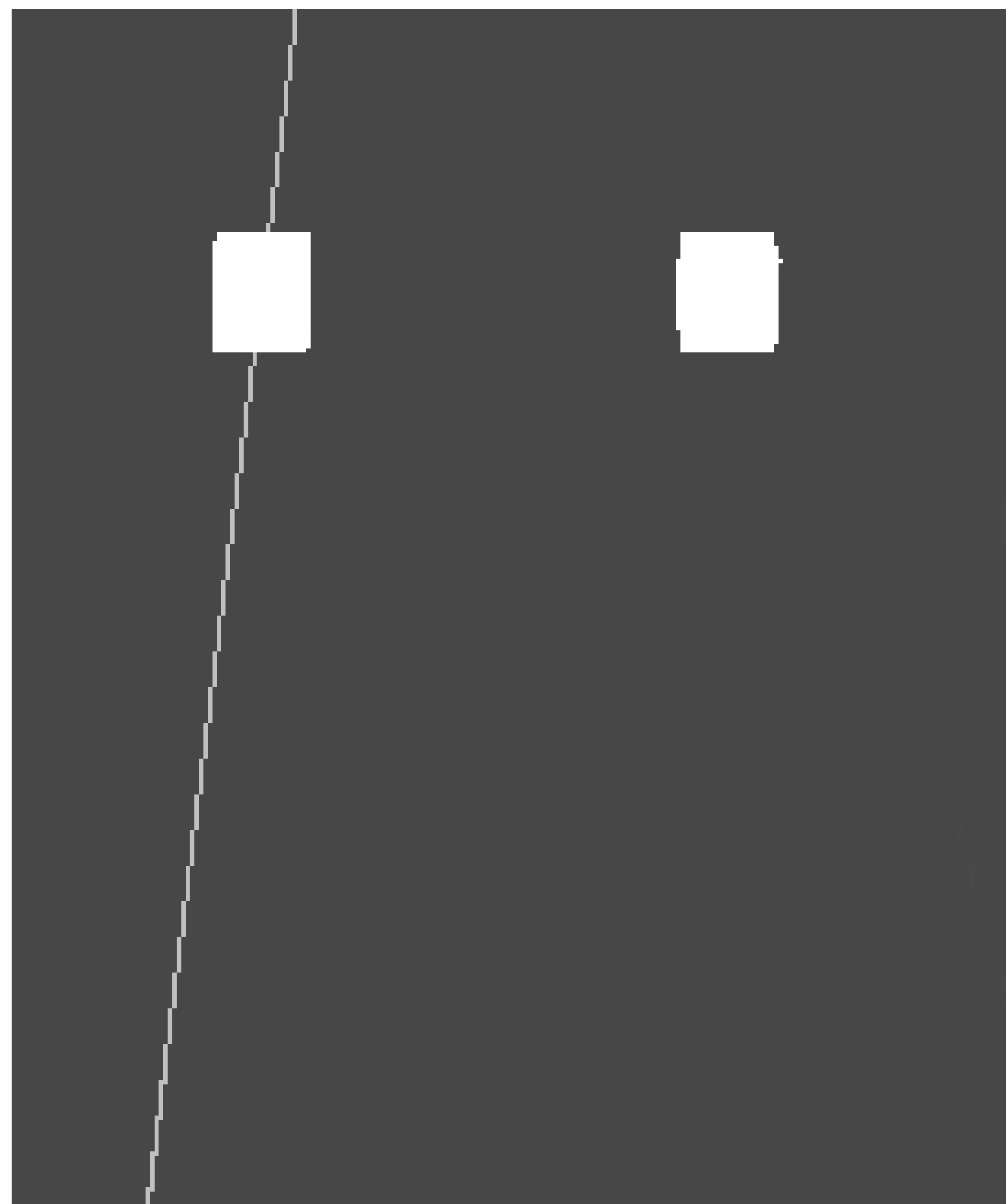


→**テンポのいいゲーム感**をめざしてみる

The image shows a state machine editor interface with three states, each representing a different HP calculation logic:

- State 1 (HPVal0):** The transition is triggered by 'Player' and results in 'Integer'. The operation is '= Multiply'. The inputs are 'RoomState' (Player) and 'LV'.
- State 2 (HPVal1):** The transition is triggered by 'Player' and results in 'Integer'. The operation is '= Multiply'. The inputs are 'RoomState' (Player) and 'HPVal0'. A 'Constant' input of 'Integer' with the value '6' is also present.
- State 3 (MHP):** The transition is triggered by 'Player' and results in 'Integer'. The operation is '= Add'. The inputs are 'RoomState' (Player) and 'HPVal1'. A 'Constant' input of 'Integer' with the value '14' is also present.

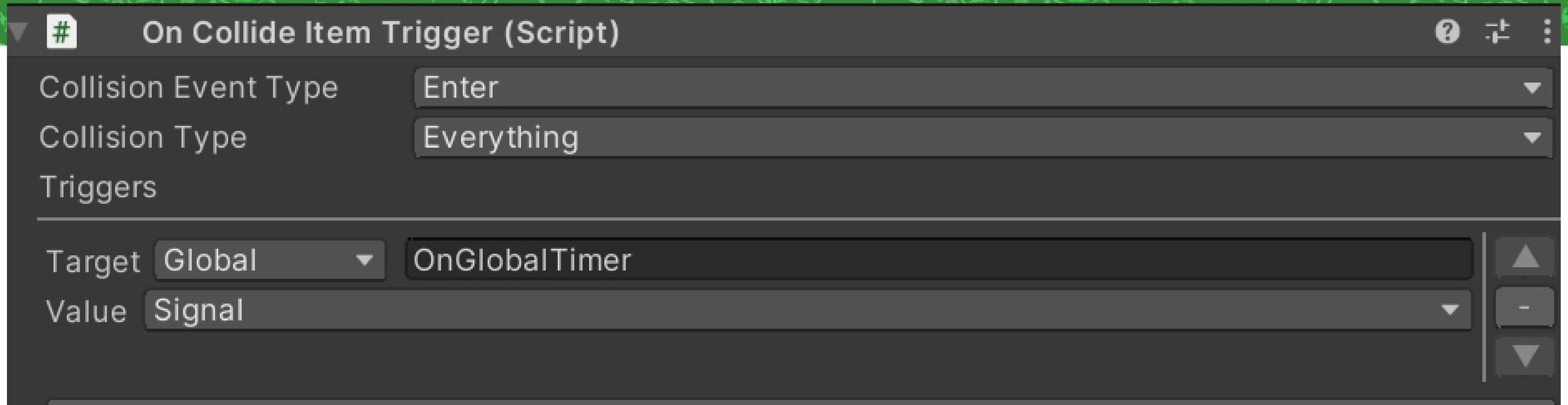
HPとかかも似たような計算で



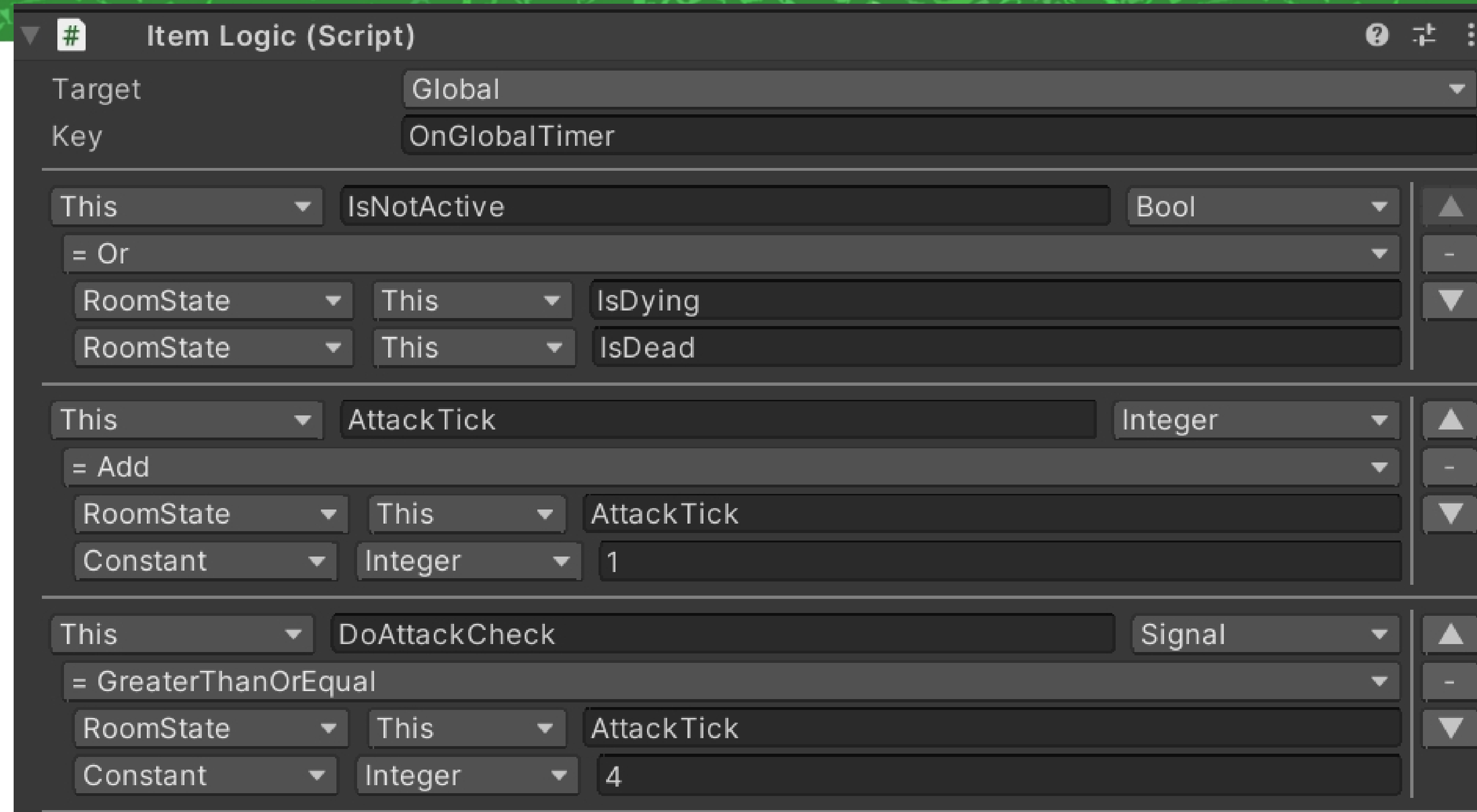
(時間があったら)

サンフラワーふじさん式

Timerの活用



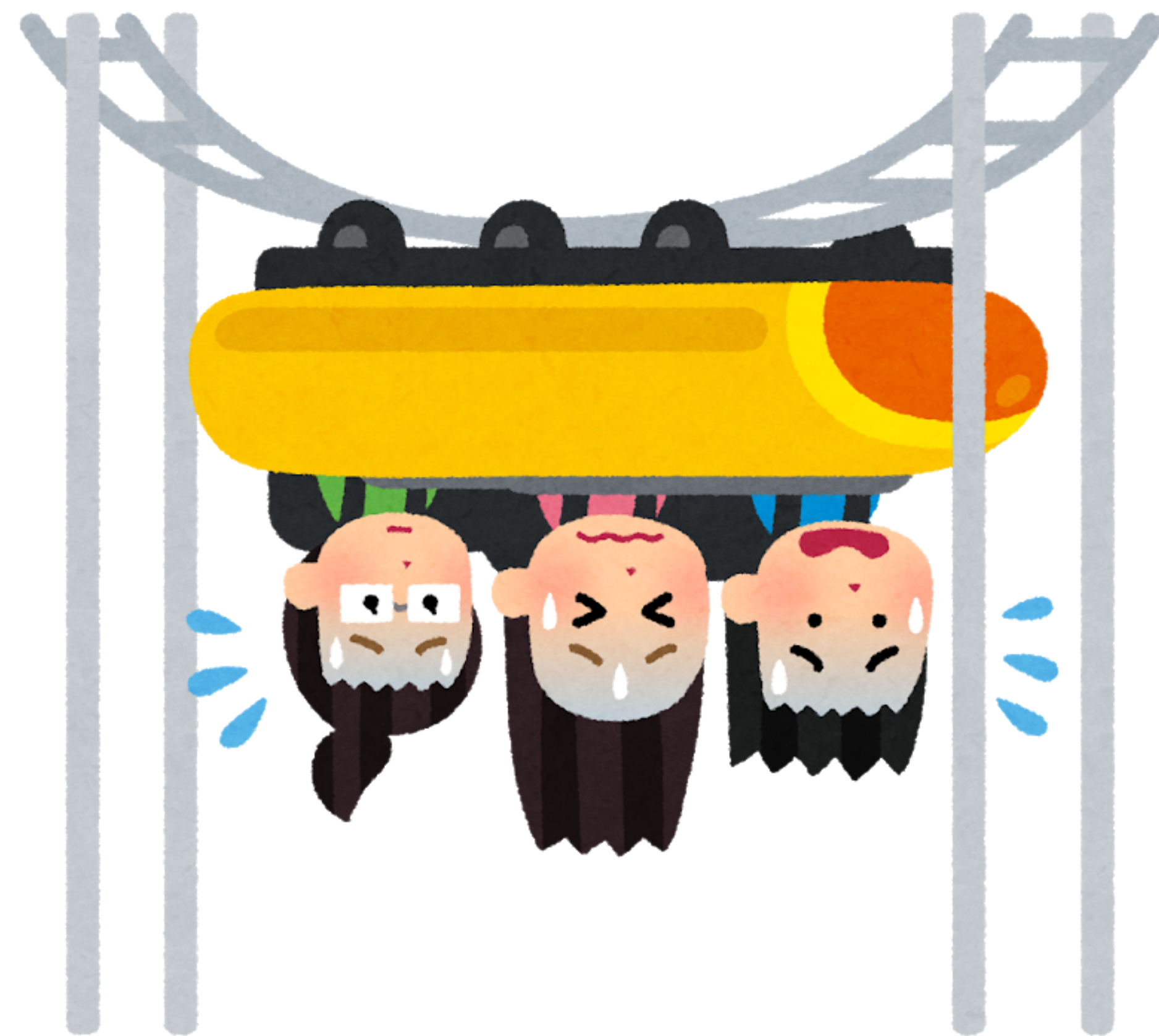
2つCube置いて1秒ごとにぶつかるように
Animator(+Timeline)設定だけ
(そしてGlobalにSignalを送る)



あとはTarget GlobalのItemLogicとかで受けるだけ

ゆるゆる勉強会#2でもあった通り、
clusterのTimerはたまに
発生しないことがある

Timerが発生しない
→そこでループが
止まるおそれ?



Cubeが1秒ごとにぶっかかり続ける
サンフラワーふじさん式Timerなら

途中で取りこぼしがあっても

次でフォローできる

**Timelineは全Player間で
再生位置が同期されるので、
精度がより高そうというメリットも**

●「4秒後に……」とかしきたい場合、

Logic書かないといけない

●「0.2秒後に……」とかしきたい場合、

間隔を短くしないといけない

絶対こぼしたくないTimerに
採用するといいかも？

(最近のclusterは同期の

精度が良くなったので以前ほどは

気にしなくてもいいかもしれない……)



ありがとうございました!