



# Cluster Game HUD v2 はじめよう

**何ができるのか？**

- 追加の「ボタン」
- アバターモーションを扱いやすい
- カメラをいくらかきいじれる
- カメラの位置を取れる
- プレイヤーにメッセージ送れる etc

# 追加の「ボタン」



PCだと左・右クリック  
そして[E][R]キー  
スマホでは右下に  
4つまでボタンが出る

# **VRでは特に何も起きないことに注意**

→ただしVRかどうかを判定できるように  
なったので、VRのときだけ立体のボタンを  
表示させるなどの手はある

(ぶっちゃけVRでもコントローラーのトリガーとか  
ボタンに割り当てちゃっていい気が……)

**どうすればボタンを出せるのか？**

●ベータ機能をONにする

●WorldRuntimeSettingsで  
「Use Cluster HUD v2」をONに



# PlayerScriptがないと ボタンを表示させることはできない



## ScriptableItemとセットでつける

持てるアイテムとか乗り物なら  
OnGrab (持ったとき)とか  
OnRide (乗ったとき)に  
**\$.setPlayerScript(player)**

```
$.onRide((isGeton, player) => {  
  if(isGeton) {  
    $.setPlayerScript(player);  
  }  
})
```

**何も持たず乗らずにPlayerScriptを  
付けたい場合は？**

ワールド入り口あたりにボタンとか作る

→押されたとき、プレイヤーを別のところへ飛ばす

→同時に**CreateItemGimmick**で

**PlayerScript**付きのアイテムを同じ場所に生成

→そのアイテムが**onUpdate**から近くのプレイヤーを探す!

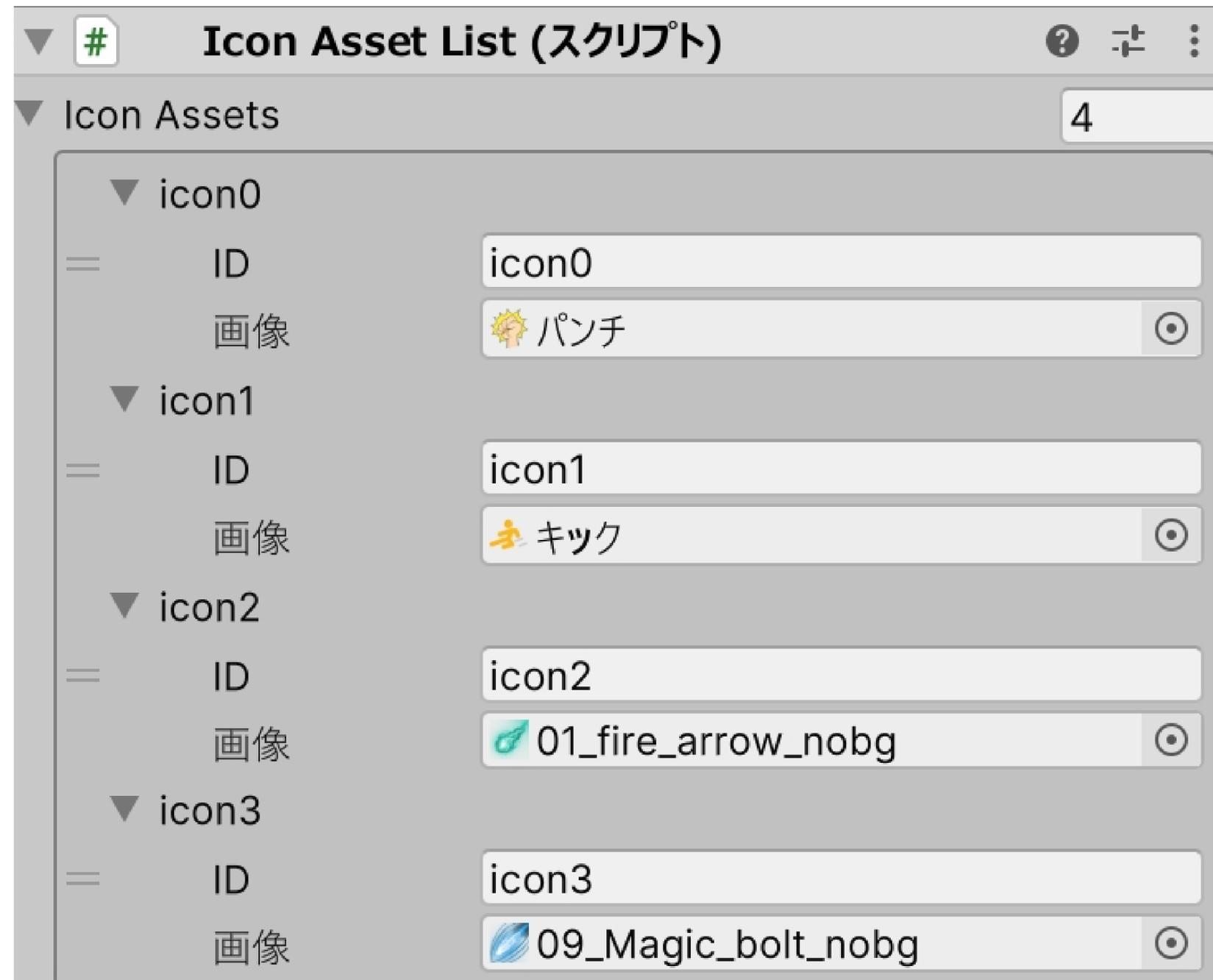


```
const findPlayer = () => {  
  let players = $.getPlayersNear($.getPosition(), 0.3);  
  if (players.length > 0) {  
    $.state.player = players[0];  
    $.setPlayerScript(players[0]);  
    $.state.playerFound = true;  
  }  
};
```

PlayerScriptの例。\$ではなく\_を使うのがポイント  
あと、アイテムの情報も保存しておく  
そしてアイテム側にVRかどうかを伝えておく

```
_.onStart(() => {  
  _.showButton(0, _.iconAsset("icon0"));  
  _.showButton(1, _.iconAsset("icon1"));  
  _.showButton(2, _.iconAsset("icon2"));  
  _.showButton(3, _.iconAsset("icon3"));  
  targetItem = _.sourceItemId;  
  
  _.sendTo(targetItem, "isVR", _.isVr);  
});
```

アイコンは同じアイテムに  
**IconAssetList**  
コンポーネントをつけると  
扱えるようになる



ScriptableItem側でメッセージを受信する例  
最後の**player: true**というのを書かないと  
**PlayerScriptからのメッセージを受け取れない!**

あとはownerに  
メッセージを送って  
「VR非対応です」って  
メッセージ出すとか

```
$.onReceive((messageType, arg, sender) => {  
  if (messageType == "isVR") {  
    $.setStateCompat("owner", "isVR", arg);  
    $.setStateCompat("owner", "isNotVR", !arg);  
  }  
}, {  
  player: true  
});
```

# PlayerScriptのonButtonの例

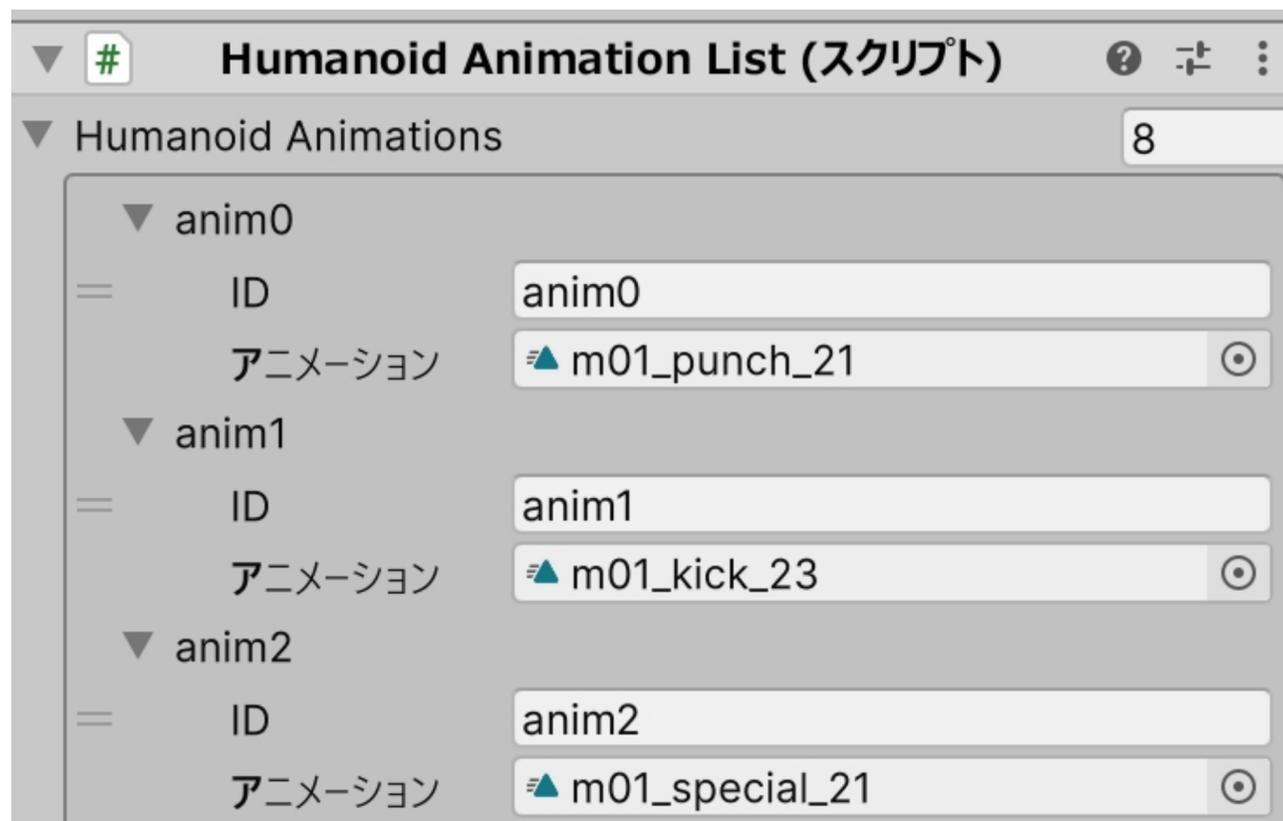
## ボタンの番号0~3までそれぞれ個別に書く

```
let motionNo = -1;  
let motionTick = 0;
```

ちなみに変数は  
\$.stateとか使わず  
グローバル変数でよい

```
_.onButton(0, (isDown) => {  
  if (isDown) {  
    motionNo = 0;  
  }  
});
```

# アニメーションはHumanoid Animation Listで登録して\_.\_humanoidAnimationで登録



```
const motions = [  
  _._humanoidAnimation("anim0"),  
  _._humanoidAnimation("anim1"),  
  _._humanoidAnimation("anim2")  
];
```

# \_.onFrameは\$.onUpdateとほぼ同じ

```
_.onFrame((deltaTime) => {  
  if (motionNo >= 0) {  
    motionProc(deltaTime);  
  }  
});
```

# モーションをつける処理

```
const motionProc = (deltaTime) => {  
  let motion = motions[motionNo];  
  let len = motion.getLength();  
  motionTick += deltaTime;  
  if (motionTick > len) {  
    |   motionTick -= len;  
  }  
  let pose = motion.getSample(motionTick);  
  _.setHumanoidPoseOnFrame(pose, 1);  
}
```

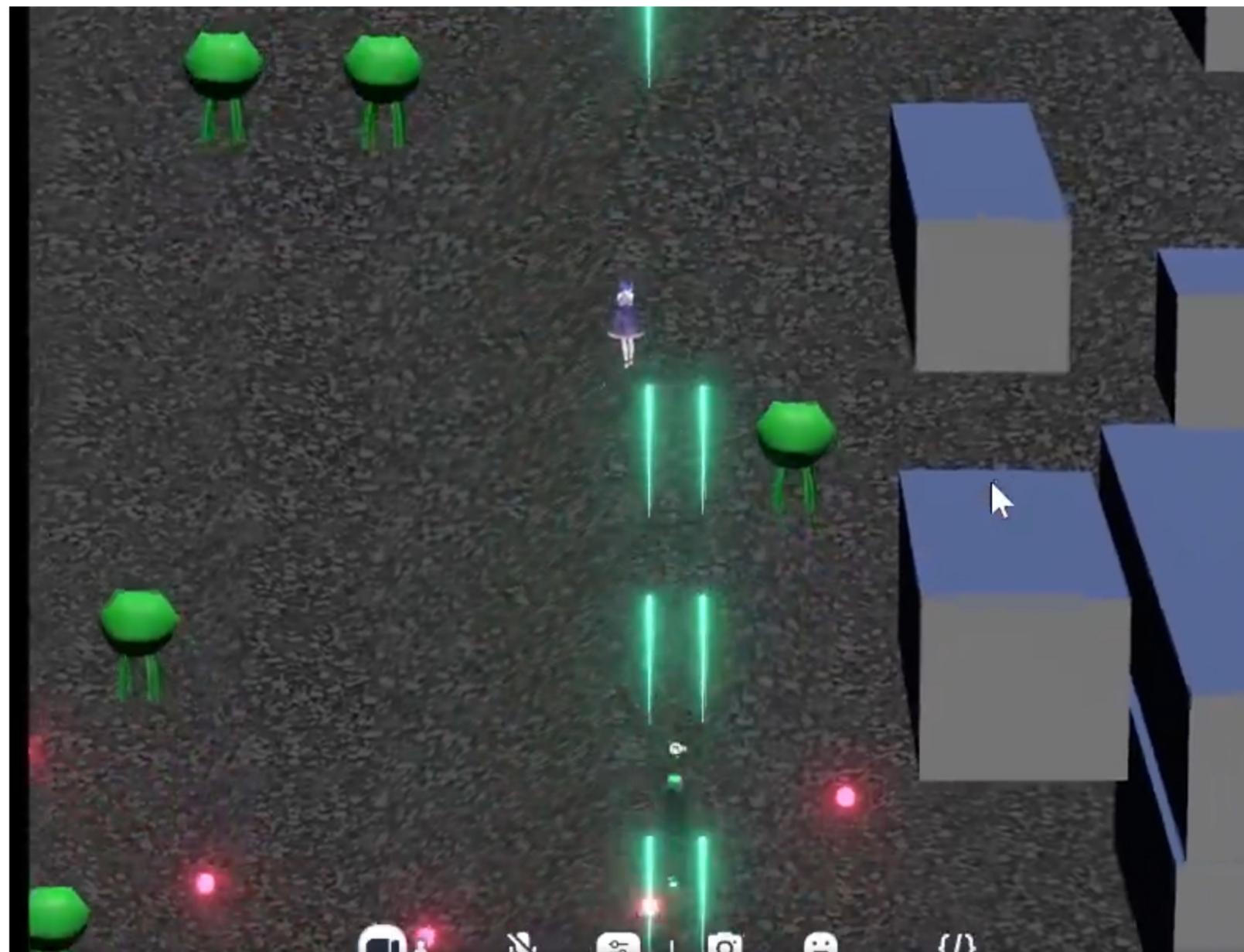
モーションにとって  
長さもとって  
時間加算して  
長さ超えてたら  
最初に戻して  
現在のポーズとって  
最後に割り当てる

**カメラの位置を取れるのも  
興味深かったりする**

**視線だけで撃つ方向を決めるとか**

**※ただし計算は自前でやらないとダメ**

カメラ位置を  
固定はできないが  
ある程度までは  
コントロール可能



というわけで説明しきれませんが  
ボタンとモーションあたりから  
どんどんGame HUD v2  
やっけていきましょう

**ありがとうございました!**