

BattleCart

CameraRotation（簡易 Ver.）

マウス感度
mouseSensitivity = 3.0f

垂直方向の角度限界
minVerticalAngle = -15.0f
maxVerticalAngle = 15.0f

水平方向の角度限界
minHorizontalAngle = -15.0f
maxHorizontalAngle = 15.0f

プレイ中の角度
verticalRotation
horizontalRotation

Start

カーソルの調整

Cursor.lockState をロック

Cursor.visible を false

LateUpdate

plyaing でなければ Return

マウス入力値の取得

mouseX = マウスの横値 * mouseSensitivity

mouseY = マウスの縦値 * mouseSensitivity

マウス値の蓄積と絞り込み

horizontalRotation += mouseX

horizontalRotation を Clamp で絞る

VerticalRotation -= mouseY

VerticalRotation を Clamp で絞る

カメラにマウス値を反映

transfrom.rotation = Quaternion.Euler(verticalRotation, horizontalRotation, 0)



X 軸
verticalRotation

Y 軸
horizontalRotation

CameraController

Player とカメラの距離
diff

Player 情報
player

カメラの追従スピード
flowSpeed = 8.0f

カメラの初期位置・初期回転値
defaultPos = new Vector3(0,6,-6)
defaultRotate = new Vector3(12,0,0)

Start

カメラの初期値を変数で決めた値にする

transform.position = defaultPos

transform.rotation = Quaternion.Euler(defaultRotate)

プレイヤー情報の取得

player = Player タグを持つオブジェクト

プレイヤーとカメラの距離感を取得

diff = (player の position) - (カメラの position)

LateUpdate

Player がいなければ Return

カメラの位置を決定

transfrom.potision = Vector3.Lerp(transform.position, player の位置 - diff, followSpeed * デルタタイム)

RoboSuvivor

CameraController

Player とカメラの距離
diff

Player 情報
player

カメラの追従スピード
followSpeed = 8.0f

カメラの初期位置・初期回転値 (例)

```
defaultPos = new Vector3(0,3.5f,-2)  
defaultRotate = new Vector3(20,0,0)
```

マウス感度

mouseSensitivity = 3.0f

垂直方向の角度限界

```
minVerticalAngle = -15.0f  
maxVerticalAngle = 15.0f
```

※水平方向 (ロボットの回転)
は絞るべきではないので不要

プレイ中の角度

verticalRotation

※水平方向の力は
そのまま Player の回転値として
指定できるので
horizontalRotation は設置しなくても OK

Start

カーソルの調整

Cursor.lockState をロック

Cursor.visible を false

カメラの初期値を設定

transform.position = defaultPos

transform.rotation = Quaternion.Euler(defaultRotate)

プレイヤー情報の取得

player = Player タグを持つオブジェクト

プレイヤーとカメラの距離感を取得

diff = (Player の position) - (カメラの position)

LateUpdate

playing でなければ Return

Player がいない場合は Return

マウス入力値の取得

mouseX = マウスの横値 * mouseSensitivity

mouseY = マウスの縦値 * mouseSensitivity

プレイヤーの左右回転 (Y 軸のみ)
mouseX と連動してプレイヤーの角度を決めてしまう

player.transform.Rotate() を使う

垂直方向のマウス値の蓄積と絞り込み

VerticalRotation -= mouseY

VerticalRotation を Clamp で絞る

カメラの行くべき位置
= Player の現在地から 距離 diff だけ離れた位置
※ただし diff は Player の角度もかけ合わせた位置

Vector3 targetCameraPosition =

Player の現在 position

- (マイナス)

Player の向き (rotation) も考慮した上での
Player とカメラのとりべき距離感


diff
*
player.transform.rotation

カメラの位置を決定

```
transform.position = Vector3.Lerp(  
    transform.position,  
    targetCameraPosition,  
    followSpeed * デルタタイム )
```

カメラの向くべき角度
= 横方向はすでに決定した Player の角度
縦方向は mouseY (をさらに絞った値) で決めた角度
のかけ合わせ

Quaternion targetRotation =

<Quaternion 値> Player の左右の角度 (Y 軸)
Quaternion.Euler(0,player.transform.eulerAngles.y,0);

*

<Quaternion 値> カメラの上下の角度 (X 軸)
Quaternion.Euler(verticalRotation,0,0);

カメラの回転値を決定

```
transform.rotation = Vector3.Slerp(  
    transform.rotation,  
    targetRotation,  
    followSpeed * デルタタイム )
```