TouchController

A modernized joypad emulation interface for touchscreen devices
based on the ShiVa 1.9.2 iPhoneCameraControl sample

Felix Caffier, 2017-02-06
www.trisymphony.com

License

Quick start

1. Drop TouchController AI into your User AI stack
2. Drop TouchController HUD into your game resource tab
3. Set a name for TouchController.sTargetAIModel for event forwarding
4. Call TouchController.onEnable(true) to activate everything

Demo

Included in a separate STE, you will find a demo that includes a sample "sTargetAIModel" AI which demonstrates the forwarding process. All output is logged to the console. When you release a real product, it is highly advised that you do not log the results of the sticks and buttons for performance reasons.

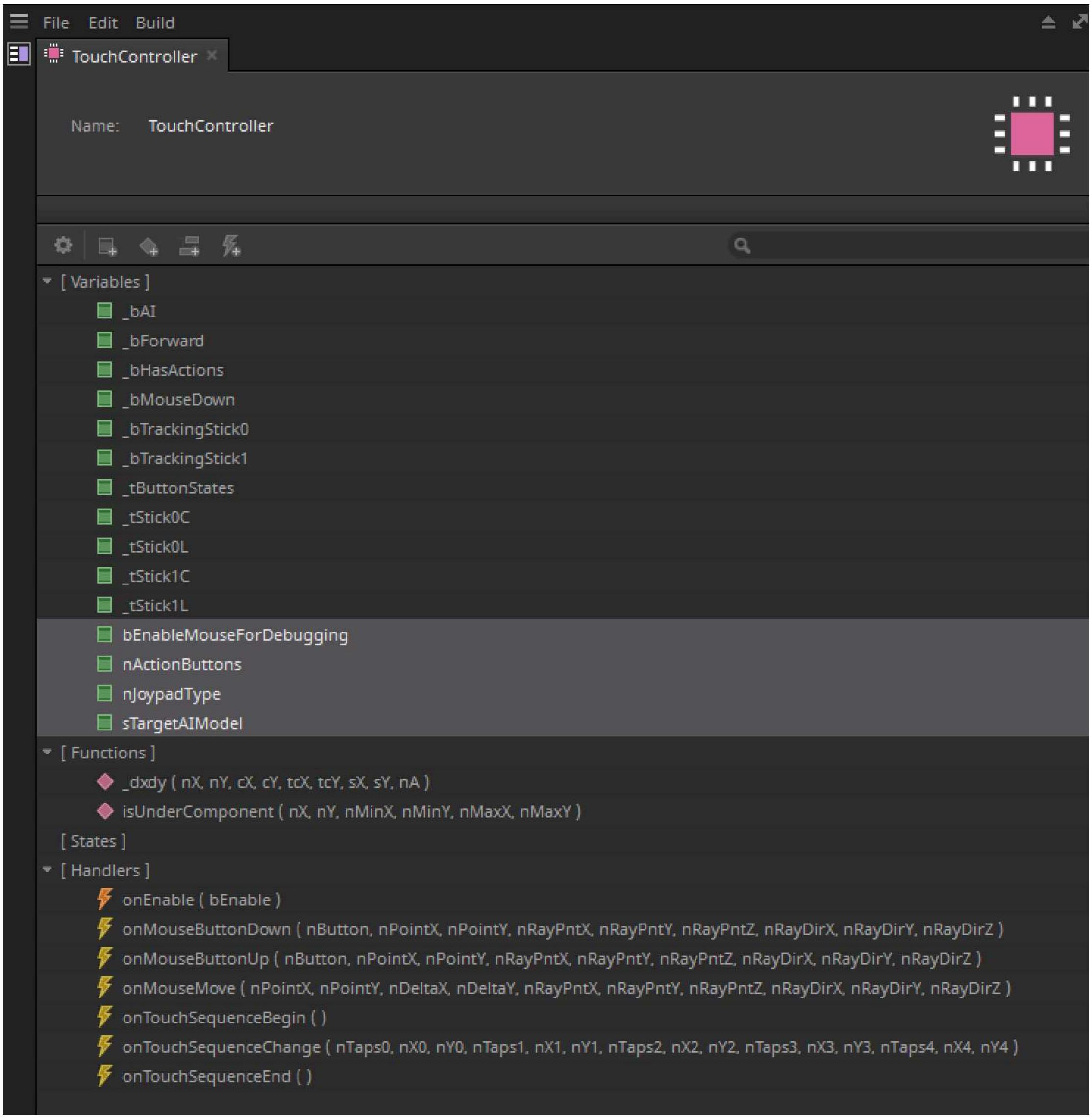
To run the demo, start the game and press "1" to activate, and "0" to deactivate the touch controller.

TouchController AI description

Variables with an underscore as prefix (i.e. "_bAI") are private and should be ignored. Handlers and functions in this AI should also not be modified, unless you expressly wish to modify the AI behaviour. For normal operations, this should not be necessary.

There are 4 non-prefixed AI variables that you should have a look at and modify:

- bEnableMouseForDebugging: turns debug mouse on or off
- nActionButtons: number of buttons (not sticks) you want to have on the touch controller
- nJoypadType: changes the algorithm used for the touchpad sticks
- sTargetAIModel: the AI where the onJoypadStickMove etc. event will be forwarded to.

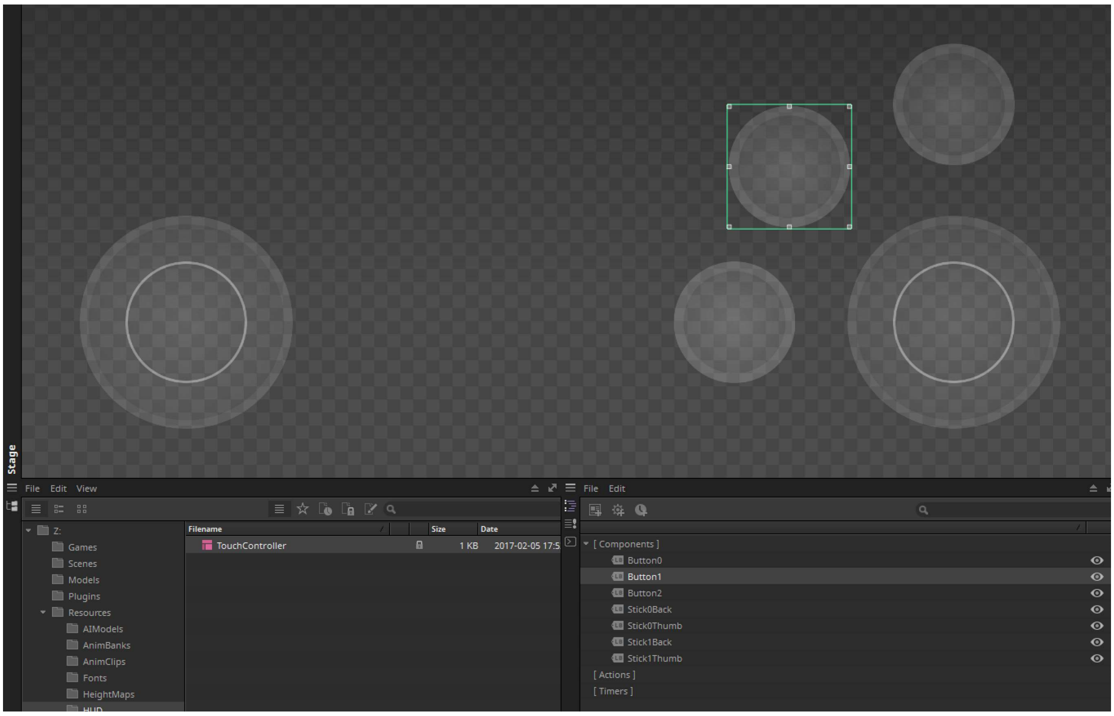


Name: TouchController

```
[ Variables ]
    _bAI
    _bForward
    _bHasActions
    _bMouseDown
    _bTrackingStick0
    _bTrackingStick1
    _tButtonStates
    _tStick0C
    _tStick0L
    _tStick1C
    _tStick1L
    bEnableMouseForDebugging
    nActionButtons
    nJoypadType
    sTargetAIModel
[ Functions ]
    _dxdy ( nX, nY, cX, cY, tcX, tcY, sX, sY, nA )
    isUnderComponent ( nX, nY, nMinX, nMinY, nMaxX, nMaxY )
[ States ]
[ Handlers ]
    onEnable ( bEnable )
    onMouseButtonDown ( nButton, nPointX, nPointY, nRayPntX, nRayPntY, nRayPntZ, nRayDirX, nRayDirY, nRayDirZ )
    onMouseButtonUp ( nButton, nPointX, nPointY, nRayPntX, nRayPntY, nRayPntZ, nRayDirX, nRayDirY, nRayDirZ )
    onMouseMove ( nPointX, nPointY, nDeltaX, nDeltaY, nRayPntX, nRayPntY, nRayPntZ, nRayDirX, nRayDirY, nRayDirZ )
    onTouchSequenceBegin ( )
    onTouchSequenceChange ( nTaps0, nX0, nY0, nTaps1, nX1, nY1, nTaps2, nX2, nY2, nTaps3, nX3, nY3, nTaps4, nX4, nY4 )
    onTouchSequenceEnd ( )
```

HUD

Everything in the HUD is a label, even though some elements behave like buttons. You can move the elements around freely, resize them, and swap their background images, no problem. You can remove one stick if you only need one, but you cannot add a 3rd stick without modifying the code. You can add new buttons if you like, but they must follow the "Button"..i naming scheme. There are no actions in the HUD as they are defined in TouchController.onEnable() for all buttons at once.



Filename	Size	Date
TouchController	1 KB	2017-02-05 17:5

[Components]
- Button0
- Button1
- Button2
- Stick0Back
- Stick0Thumb
- Stick1Back
- Stick1Thumb

[Actions]

[Timers]