

Joystick Configuration

Table Of Contents

- [1 Overview](#)
- [2 Original Urban Assault](#)
- [3 OpenUA \(UA:Source\)](#)
 - [3.1 OpenUA Joystick Solution](#)
 - [3.1.1 Creating the necessary configuration file](#)
 - [3.1.2 Finding the GUID number of the joystick models](#)
 - [3.1.3 Assigning the correct axis and button key mapping values](#)
 - [3.1.4 Quick-Fix Method](#)
- [4 Trivia](#)
- [5 Notes](#)

This article about the joystick configuration settings and options in UA and OpenUA.

(This article is currently under construction!)

1 Overview

Microsoft [Urban Assault](#) offers a comprehensive joystick support for controlling vehicles in the first-person view mode ([Virtual Presence Shift](#)). Some disc copies were also occasionally bundled with Microsoft SideWinder 3D Force Feedback joysticks after the initial release of the game for promotion purposes (without any other contents from the full retail game box package).

This article will explain various configuration options for joysticks in Urban Assault.

2 Original Urban Assault

The original Urban Assault game engine has a built-in function that can automatically detect and assign the joystick key input settings for individual joystick models, and therefore does not normally require any extra configuration setup or settings to use joysticks in the game.

The original Urban Assault can natively and flawlessly support the Force Feedback feature, which is enabled by default inside the game options and can be turned on/off by the user as necessary. UA was also one of the first game titles to demonstrate such features.

Normally, using joysticks on the original UA software will not have any issues, since every joystick configuration parameter is detected and handled automatically by the game engine.

3 OpenUA (UA:Source)

Main Article: [OpenUA \(UA:Source\)](#)

Unlike the original UA, OpenUA uses custom SDL modules to provide support for its joystick functionality. The current design mechanism requires a user to create a custom key mapping configuration file in order to manually assign input options for the individual joystick models.

Otherwise, the control key mappings for joysticks in OpenUA will be frequently misplaced or misconfigured into wrong axes and buttons in most of time, which can be very inconvenient and usually renders them

almost unusable.

In order to rectify this particular joystick issue in OpenUA, check the following solutions/instructions below.

3.1 OpenUA Joystick Solution

Read this section carefully and if you encounter any further problems, feel free to ask in the community forums.

3.1.1 Creating the necessary configuration file

The proper joystick set-up method for OpenUA requires a creation of a custom configuration file in the game's root folder, with the file name `joyconf.txt` (it should be an empty text file). The latest OpenUA Standalone Package contains this file (empty) by default in its root folder.

Once the joystick configuration file is created in the game folder, it must contain the following information: The GUID of the connected joystick model, as well as necessary parameters that are required to correct the incorrect key mappings.

3.1.2 Finding the GUID number of the joystick models

There are multiple ways of finding the GUID of the connected joystick. You can skip this part if you already know your joystick's GUID number.

If OpenUA was launched at least once after a joystick was connected to the machine, it tracks and records the relevant technical information of the last detected joystick model inside `sdInputLog.txt` file that is located at ENV folder. Otherwise, the file will contain 'No joysticks found' message.

Inside the `sdInputLog.txt` file there will be information about the joystick model, GUID, as well as how many inputs are available on the joystick. Copy the GUID number and paste it into the newly created `joyconf.txt` in the game folder, and then add one colon `:` at the end of the GUID number, then save the file.

3.1.3 Assigning the correct axis and button key mapping values

This is the tricky part as it requires knowing which axis and button functionality are currently assigned to certain axes and buttons on the individual joystick models inside OpenUA. It is worsened by the fact that individual joystick models will inherently have different initial key mappings and settings, so it is impossible to describe a general solution here since the case will be different for everyone.

The examples of syntax to correct the key mappings are as follows:

Code

```
a1:3      -      Axis      1      re-assigned      as      Axis      3
a2:1      -      Axis      2      re-assigned      as      Axis      1
b3:4      -      Button     3      re-assigned      as      Button     4
b9:1 - Button 9 re-assigned as Button 1
```

It should be quite straightforward, but figuring out which axes and buttons are currently assigned in OpenUA relative to individual joystick models may require some trial-and-error. It may be helpful if there's a joystick manual or other reference materials available to facilitate the process.

The mapping parameters above should be continually added as necessary followed by the initial GUID number (with spaces in-between). The final format of a line inside `joyconf.txt` should look like this (as an example):

Code

790006000000000000000000504944564944: a1:3 a2:1 b3:4 b9:1

It is also possible to add support for multiple joysticks in this way (add new lines to enter information about multiple GUIDs and key mappings information for individual joystick models).

Code

```
790006000000000000000000504944564944:      a1:3      a2:1      b3:4      b9:1
12345678901234567890123456789012:      a2:4      a3:1      b4:8      b12:6
02147483648429496729685899345920:  a1:2      a3:4      b5:6      b7:8      b9:10      b11:12
...
```

As mentioned above, each joystick model may requires different settings and configuration combinations. So it is impossible to describe a general solution here. Once finished, save the file and test the updated inputs in OpenUA.

3.1.4 Quick-Fix Method

DISCLAIMER: This method is NOT universally verified and applicable! The individual joystick models have different base designs, and so the following method is not guaranteed to work for all joystick models. However, apparently it works for many joystick models by default.

Swapping the function numbers of axis 2 with axis 3 and axis 3 with axis 2 (a2:3 a3:2) appears to resolve the most common key mapping mismatch issue for many joystick models. There is no guarantee that this method will work, but it is still recommended as the first option to try.

4 Trivia

Under Construction.

5 Notes

Under Construction.