Chrome Dino WebHID

Play the Chrome dino game via WebHID by jumping with a Nintendo Joy-Con controller in your pocket

Thomas Steiner
tomac@google.com
Google Germany GmbH
Hamburg, Germany

ABSTRACT

In this demonstration, we show how special hardware like Nintendo Joy-Con controllers can be made accessible from the Web through the new WebHID API proposal. This novel technology allows developers to write "drivers" in pure JavaScript that talk to Human Interface Device (HID) devices via the WebHID protocol. One such example is the project Joy-Con-WebHID, which allows for fun demos of the kind of playing pastimes like the Chrome offline dinosaur game by jumping, which is detected through the accelerometer built into Joy-Con controllers whose signals are read out by the driver and are used to control the game character in the browser. A video demo of the experience is available online.

CCS CONCEPTS

• **Information systems → Web applications**; **Browsers**;

KEYWORDS

Progressive Web Apps, Web APIs, WebHID

ACM Reference Format:
Thomas Steiner. 2021. Chrome Dino WebHID: Play the Chrome dino game via WebHID by jumping with a Nintendo Joy-Con controller in your pocket. In *Companion Proceedings of the Web Conference 2021 (WWW '21 Companion), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 2 pages. https://doi.org/123

1 INTRODUCTION AND BACKGROUND

Universal Serial Bus (USB) is a communications architecture that gives a personal computer (PC) the ability to interconnect a variety of devices using a simple four-wire cable. The USB is actually a two-wire serial communication link that runs at either 1.5 or 12 megabits per second (mbs). USB protocols can configure devices at startup or when they are plugged in at run time. These devices are broken into various device classes. Each device class defines the common behavior and protocols for devices that serve similar functions. One of these classes is the Human Interface Device (HID) class. The HID class consists primarily of devices that are used by humans to control the operation of computer systems. Typical examples of HID class devices include [2]:

(1) Keyboards and pointing devices—for example: standard mouse devices, trackballs, and joysticks.
(2) Front-panel controls—for example: knobs, switches, buttons, and sliders.
(3) Controls that might be found on devices such as telephones, VCR remote controls, games or simulation devices—for example: data gloves, throttles, steering wheels, and rudder pedals.
(4) Devices that may not require human interaction but provide data in a similar format to HID class devices—for example, bar-code readers, thermometers, or voltmeters.

Many typical HID class devices include indicators, specialized displays, audio feedback, and force or tactile feedback. Therefore, the HID class definition includes support for various types of output directed to the end user. For the context of this demonstration, we focus on devices of the category (1).

2 RELATED WORK

The Gamepad specification [1] defines a low-level interface that represents gamepad devices. Currently, the only way for a gamepad to be used as input would be to emulate mouse or keyboard events, however this would lose information and require additional software outside of the user agent to accomplish emulation. The Gamepad API provides a solution to this problem by specifying interfaces that allow web applications to directly act on gamepad data. Interfacing with external devices designed to control games has the potential to become large and intractable if approached in full generality. The authors of the Gamepad specification explicitly chose to narrow the scope to provide a useful subset of functionality that can be widely implemented and that is broadly useful. Specifically, they chose to only support the functionality required to support gamepads. Support for gamepads requires two input types: buttons and axes. Both buttons and axes are reported as analog values, buttons ranging from [0..1], and axes ranging from [-1..1]. They specifically excluded support for more complex devices that may also be used in gaming contexts, including those that that do motion or depth sensing, video analysis, gesture recognition, *etc.*

REFERENCES

[1] Steve Agoston, James Hollyer, and Matt Reynolds. 2020. *Gamepad.* Working Draft 29 October 2020. W3C. https://www.w3.org/TR/gamepad/.
[2] USB Implementers' Forum. 2001. *Device Class Definition for Human Interface Devices (HID), Firmware Specification—5/27/01, Version 1.11.* Technical Report. Universal Serial Bus (USB). https://usb.org/sites/default/files/hid1_11.pdf.

WWW '21 Companion, April 19–23, 2021, Ljubljana, Slovenia

ACM ISBN 123.
https://doi.org/123

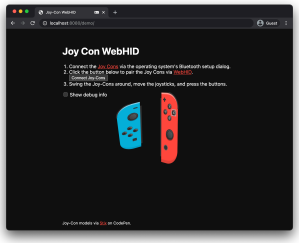

Figure 1: Joy-Con WebHID driver demo

Figure 2: Chrome Dino WebHID demo