בס"ד

http://www.healthgamez.fun/

Web Bluetooth Cycling and Cadence sensor plugin – Lite

Documentation

Version 1.0

The Web Bluetooth Cycling and Cadence sensor plugin is is aimed at enabling and simplifying development of exercise, fitness and wellness games based on stationary bikes or bikes on stationary trainers. The plugin may also be used for development of web/hybrid apps or games for non stationary biking.

The plugin allows connecting a WebGL build to a BLE (Bluetooth Low Energy) Cycling and Cadence sensor sensor supporting the GATT (Generic Attribute Profile) profile CSCP (Cycling Speed and Cadence Profile).

The plugin uses the Web-Bluetooth API and the games/apps built with the plugin will only work on web-browsers that implement this API. Browser compatibility can bee found in this table: https://developer.mozilla.org/en-US/docs/Web/API/Web_Bluetooth_API#browser_compatibility. (Currently Chrome, Microsoft Edge, Opera[1] and Samsung Internet support Web-Bluetooth, while Firfox, Android WebView, Safari and other iPhone webkit based browsers do not).

The computer or device running the game/app must have Bluetooth 4 support.

The plugin has been developed and tested with Magene S3+ Speed/Cadence Dual Mode Sensor which may be oredered from the HelthGamez store.

Examples of games using this plugin are available at www.healthgamez.fun/gamez

1 Disable by default

Security limitations of Web-Bluetooth API

1. HTTPS only

Bluetooth API can be used only from *localhost* or from a https:// site (secure contexts). It can be used on site hosted on github.io

2. User gesture required

As a security feature, discovering Bluetooth devices, in plugin done by invoking method *Connect()*, must be triggered by a user gesture such as a touch or a mouse click. In the example games the *Connect()* is invoked when clicking on a UI-button.

Invoking *Connect()* not as result of user gesture, example from *awake* method, will result in error.

3. No cross-origin frame

In Chromium (Chrome, Ms Explorer, etc.) the API currently only allowed to be used in i-frame if the frame domain is the same as the page domain. Thus it is not possible to embed in a site a frame from a different top-domain, such is commonly done on site such as Unity play, Itch etc. This issue is currently open as bug, and should eventually be solved with using *Feature Policy*, see https://bugs.chromium.org/p/chromium/issues/detail?id=518042

Until then, Web Bluetooth is limited to top-level origin only

Content of plugin: Prefab, Script, Public Methods and Public Fields

2.1 Prefab

A single prefab is included in the plugin, *WebBTCycleSpeedCadencePlugin.prefab*. The prefab is a game-object that contains script *CscSensor.cs*

2.2 Script/Class

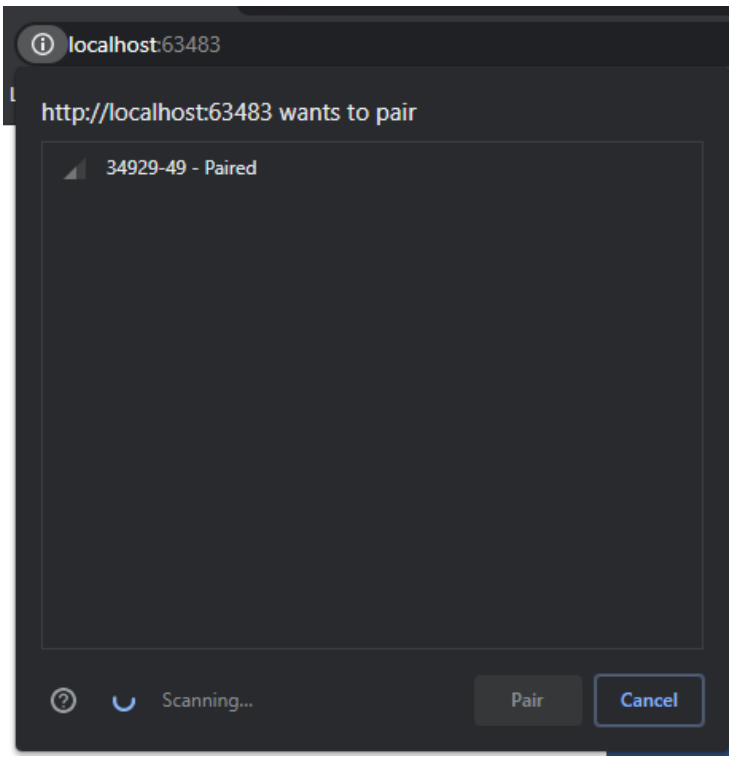
The functionality of the plugin is provided by the *CscSensor.cs* containing the *CscSensor* class.

The class provides the following methods and fields.

Methods:

`public void Connect()` - Will initialize scan for compatible Bluetooth sensors. A popup window will be displayed in web-browser showing all compatible Cycling and Cadence sensors in range.[2]

2 Bluetooth has to be enabled on computer/device. Many Cycling and Cadence sensors enter sleep mode when not active and require some active rotation to allow them to be discovered.



To complete connection, user need to select the desired sensor and click pair. (For security reason, the Web-Bluetooth API require user gesture for every connection, see above).

`public bool IsConnected()` - This method is used to to check if sensor is connected/paired. Typically this will be used to check that a sensor has been connected before allowing user to play game. It can also be used, as in demo, to diplay icon showing connection status.

Public Fields:

```
public byte SensorType { get; private set; } //Sensor type, reading content [1=Wheel
Rotation Sensor, 2=Cadence Sensor, 3=Wheel and Cadence]

public uint CumulativeWheelRevolutions { get; private set; } //Cumulative Wheel Revolution,
read only, do not roll over max 4,294,967,296 revolutions

public ushort LastWheelTimeStamp { get; private set; } //Last Wheel Revolution Event Time-
stamp [seconds*1024], read only, roll over every 64 seconds

public double WheelRotationSpeed { get; private set; } //Wheel rotation speed [rpm], read
only

public ushort CumulativeCrankRevolutions { get; private set; } //Cumulative Crank
Revolution,  read only, roll over every 65,535 revolutions.

public ushort LastCrankTimeStamp { get; private set; } //Last Crank Revolution Event Time-
stamp [seconds*1024], read only, roll over every 64 seconds

public double Cadence { get; private set; } //Cadence (Crank rotation speed) [rpm]

public double wheelReadingTimeout = 3.0; //Maximum time between cumulative readings before
setting speed value to zero [seconds]

public double crankReadingTimeout = 3.0; //Maximum time between cumulative readings before
setting speed value to zero [seconds]
```

Usage

1. Add the *WebBTCycleSpeedCadencePlugin.prefab* to a scene. As *CscSensor* contain *DontDestroyOnLoad* it is sufficient to add to an initial scene at it will continue to following scenes.

2. Add a *CscSensor* field in the script from where you want to use the plugin functionality and link it to *WebBTCycleSpeedCadencePlugin* game-object by either using `GameObject.Find("WebBTCycleSpeedCadencePlugin").GetComponent<CscSensor>()` or manually in inspector.

3. Invoke Connect() through a user gesture such ex clicking on a button.

4. Use above given public fields to read data from sensor.

See also the kart-game tutorial at
https://healthgamez.fun/devzone/webbtcscplugin/kartbikegamedemo/

Demo

A demo-scene demonstrating the usage of the plugin is included. The plugin can tested live at https://healthgamez.fun/devzone/webbtcscplugin/

In addition a kart-game tutorial and example, based on the Unity Kart game tutorial, is available at https://healthgamez.fun/devzone/webbtcscplugin/kartbikegamedemo/

Support

For support visit https://healthgamez.fun/support

Enjoyed?

1. Pleas rate the plugin in asset store.
2. Share your games on healthgamez, https://healthgamez.fun/gamez/publish/
3. Pleas support development work by making donation through https://healthgamez.fun/donate
4. Follow healthgamez on social media.