Unity Game to Browser Extension(UGBE):

This tool exports a fully working Unity' game to a browser extension. Currently export support chromium based browsers like Microsoft Edge, Google Chrome etc.

UGBE works for WebGL builds. This is a two step process;
1- Unity makes WebGL build.
2- UGBE processes built files for extension.
3- Add extension specific files.

Configuration:

Settings are located in the Unity toolbar at this path.
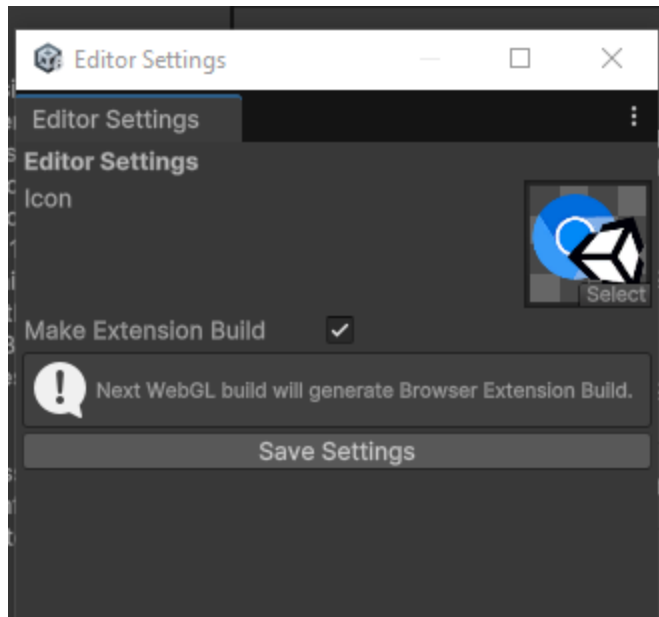***Window/BroswserExt/Editor Settings***

UGBE relies on prebuild and postbuild scripts in Unity. It configures extensions based on information in the WebGL build profile.

You can control or switch between regular WebGL build and Extension Build by toggling "Make Extension Build" from UGBE settings menu.

Manual input is needed for the icon of the extension. This icon will be applied to the browser toolbar or extension menu. This icon is configured in the UGBE Editor Window.

You can edit manifest file for extension which is located at;
"Assets/Editor/ExtensionCreator/manifest_template.json".

Note: All capitalized values like BUILD_VERSION or GAME_NAME should not be changed.

Build Profile/Setting Requirements:

WebGL build on which extension must have;
1- Data caching disabled. You can disable it from;
Build Profiles -> Publishing Setting -> Data Caching

2- Compression Disabled: You can disable file decompression from;
Build Profiles -> Publishing Setting -> Compression Format

Load an unpacked extension(Install local extension):

To load an unpacked extension in developer mode:
- Go to the Extensions page by entering chrome://extensions in a new tab. (By design chrome:// URLs are not linkable.)
- Alternatively, click the Extensions menu puzzle button and select Manage Extensions at the bottom of the menu.
- Or, click the Chrome menu, hover over More Tools, then select Extensions.
- Enable Developer Mode by clicking the toggle switch next to Developer mode.
- Click the Load unpacked button and select the extension directory.

When to reload the extension

During unity game development cycle, you don't need to reload. However in certain cases you need to reload an extension. The following table shows which components need to be reloaded to see changes:

Extension component	Requires extension reload
The manifest	Yes
Service worker	Yes

Content scripts	Yes (plus the host page)
The popup	No
Options page	No
Other extension HTML pages	No