

# iGP11

Application alters rendering pipeline of the games based on loaded plugin. This includes textures modifications (dumping, overriding and specifying level of details) and custom effects application based on color and depth information. Full support/compatibility in chosen games is ensured by the already implemented profiles. Not yet supported games might be handled by generic plugin, though some things might not work properly.

## Requirements:

- Windows 7 64bit or newer

## Installation

- Application can be installed with *setup.exe*.
- In case of failure, try to install prerequisites manually and then run *iGP11.Installer.msi*:
  - Microsoft .NET Framework 4.6.1 (by default included in Windows 10):  
<https://www.microsoft.com/en-us/download/details.aspx?id=49981>
  - Visual C++ Redistributable: <https://www.microsoft.com/en-us/download/details.aspx?id=48145>
  - Run *iGP11.Installer.msi* and follow setup steps
    - **DO NOT** install the program into game directory, directly into C:\ or C:\Users (this includes Desktop etc.)

## Configuration

- Launch *iGP11.Tool* with *iGP11* shortcut
- Choose game (default: *Dark Souls 3*)
  - Every game can have many game profiles, but only the one chosen in *game profile* section is the default one
- Choose executable file path for picked game
  - **DO NOT** specify location to game launcher (.exe file, which launches borderless window with possibility of updating settings and actually launching the game).
  - Specify location to real .exe file, which should be in the same directory as game launcher (while game is running this .exe file can be found in *Windows Task Manager*)
- On game launch *iGP11* folder is created in game directory (path is listed in *game* section)
- Additional folders might be created depending on chosen plugin (currently only *Direct3D11* is available). Those folders can be accessed in specific plugin configuration.
  - *Direct3D11* plugin: copy textures to *tex\_override* folder (path is listed in *Textures* section), make sure *override mode* is chosen

## Usage

- *iGP11* shortcut always opens the *iGP11.Tool* configuration window. Game with *iGP11* can be loaded by:
  - Clicking *Launch game* button (game launched with this option inherits permissions granted by operation system to *iGP11.Tool*)
  - Minimize (not necessary) *iGP11.Tool* and launch the game through its shortcut. Assuming configuration is correct (executive file path is correct etc.), the game will launch with *iGP11* (*iGP11.Tool* will detect game launching and will configure it for you)
- *iGP11.Launcher* shortcut launches current game with configuration specified in game profile (it is shortcut for opening *iGP11* and clicking *Launch game* button)
- Communication can be established with game with already loaded *iGP11* by clicking *Connect* button (by default port 7321 is used). With this, changes can be made in real time.

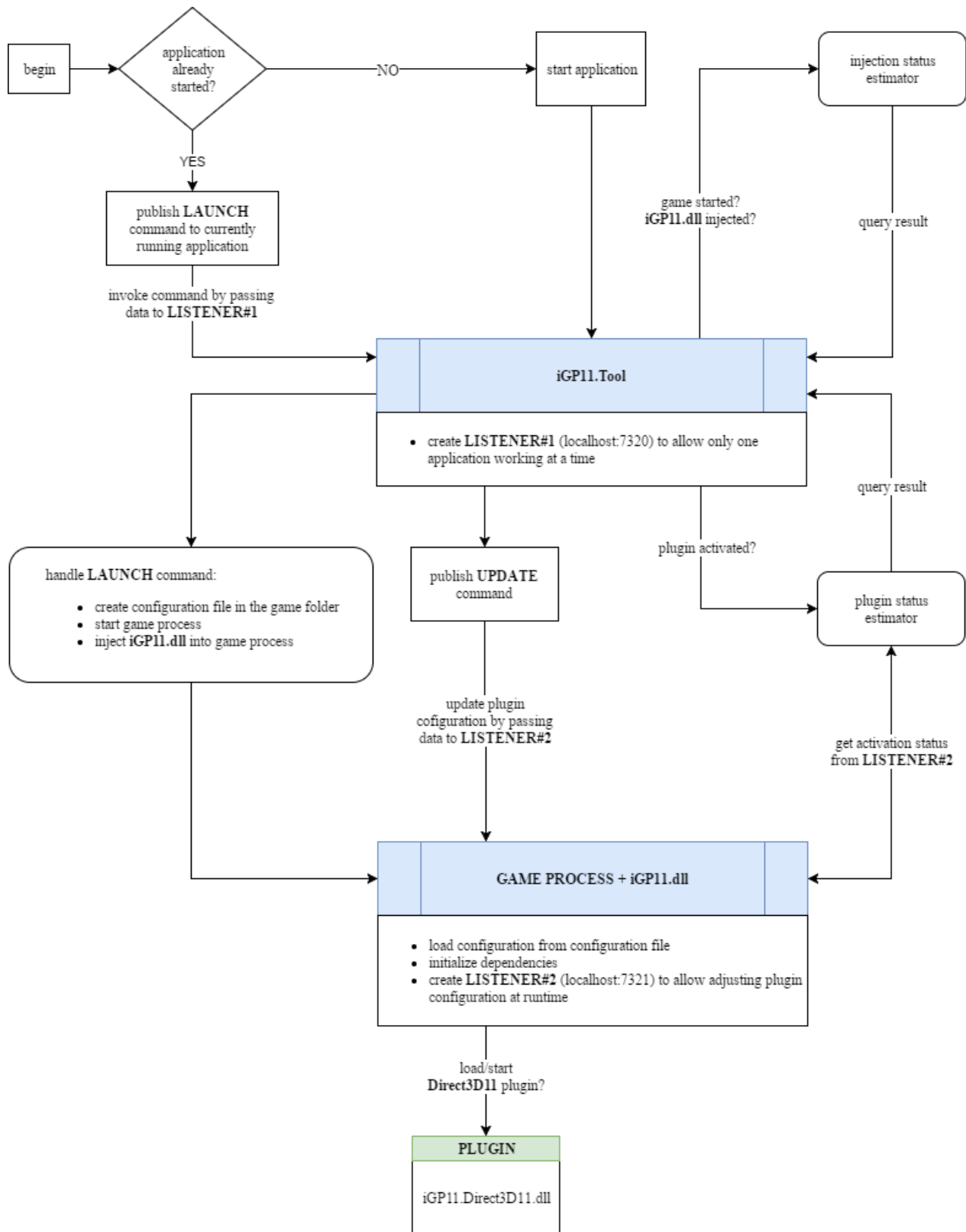
## Risks

- On game launch *iGP11* folder is created in game directory with:
  - *logs*, *tex\_override* and *tex\_dump* folders
  - *settings.igp* file

Opening *iGP11.Tool* with Administrator privileges does not ensure that those folders/files will be created, since game is located in different directory (in normal case, they will be created). However, if you have installed extremely restrictive antivirus (they work on OS level), they might prevent those elements from being created – the same goes for *iGP11* initialization as well. If game crashes on launch, that's the first thing to check.

Just in case: everything works just fine with UAC enabled, Windows Firewall, Avast and AVG on Windows 10.

## Application Flow



## Texture modding

### 1. Dumping textures

- Select *Dumping mode* in configurator and launch your game by clicking *Launch* button or double clicking *iGP11.Launcher.exe*
- Start new game/load save and play – textures used by the game will be dumped to location specified in *dumping directory* in configuration.

### 2. Converting textures

Most dds editors won't be able to open and edit formats introduced in *DirectX10/DirectX11* (like *BC7*). In order to get around this issue click *Texture management* menu item.

This window allows you to convert textures to other formats (the best ones are those supported by *DirectX9*). Conversion must be done with caution. Especially, take closer look at textures saved in *Typeless* format – convert them to *UNORM* and adjust their color space (sRGB).

It is advised to convert textures to *BC3\_UNORM*, because uncompressed textures take a lot of space.

Under the hood, the application uses *DirectXTk* to load custom textures – it is important for you to know, that this library doesn't support *DirectX10/DirectX11* formats.

Documentation:

[https://msdn.microsoft.com/en-us/library/windows/desktop/bb173059\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/bb173059(v=vs.85).aspx)

[https://msdn.microsoft.com/en-us/library/windows/desktop/hh308955\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/hh308955(v=vs.85).aspx) (look at minimum supported feature level)

### 3. What textures can be applied?

You can alter dumped textures in any way you want (change size, format etc.). Just remember to save textures in the format supported by *DirectX9*.

### 4. How to force *Srgb* color space for selected texture?

Add to *tex\_override* folder a .txt file named: *tex\_id.txt* with content set to: *force\_srgb=1* (example in *tex\_cache* folder, works only for textures, which were dumped as *Typeless*)

## Supported games

Currently only *Direct3D11* rendering library is supported. Supported games have effects applied before the game renders the UI:

- Dark Souls 2 Scholar of the first sin
- Dark Souls 3
- Fallout 4

## Known issues

Some applications acting as overlays might cause compatibility issues. One of those applications seems to be *FRAPS*, which for some reason prevents *IDXGISwapChain::Present* to be hooked and called. This can be recognized by having *plugin partially initialized, awaiting final initialization...* status not changing after quite some time. Solution seems to be quite simple: launch *FRAPS* after the game.

## Libraries used:

- Autofac: <https://autofac.org/>
- Automapper: <http://automapper.org/>
- Log4net: <http://logging.apache.org/log4net/>
- DirectXTex: <https://github.com/Microsoft/DirectXTex>
- DirectXTk: <https://github.com/Microsoft/DirectXTK>
- MinHook: <https://github.com/TsudaKageyu/minhook>