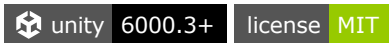# Unity Fast Play

`unity 6000.3+`  `license MIT`

**Fast Play** adds a ⚡ button to the main toolbar in the Unity Editor (Unity 6000.3+). It allows you to enter Play Mode almost instantly by temporarily disabling Domain and Scene reloading.

It acts as a safe, convenient wrapper around Unity's built-in **Enter Play Mode Options**. It doesn't perform any magic; it simply makes these powerful settings accessible when you need speed, and automatically restores your safe defaults when you're done.

> ⚠️ **Warning:** Disabling Domain Reload means **static variables are not reset** between play sessions. Ensure your code handles static variable initialization correctly (see Handling Static Variables).

## Table of Contents

## Features

- **Instant Play Mode:** Skips Domain and Scene reload for rapid iteration.
- **Toolbar Integration:** Adds a toggle button directly to the main toolbar (next to Play/Pause).
- **Keyboard Shortcut:** Press `Shift+Alt+P` to toggle Fast Play.
- **Safety First:** Automatically restores your original Play Mode settings when you stop playing.
- **Visual Feedback:** The ⚡ icon changes color to indicate that Fast Play is active.
- **Smart Overlay:** Automatically detects if the button is hidden from the toolbar and offers to enable it.

## Getting Started

1. **Install:** Import this package into your Unity project.
2. **Enable:** Upon first load, a dialog should ask if you want to show the Fast Play button. Click "Yes".
3. **Use:**
   - Click the ⚡ button in the toolbar.
   - Or use the shortcut **Shift+Alt+P**.
   - Or go to **Edit > Play Mode > Fast Play**.

> **Tip:** If the button is missing, you can **right-click** anywhere on the main toolbar and select **Fast Play**, or go to **Edit > Play Mode > Show Fast Play Button**.

## Compatibility

- **Requires Unity 6000.3 or newer.**
- Uses the new `UnityEditor.Toolbars` API.

## Handling Static Variables

When Domain Reload is disabled (which Fast Play does), **static variables are not reset** between play sessions. This is standard Unity behavior for "Fast Play" modes.

You must manually reset your static variables to ensure your game logic works correctly when restarting. The best way to do this is using the `[RuntimeInitializeOnLoadMethod]` attribute with `SubsystemRegistration`.

**Example:**

```csharp
public class MyScoreManager : MonoBehaviour
{
    public static int Score = 0;

    // This method runs before the scene loads, resetting the static variable.
    [RuntimeInitializeOnLoadMethod(RuntimeInitializeLoadType.SubsystemRegistration)]
    static void ResetStatics()
    {
        Score = 0;
    }
}
```

## Known Issues & Limitations

- **Static Variables:** As mentioned above, static fields persist. Ensure you reset them.
- **Scene Persistence (Scene Reload):** Since Scene Reload is disabled, objects in the scene are not handled the same way. This may lead to unexpected behavior in some cases.
- **Third-Party Assets:** Some assets or plugins rely on a full domain or scene reload to initialize correctly.
- **Recommendation:** Use Fast Play for rapid iteration on gameplay logic, but frequently test with the regular Play button to ensure your game works correctly with a fresh start.
- **Reporting:** Issues can be reported on GitHub: https://github.com/JonathanTremblay/UnityFastPlay/issues

## Contact

**Jonathan Tremblay**
Teacher, Cegep de Saint-Jerome
jtrembla@cstj.qc.ca

Project Repository: https://github.com/JonathanTremblay/UnityFastPlay

## Version History

- 1.0.0
  - First public version for Unity 6000.3+. Added toolbar button, shortcuts, and safety checks.

## License

- This project is licensed under the MIT License - see the LICENSE file for details.