Rainity

Setting up a desktop application in Unity is easy with Rainity. All you have to do is place a Rainity component onto any object in the scene, set your runtime version and API compatibility level to .NET 4.x in Edit > Project Settings > Player > Other Settings, set the build target to x86_64 bit Windows, and you're good to go.

Settings

In the Rainity component, you can customize how your window behaves:

- `Vector2` Window Offset

  Changes the default position of the window. This is mainly used to prevent the window from obscuring the task bar.
  `float x` - X offset of the window
  `float y` - Y offset of the window

- `bool` Hide From Taskbar

  Hides the application's icon from the task bar. This takes no effect if Never Hide Window is true.

- `bool` Never Hide Window

  Sets the application window's owner to the desktop handle, preventing it from hiding when "show desktop" or Win+D is activated. This will always hide the application from the task bar.

- `bool` Keep Bottom Most

  Keeps the window on the bottom of the visible window z-order and right above the desktop.

- `bool` Borderless

    Removes window borders and the close/maximize/minimize buttons.

- `bool` Behind Icons

    Places the window behind desktop icons, however key input and mouse button input is disabled with this on. Certain mouse position and Unity functions like OnMouseEnter() still work.

- `bool` Use Rainity Input

    In applications that run behind desktop icons, normal built-in mouse and keyboard input won't work, so a custom Input system has to be used instead. This setting enables the use of this system.

API

Rainity's API is the most important part about making desktop applications:

- Rainity.GetFileIcon `public static Texture2D GetFileIcon(string filePath)`

    Gets a 256x256 texture of the specified file's icon.
    `string filePath`   - The file's absolute path.
    returns `Texture2D`   - The file's icon as a texture.

- Rainity.GetMemoryInformation `public static MemoryInformation GetMemoryInformation()`

    Gets information about the RAM and VRAM.
    returns `float MemoryInformation.ramTotal`   - The total number of bytes of RAM.
    returns `float MemoryInformation.ramUsed`   - The number of bytes of RAM used.
    returns `float MemoryInformation.vRamTotal`   - The total number of bytes of VRAM.
    returns `float MemoryInformation.vRamUsed`   - The number of bytes of VRAM used.

- Rainity.GetCPUUsagePercent `public static float GetCPUUsagePercent()`

    Gets an approximate reading of how much CPU is being used. This value updates once every second.
    returns `float`   - Percent of CPU being used.

- Rainity.GetDiskInformation `public static DiskInformation GetDiskInformation(string driveName)`

  Gets information about a specified disk. Drive names must include the colon and backslash (e.g. C:\).
     `string driveName` - The drive's letter followed by a colon and a backslash.
     returns `string DiskInformation.driveName` - The drive's letter followed by a colon and a backslash.
     returns `long DiskInformation.bytesTotal` - The total number of bytes on the drive.
     returns `long DiskInformation.bytesFree` - The number of bytes available on the drive.

- Rainity.SimulateKey `public static void SimulateKey(uint keyCode)`

  Simulates any key press including media buttons. Use the DesktopApplication.VirtualKeys enum for codes.
     `uint keyCode` - The key to be simulated. Use DesktopApplication.VirtualKeys or a number from this table.

- Rainity.GetAverageColorOfTexture `public static Color32 GetAverageColorOfTexture(Texture2D tex)`

  Gets the average color of a texture. Does not average completely transparent pixels. This could be used for icon borders or dynamic backgrounds based on icons.
     `Texture2D tex` - The texture to be averaged.
     returns `Color32` - The average color of tex.

- Rainity.GetWallpaperImage `public static Texture2D GetWallpaperImage()`

  Gets the user's wallpaper as a Texture2D.
     returns `Texture2D` - The wallpaper texture.

- Rainity.AddToStartup `public static void AddToStartup()`

  Adds the program to the Windows startup folder (C:\Users\ [USER]\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup).

- Rainity.RemoveFromStartup `public static void RemoveFromStartup()`

  Removes the program from the Windows startup folder (C:\Users\

[USER]\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup).

- Rainity.GetUserName `public static void GetUserName([optional] bool includeDomain)`

    Gets the current logged-in user's Windows username.
    `bool includeDomain` - If true and account is on a network, include the domain name.
    returns `string` - The user's username (and/or domain).

- Rainity.GetUserAvatar `public static Texture2D GetUserAvatar()`

    Gets the user's avatar image as a Texture2D.
    returns `Texture2D` - The user's avatar image.

- Rainity.CreateSystemTrayIcon `public static SystemTray CreateSystemTrayIcon()`

    Adds an icon to the system tray located on the taskbar. This can include a context menu used for interaction with the wallpaper.
    returns `SystemTray` - The system tray class associated with the newly created icon.
        `SystemTray.SetTitle(string title)`
        Sets the system tray title to a string (shows when you hover over it).
        `string title` - The title to set.

        `SystemTray.AddItem(string label, Action function)`
        Adds an item to the system tray's context menu and attaches a function to it.
        `string label` - The context menu item label.
        `Action function` - The function to call (this can be any void/function).

        `SystemTray.AddSeparator()`
        Adds a separator to the context menu.

        `SystemTray.Dispose()`
        Removes the system tray icon.

- Rainity.OpenFile `public static void OpenFile(string path)`

    Opens the specified file or directory. If 'path' is a directory, Windows explorer will open to the specified directory, otherwise open the file in the file extension's default program.
    `string path` - The file to be opened.

- Rainity.GetWeatherInformation `public static WeatherObject GetWeatherInformation()`

    Gets information about the weather using your computer's location including temperature, wind speed, atmosphere, sunset and sundown times, and a 7-day forecast all using Yahoo Weather.
    returns `WeatherObject` - A Yahoo Weather JSON decoded object containing all the weather information.

- Rainity.GetFiles `public static RainityFile[] GetFiles(string directory)`

    Gets information about all the files in the specified directory.
    `string directory` - The directory to search.
    returns `RainityFile[]` - An array of RainityFile objects that contain the file's name, path, extension, and whether it is a directory.

Rainity Input API

In applications that run behind desktop icons, normal built-in mouse and keyboard input won't work, so a custom Input system has to be used instead. For UI elements, replace the StandaloneInputModule script in the EventSystem to a RainityStandaloneInputModule script or use the RainityEventSystem prefab provided in the package.

- RainityInput.GetMouseButton `public static bool GetMouseButton(int button)`

    Returns true if mouse button is pressed (left click = 0)
    `int button` - The button index to test (currently, only left click (0) is implemented).
    returns `bool` - Whether or not the mouse button is pressed.

- RainityInput.GetMouseButtonDown `public static bool GetMouseButtonDown(int button)`

    Returns true once mouse button is down (left click = 0)
    `int button` - The button index to test (currently, only left click (0) is implemented).
    returns `bool` - Whether or not the mouse button is down.

- RainityInput.GetMouseButtonUp `public static bool GetMouseButtonUp(int button)`

    Returns true once mouse button is up (left click = 0)
    `int button` - The button index to test (currently, only left click (0) is implemented).

returns `bool` - Whether or not the mouse button is up.

- RainityInput.GetKey `public static bool GetKey(VirtualKeys keycode)`

  Returns true if the specified key is pressed.
  `VirtualKeys keycode` - The key value to test.
  returns `bool` - Whether or not the key is currently pressed.

- RainityInput.GetKeyDown `public static bool GetKeyDown(VirtualKeys keycode)`

  Returns true once the specified key is down.
  `VirtualKeys keycode` - The key value to test.
  returns `bool` - Whether or not the key is currently pressed.

- RainityInput.GetKeyUp `public static bool GetKeyUp(VirtualKeys keycode)`

  Returns true once the specified key is up.
  `VirtualKeys keycode` - The key value to test.
  returns `bool` - Whether or not the key is released.

- RainityInput.GetButton `public static bool GetButton(string buttonName)`

  Returns true if the specified button defined by the RainityInput script is pressed. To add buttons, place the RainityInput script on any object and modify the Input Entries array to your liking.
  `string buttonName` - The button to test.
  returns `bool` - Whether or not the key is pressed.

- RainityInput.GetButtonDown `public static bool GetButtonDown(string buttonName)`

  Returns true once the specified button defined by the RainityInput script is down. To add buttons, place the RainityInput script on any object and modify the Input Entries array to your liking.
  `string buttonName` - The button to test.
  returns `bool` - Whether or not the key is down.

- RainityInput.GetButtonUp `public static bool GetButtonUp(string buttonName)`

Returns true once the specified button defined by the RainityInput script is up. To add buttons, place the RainityInput script on any object and modify the Input Entries array to your liking.

`string buttonName`   - The button to test.
returns `bool`   - Whether or not the key is up.