

# STATE OPTIMIZER

BY HISHAM KHUNAYN

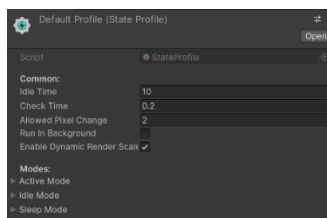
## CONTENTS

|                          |   |
|--------------------------|---|
| Description .....        | 1 |
| How it works .....       | 1 |
| List of Properties ..... | 2 |
| State Optimizer .....    | 2 |
| State Profile .....      | 2 |
| S Mode .....             | 2 |
| How to use .....         | 3 |
| Code Example .....       | 4 |

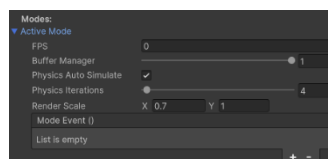
## DESCRIPTION

A simple unity asset to use less CPU/GPU power by dynamically limiting some unity features when it is not needed.

## HOW IT WORKS



The idea is simple, The State Profile has 3 modes: Active, Idle, and Sleep. Each mode has 6 settings: FPS, buffer Manager, physics Auto Simulate, physics Iterations, Render Scale, and Mode Event.



Active mode: called when the player is active (detected by keyboard/mouse input)

Idle mode: called when the player is not giving any input for 10 sec (can be changed)

Sleep mode: called when the player is outside/minimized the game/application!

## LIST OF PROPERTIES

### STATE OPTIMIZER

| PROPERTY NAME           | TYPE         | FUNCTION   |
|-------------------------|--------------|--|
| _disableInEditor        | bool         | Will not be working in Editor if enabled         |
| _profile                | StateProfile | The profile used                                 |
| SetMode(StateMode mode) | void method  | Sets the mode for the StateOptimizer.            |
| GetMode                 | StateMode    | Returns the current mode of the StateOptimizer.  |
| ModeEvent               | UnityEvent   | Returns the UnityEvent for a specific StateMode. |

### STATE PROFILE

| PROPERTY NAME            | TYPE  | FUNCTION   |
|--------------------------|-------|--|
| IdleTime                 | float | Time of no active to be idle in seconds          |
| CheckTime                | float | Time interval between activity checks in seconds |
| AllowedPixelChange       | float | Minimum pixel movement for mouse activity        |
| RunInBackground          | bool  | Whether application runs in the background       |
| EnableDynamicRenderScale | bool  | Enables dynamic render scale (URP only)          |
| ActiveMode               | SMode | Settings for active mode                         |
| IdleMode                 | SMode | Settings for idle mode                           |
| SleepMode                | SMode | Settings for sleep mode                          |

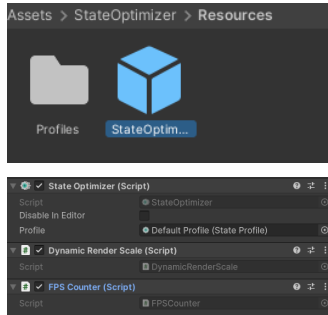
### S MODE

| PROPERTY NAME       | TYPE       | FUNCTION   |
|---------------------|------------|--|
| BufferManager       | float      | Scale factor for the BufferManager                 |
| PhysicsAutoSimulate | bool       | Enables physics simulation when the mode is active |
| PhysicsIterations   | int        | Number of physics iterations (affects accuracy)    |
| RenderScale         | Vector2    | Minimum and maximum scale factor for rendering     |
| ModeEvent           | UnityEvent | Event triggered when the mode is enabled           |

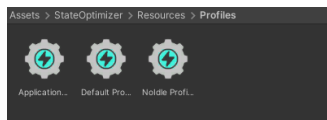
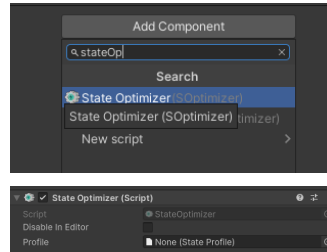
## HOW TO USE

There are 2 ways:

1- Add the “StateOptimizer” prefab **OR**



2- Add “StateOptimizer” script to any gameObject



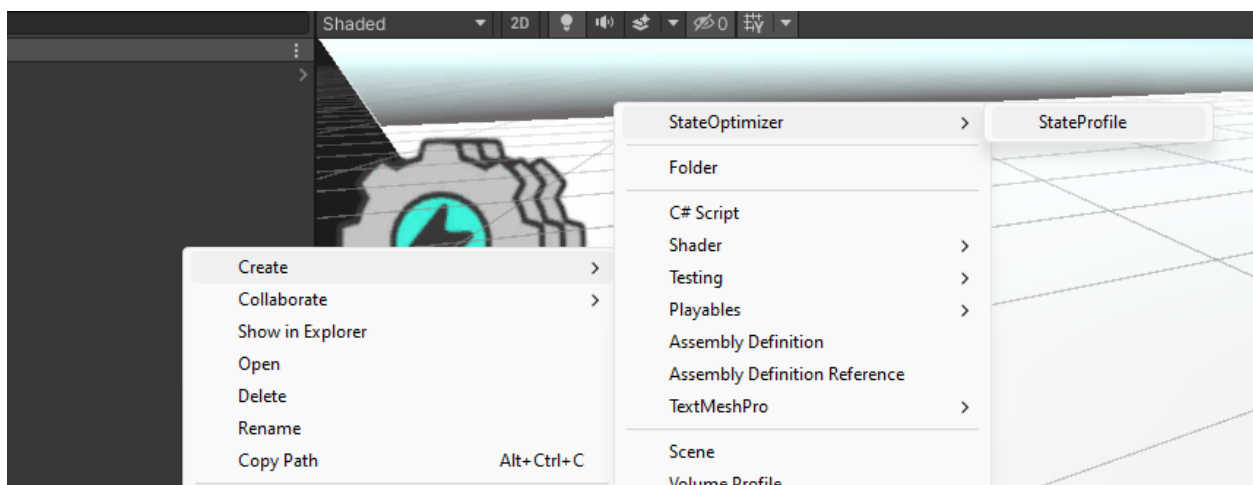
There are 3 predefined profiles: Application Profile, Default Profile, and No Idle Profile

Default Profile: Recommended for most games.

Application Profile: Recommended for Application like games.

No Idle Profile: with no idle state.

To create your own:



Right click at project window then *Create -> StateOptimizer -> StateProfile*

## CODE EXAMPLE

While coding isn't required, this serves as an illustrative example.

```
1  using UnityEngine;
2  using SOptimizer;
3
4
5  Unity Script (1 asset reference) | 0 references
6  public class StateOptimizerExample : MonoBehaviour
7  {
8      // Start is called before the first frame update
9      Unity Message | 0 references
10     void Start()
11     {
12         // Get The current mode
13         Debug.Log("current mode is " + StateOptimizer.GetMode());
14
15         // Set The mode for the optimizer
16         StateOptimizer.SetMode(StateMode.Active);
17
18         // call a function when mode state change
19         StateOptimizer.ModeEvent(StateMode.Idle).AddListener(IdleFunction);
20     }
21
22     1 reference
23     void IdleFunction()
24     {
25         Debug.Log("its idle now");
26     }
27 }
```