

# Synchronization

Synchronization is a Unity3D package which allow to execute synchronized tasks.

## Features

- Execute synchronized tasks
- Execute asynchronous tasks
- Define ordered tasks queue with synchronization

## Usage

Define your task by extending `Executable` class:

```
public class MyTask : Executable {  
  
}
```

### Asynchronized Tasks

For asynchronous task, implements `Execute` parameters less method :

```
public override void Execute() {  
    // Do async task  
}
```

Next task in the queue will be executed right after this method execution.

### Synchronized Tasks

For synchronized task, implements `Execute` with parameters method :

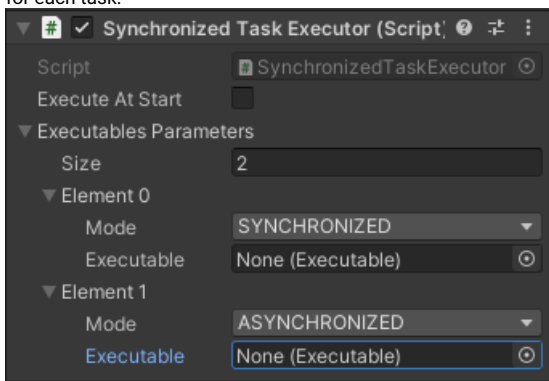
```
public override void Execute(TaskCompletionSource<bool> completion)  
{  
    // Do sync task  
}
```

When your synchronized task is completed, set completion result to true. It will trigger the next task.

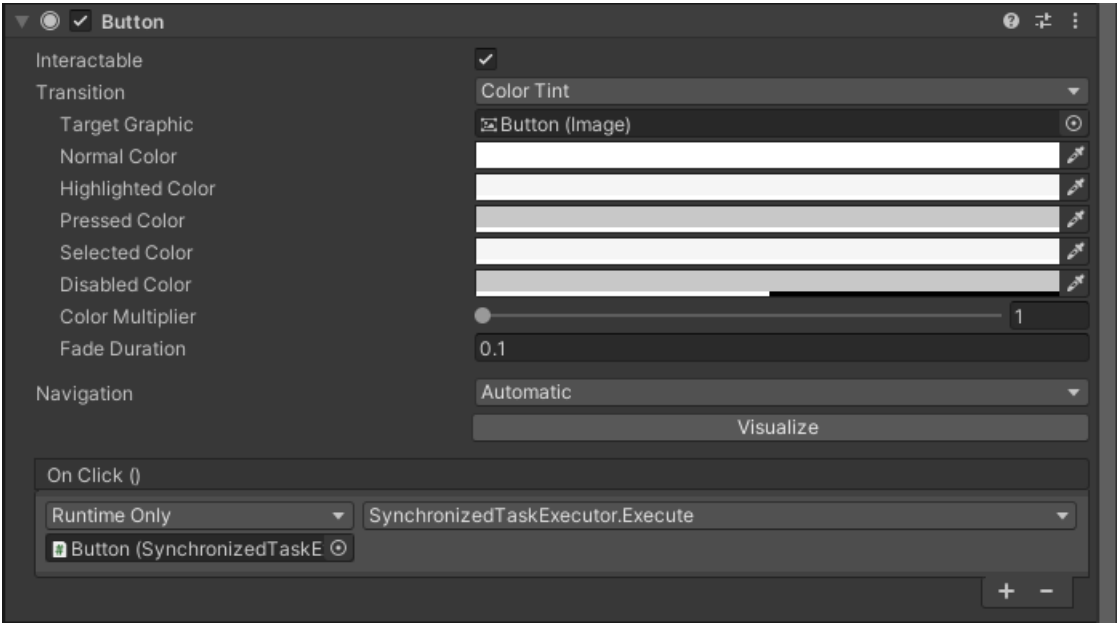
```
completion.SetResult(true);
```

### Execute tasks

Add [SynchronizedTaskExecutor](#) class to a gameobject, and define your queue by adding task to `ExecutableParameters` variable. You can choose execution mode for each task:



Then run the `Execute` method of [SynchronizedTaskExecutor](#) from a script or from Unity Inspector.  
For example with a Button:



## Releases

Clone or download this repository.

## License

MIT