**Game Scripting Foundation Pack**

Thank you for purchasing this pack. If you have any questions, or suggestions for improvements, please email [robert.wells@gandhigames.co.uk](mailto:robert.wells@gandhigames.co.uk).

**Demo Scene Overview**

* Shows the event, audio, and object pool system in use.
* Controls: Left-click to duplicate shape and half its size, right-click on shape to cause an explosion at that point.

When you right-click on a shape a CubeSelectedEvent or SphereSelectedEvent is raised depending on the shape selected. The shapes are subscribed to the relevant events and respond.

Left-click on a shape to raise an ExplosionEvent and an AudioEvent2D. The ExplosionEvent contains information on the source, radius, and force of the explosion. The AudioEvent2D contains the explosion audio. The audioplayer in the scene is subscribed to all AudioEvents raised and plays the audioclip contained in the event.

The shapes are added and removed to the scene using an object pool.

Further details for each system are below.

**Event System**

Decouples code. Any of your scripts can raise an event – you can keep your systems (i.e. audio, physics, animation etc) separate.

How to setup your own events:

Three step process:

1. Create the event class
2. Register any interested classes
3. Raise the event at the appropriate time

Creating the event class.

* Extend GameEvent
* Should contain any required information and methods (although empty events are fine).

Registering Interested Classes.

It should become common practice to register the event in the OnEnable method and to deregister in OnDisable.

Raising an event.

An event can be raised by any class.

**Audio System: 2D and 3D**

Works in conjunction with the event system.

How to implement the audio system.

**Object Pool**

What is it used for

* Improve memory usage performance by reusing objects (enabling/disabling) instead of instantiating and destroying. Objects are obtained from a pool.
* Especially important for mobile development where memory is scarce.
* Can cause memory fragmentation

Why is it used

* Useful for when you have a large number of objects on screen. Examples include:
  + Tile systems, where the environment consists of a large number of tiles.
  + Custom particle effect systems.
* Object pool improves performance and prevents memory fragmentation by allocating most of the required objects at runtime. The objects and can then be enabled and disabled when required.

How is it used

* It is not fixed pool, you have the option to retrieve objects even if the pool

How to use the Object Pool.