Round Spawn System

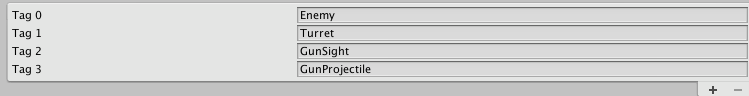
1.0

If you have any questions, or suggestions for improvements, please email robert.wells@gandhigames.co.uk.

# Demo Scene Setup

For the demo scene to work correctly please add the following tags and layers. These are only required for the demo scene; you do not need to setup any custom tags or layers when using the Round Spawn System.

Tags: Enemy, Turret, GunSight, GunProjectile



Layers: Enemy



# Overview

Each round has a user defined number of checkpoints, within each checkpoint you can define: how long the checkpoint runs for, what enemies are spawned, enemy spawn chance, time between enemy spawns, and the maximum number of enemies that can be spawned.

By separating a round into a number of checkpoints you can easily set the intensity of not only the game, but each individual round.

# Quick Start

1. Add Round Manager class to scene
2. Setup round manager: setup enemy prefabs, round time, bosses, preparation time etc.
3. Implement the request events (see Events section below for more information).
4. Call RoundManager.Instance.Begin () to start the round manager.

# Querying Round Manger

For full code documentation please see the separate code documentation. However, a number of queries that can be run against the Round Manager are shown below.

*Please note: most of the interaction with the round manager is through reacting to events rather than direct querying.*

RoundManager.Instance: returns an instance of the RoundManager class. Enables access to the RoundManager class from any script.

RoundManager.Instance.CurrentRound: returns the currently executing round.

RoundManager.Instance.CurrentRound.CurrentCheckpoint: returns the currently executing checkpoint.

RoundManager.Instance.CurrentRound.InPreperation: returns true if the current round is in the preparation stage.

RoundManager.Instance.CurrentRound.RoundTime: returns the current remaining round time as a float.

RoundManager.Instance.CurrentRound.RoundTimeInSeconds: returns the remaining round time in seconds.

RoundManager.Instance.CurrentRound.RoundTimeInMinutes: returns the remaining round time in minutes.

RoundManager.Instance.CurrentRound.RoundTimeInMinutesSeconds: returns the remaining round time in minutes and seconds.

RoundManager.Instance.CurrentRound.RoundTotalTime: returns the total round time (including preparation time) as a float.

RoundManager.Instance.CurrentRound.RoundOver: returns true if round is over.

# Events

Events are raised during each round; they allow you to easily extend the round managers functionality. It is up to you how (or if) you react to these events.

*Please note: Events can contain useful information. For example, the event ‘EnemySpawnRequestEvent’ contains the enemy prefab that should be spawned.*

A number of examples of how to react to events are included in:

Scripts/Demo Scripts/Event Examples/

A template for each event is included in:

Scripts/\_Round System/Event System/\_Templates

These templates provide a quick starting point.

**Important**: there are a number of important events that should be reacted to:

* EnemySpawnRequestEvent
* ObjectSpawnRequestEvent
* BossSpawnRequestEvent

It is important to include code to react to these request events otherwise enemies/objects will not be spawned, and if you enable ‘destroy enemies on round start/end’ in the round manager you need to respond to the event ‘DestroyCurrentEnemiesRequestEvent’. Examples of all three are included in the Event Examples folder.

## Rough Order Of Events

FirstRoundEvent: raised at the beginning of the first round. Only called once per game.

LastRoundEvent: raised the beginning of the last round. Only called once per game.

RoundStartEvent: raised at the beginning of each round.

PreperationStartEvent: raised at the beginning of a rounds preparation stage. If there is no preparation stage, then the event is not raised.

PreperationEndEvent: raised at the end of a rounds preparation stage. If there is no preparation stage, then the event is not raised.

CheckpointStartEvent: raised at the beginning of each of a rounds checkpoint.

CheckpointEndEvent: raised at the end of each of a rounds checkpoint.

BossCountDownBegunEvent: raised at the beginning of a boss countdown, if: a round has a boss, and the boss has a countdown period.

BossKilledEvent: raised when a boss is killed.

RoundEndEvent: raised at the end of each round.

FinishedRoundEvents: raised when all rounds have finished.

Request Events:

**It is important to respond to these events to enable the full functionality of the round manager**.

DestroyCurrentnemiesRequestEvent: raised when all currently spawned enemies should be destroyed. For example, this is raised when ‘DestroyAllEnemiesOnRoundEnd’ has been enabled in the round manager and a round ends.

EnemySpawnRequestEvent: raised when an enemy should be spawned. The event contains the enemy prefab.

ObjectSpawnRequestEvent: raised when a preparation object should be spawned. The event contains the object prefab.

BossSpawnRequestEvent: raised when a boss should be spawned. The event contains the boss prefab.

## Other Important Classes

RoundBoss: attach to any round boss. This script raises the BossSpawnedEvent and informs the current round when the boss has been killed.

RoundEnemy: attach to any round enemy. Informs the current round when an enemy has been spawned and killed.