Welcome to the Shooter Game Kit.

~~########~~

ITWEEN

First thing you will want to do is import the ITween package from the unity store package. Next you'll want to edit the following files ZombieEnemy, ChunkManager, LightChanger and FlyToPortal and change the name from *//#define ITWEEN, to #define ITWEEN then save all.*

OVERVIEW

So first thing's first. First load up the main scene Scenes/scene0 (found in the ActionKit/Scenes folder).

So lets look at the gameObjects in alphabetical order.

World

The world object has a script called Backgroundscene which will ensure that the objects will not be destroyed when changing scenes normally). If you reload back to this level it will destroy these objects otherwise you have 2 instances of it. Because we are either are going to load additively where none of the scene objects get destroyed -- except all objects under the GameObject "Room" -- using our own code, or unloading the scenes normally.

World/CAMERA

CameraShakescript which will simply shake the camera.
Constraintscript which will simply stop the camera from moving along certain axises, we have it set to be constrained along the xaxis.
GrayScaleEffect this is our simple grayscale post filter, I've actually modified the shader so that if the red component is set to 1 and the green is set to 0 it will actually use the original color giving a sort of "sin city type effect". SimpleLerpCamerascript will simply follow the player gameObject, it will lock along the zaxis.
ConstraintEhance script will simply make the scene "pop abit more". If your doing mobile you'd want to disable this script.
BloomOptimized script will make the scene look abit more vibrant

World/PACKAGEMANAGER

Packagemanagerscript which will simply create a random Package from an array of packagesafter xnumber of enemies have been killed. The package will fall from the sky and land, using an offset from the player's position

The package gameObjects contain a damagable script (which controls how much health the object has), a rotator script which will rotate the crate, a Package script. Once a package is destroyed, it will create a health (which has a healthPowerup script), a mana shake (which has a ManaPowerup script), and will activate a secondary weapon (which has a weapon activator script).

World/Light

Light Changer which will change the color of the light depending, and turn on or off

the rain system depending on which event is fired.

Rain System

Simply has a rain sound that plays and a child rain particle system

Room

The room gameObject will be scene dependant and will be destroyed when the next scene is additively loaded. Its important to keep this one called "Room" so it will simply work with the current system. All enemies, projectiles will be spawned as a child of the "room" object so they will be destroyed when you load the next scene.

Canvas This is simply a canvas group in world space which will display the current starting pipes.

There are 4 starting teleporters.

The teleporter has a green and render material. When all enemies have been killed it display green. Then when the player walks into the teleporter he advances to the

next level, if there is a ChunkLoader script will skip ahead to a certain level offset (which will be handled by the chunkmanager script), if the isFinalChunk is checked it will skip to the final levle (which should be the boss level).

World/Game

The Game gameObject has alot of scripts, so let's go through them 1 by 1.

The first is the gamescript script which will handle when the player dies, enemies die, enemies are sucked into the portal. Or the player picks up a powerup.

The Music script will handle changing between regular battles, and boss battles. The UIHealth, UIMana, and UIAmmo will display how much of each thing the player has.

The floating text manager will be used for displaying floating texts, spawning a floating text gameObject. The floating text gameObject has a floating text script which will simply float the text upwards.

The ChunkManager script will load the next level chunk, unload the current "Room" GameObject. Moving the camera and player back to the start

The PlayerLevel script which can contains an array of ints, which determine when the player levels up (and how many levels he has).

Game has some children including
Canvas (which has the the game UI). It has 2 Panels called GameOverPaneland PausePanelwhich get activated when its either the player wins (gets to the last level), or the player dies and the pause panel when the games paused, playing the slide in reverse animation. The GameOverPanelhas a gamescript the gameoverAnimations (which has an array of objects set by the user) which will sequentially enabled the child gameObjects after 1 second delay (which will then play there animations).
Game/Menu GameObject
Has two scripts GameOverMenuwhich sets the time and kills, and the gameMenu panel which handles pause, rertry, main menu.

Hallway

This contains our street scene, along with a backplane and attack spawner gameObject. The attackSpawner gameObject will disable the spawner when all enemies have been killed, and reenable it when you load a new level.

World/Player

IsometricPlayer controls the movement of the player
Damagable which contains the amount of health, mana etc
SlowDownTime will stop the enemies, slow down the music pitch.
Constraint the player is constrained to the axises (x,z anx y).
PlayerAnimation controls the animation of the player gameObject. UpgradeManager controls the list of possible perks in an arary
OneLiners controls the list of grunts, and deaths sounds gord will make
PlayerWeaponManager holds an array of weapons, which the order will correspond to the weapon that gets activated by the weaponactivator script (which is created when you destroy a “package”).

World/Player/Magnet

Contains how far away powerups have to be before they move towards the player, and how fast these magnets are.

World/Player/Weapons

Contains a number of children prefabs which contain the various weapons. Loading a scene using the same geometry.
Okay now lets open up Scenes/Level1/floor1.

You'll notice that we only have a room gameObject. Our enemies, a teleporter, start location. It doesnt have a camera, gamescript, or anything else, because we simply kept everything we had from the previous scene. Everything from the previous scene will be loaded into this scene except for the gameObject “Room” which will be destroyed.

That means we can have 1 game from the scene for the whole game (and never need to rebake lighting or navigation mesh. Great for keeping your file sizes down.

Okay now lets open up Scenes/Level1/floor1.

You'll notice that we only have a room gameObject. Our enemies, a teleporter, start location. It doesnt have a camera, gamescript, or anything else, because we simply kept everything we had from the previous scene. Everything from the previous scene will be loaded into this scene except for the gameObject “Room” which will be destroyed.

That means we can have 1 game from the scene for the whole game (and never need to rebake lighting or navigation mesh. Great for keeping your file sizes down.

Okay that’s great but what if I want to have a different behavior (for example the boss room).

In that case let’s take a look at the boss room scene.

Open the file ActionKit/Scenes/BossRoom/ActionKit
Now instead of loading the level additively, we load it normally (ie unloading all the other stuff, including the navmesh).

So we have some new scene geometry, a new navmesh. We have a second UI for the boss, and a boss gameObject.

We have a gameObject called - **ChangeLevelScene** which changes the constraints of the axis, changes our music, and changes the light color.

See the document ReplacingBackground for replacing background art/scenes.