Additive Scene Manager v1.5

NOTICE

1.4 has renamed functions from *Load**Level*** to *Load**Scene***.

- The **Level** word has been replaced with the word **Scene**.
- The previous *UnloadLevels* function has been renamed *Unload**AllScenes***.
- A new function *UnloadScenes* has been added which directly unloads the scenes.
- More new functions, including Reload functions. See here: SceneController.cs

Introduction

Thanks for the purchase and support! We are a community of game & VR devs, working together to create games, experiences, development tools, and tutorials in an effort to empower emerging developers worldwide. Join us here: https://www.youtube.com/nurfacegames/

Video Tutorials

https://www.youtube.com/watch?v=y_D0O2Ove_0

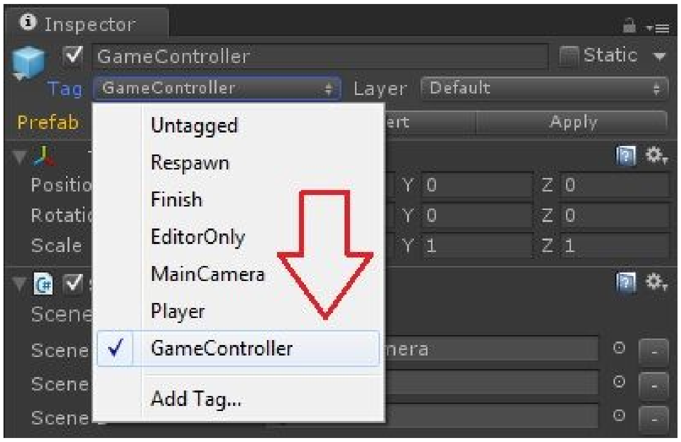
What is Additive Scene Manager?

Dealing with additive scenes and asynchronous scene loading can be difficult and confusing. This asset provides a set of functions that simplifies the asynchronous loading and unloading of additive scenes. The SceneController script has a custom inspector to display which scenes are loading/loaded so it's easy to understand what is happening and what loading stage each scene is at.

How To Use

The player/camera is the object that needs to move between the additive scenes. The main scene with **Main Camera** should have a Game Controller. More info can be found here: https://unity3d.com/learn/tutorials/topics/scripting/persistence-saving-and-loading-data

If you do not have a game controller script, add a new empty GameObject, name it appropriately, and add Unity's default tag *GameController*:

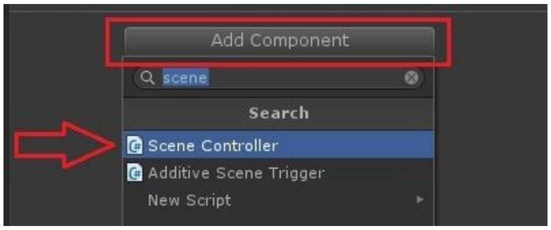

☑ GameController

Tag GameController

Layer Default

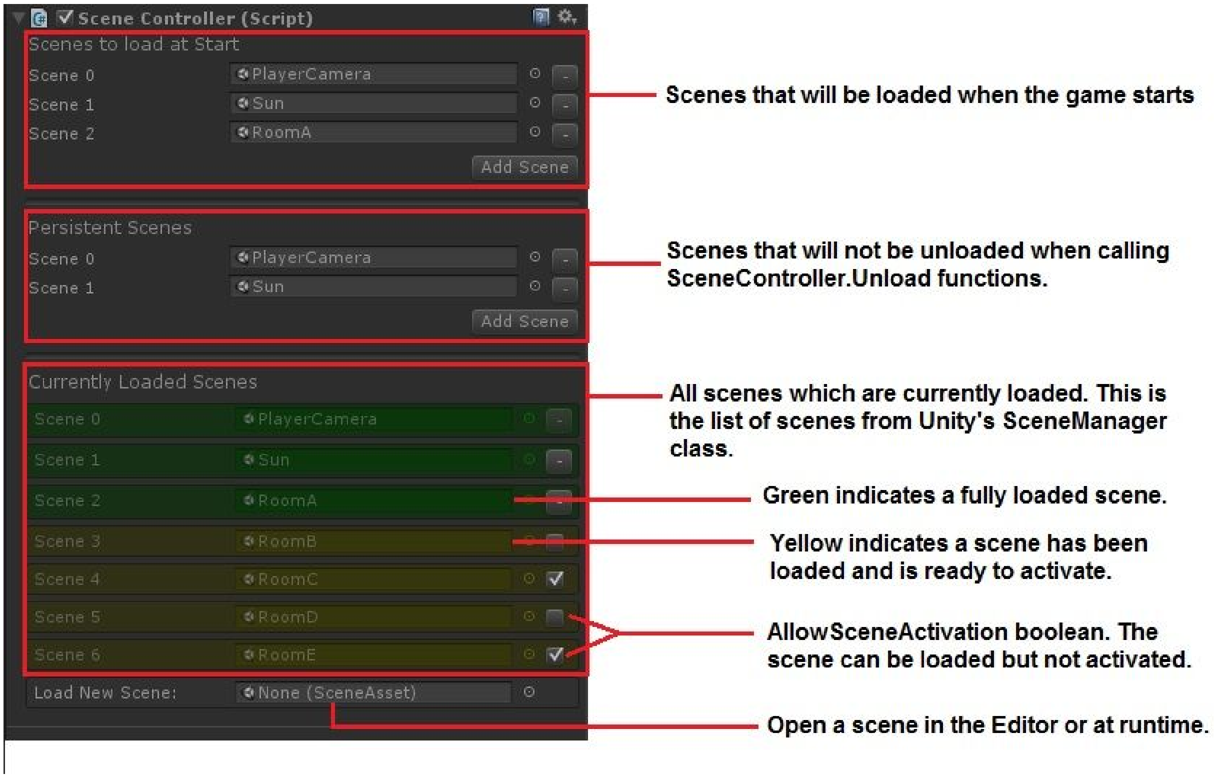
SceneController.cs

The SceneContoller script contains functions for loading and unloading scenes. Add this script to your GameController:



The SceneController class gives the ability to:

- Add scenes that will be loaded when the game starts.
- Add 'persistent' scenes that will not be unloaded by SceneController's Unload functions.
- View all scenes which are currently loaded by Unity's SceneManager class (https://docs.unity3d.com/ScriptReference/SceneManagement.SceneManager.html)
- View what loading stage each scene is at:
  Green = Fully Loaded. Async progress 1.0 (Async operation is completed/null)
  Yellow = Loaded but not activated. Async progress 0.9 (allowSceneActivation if false)
  Red = Loading in progress. Async progress 0.0 - 0.8. (Async operation is loading data)
- Set the *allowSceneActivation* boolean via a checkbox for scenes that are loading.
- Load new scenes into the editor or at runtime.



☑ Scene Controller (Script)

Scenes to load at Start

Scene 0 PlayerCamera
Scene 1 Sun
Scene 2 RoomA

Persistent Scenes

Scene 0 PlayerCamera
Scene 1 Sun

Currently Loaded Scenes

Scene 0 PlayerCamera
Scene 1 Sun
Scene 2 RoomA
Scene 3 RoomB □
Scene 4 RoomC ☑
Scene 5 RoomD □
Scene 6 RoomE ☑

Load New Scene: None (SceneAsset)

Scenes that will be loaded when the game starts

Scenes that will not be unloaded when calling SceneController.Unload functions.

All scenes which are currently loaded. This is the list of scenes from Unity's SceneManager class.

Green indicates a fully loaded scene.

Yellow indicates a scene has been loaded and is ready to activate.

AllowSceneActivation boolean. The scene can be loaded but not activated.

Open a scene in the Editor or at runtime.

SceneController.cs Functions

public bool SceneExists(string name**)**
Check if a scene exists in the Unity project.

public void LoadScene(string sceneName**)**
Load a single scene additively and asynchronously. allowSceneActivation will be set to true.

public void LoadScene(string sceneName, bool allowSceneActivation**)**
Load a scene additively and asynchronously, with control of allowSceneActivation boolean.

public void LoadScenes(string[] sceneNames**)**
Load an array of scenes, they will be loaded in the order of the array. allowSceneActivation is true for all scenes.

public void LoadScenes(string[] sceneNames, bool allowSceneActivation**)**
Load an array of scenes, they will be loaded in the order of the array. The allowSceneActivation boolean will be assigned to all scenes.

public void UnloadScene(string sceneName**)**
Unload a scene. The scene will NOT be unloaded if it's a Persistent Scene (set via inspector).

public void UnloadScenes(string[] sceneNames**)**
Unload an array of scenes. Persistent Scenes will not be unloaded.

public void UnloadAllScenes()
Unload all scenes. Persistent Scenes will not be unloaded.

public void UnloadAllScenes(string exception**)**
Unload all scenes except '*exception*'. Persistent Scenes will not be unloaded.

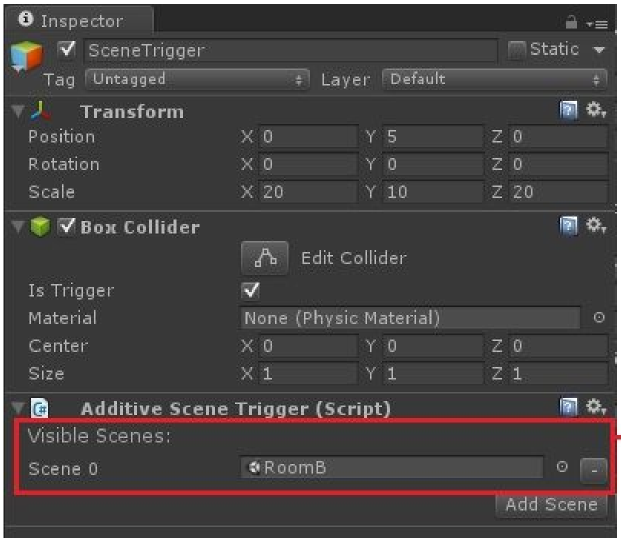
public void UnloadAllScenes(string[] exceptions**)**
Unload all scenes. An array of exceptions will not be unloaded. Persistent Scenes will not be unloaded.

public void ReloadScene(string sceneName**)**
Reload a scene. Persistent Scenes will not be reloaded.

public void ReloadScenes(string[] sceneNames**)**
Reload an array of scenes. Persistent Scenes will not be reloaded.

AdditiveSceneTrigger.cs

This script utilizes the SceneController functions so you can get additive scenes working in your project without using any code. AdditiveSceneTrigger requires a Unity trigger, so add a collider and set *Is Trigger:*



☑ SceneTrigger □ Static
Tag Untagged Layer Default

Transform

Position	X 0	Y 5	Z 0
Rotation	X 0	Y 0	Z 0
Scale	X 20	Y 10	Z 20

☑ **Box Collider**

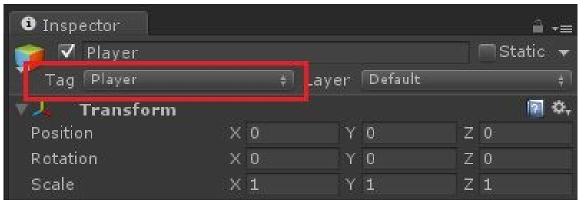
Is Trigger	☑		
Material	None (Physic Material)		
Center	X 0	Y 0	Z 0
Size	X 1	Y 1	Z 1

Additive Scene Trigger (Script)

Visible Scenes:
Scene 0 RoomB

Scenes that are 'visible' from here and will be loaded when player enters this trigger.

- **Visible Scenes** is a list of scenes that are visible from this trigger. Any scene that is set here will be additively loaded when the player enters this trigger. All other scenes will be unloaded, except *Persistent Scenes*.
- Tag "**Player**" must be assigned to the Player/Main Camera object:



☑ Player □ Static
Tag Player Layer Default

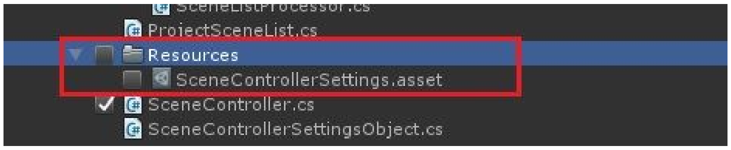
Transform

Position	X 0	Y 0	Z 0
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

(see AdditiveSceneTrigger.cs line 15: *if (!other.gameObject.CompareTag("Player")) return;*)

Upgrading Additive Scene Manager

When upgrading Additive Scene Manager, do not overwrite your settings file. The settings file is located at */Nurface/AdditiveSceneManager/Resources/SceneControllerSettings.asset.* Deselect this folder/file when upgrading Additive Scene Manger, or the settings on the SceneController class will be overwritten.

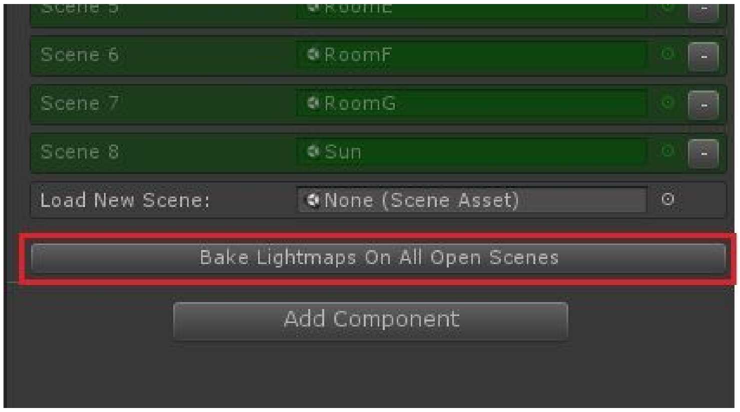


SceneListProcessor.cs
ProjectSceneList.cs
□ Resources
□ SceneControllerSettings.asset
☑ SceneController.cs
SceneControllerSettingsObject.cs

Lightmapping Multiple Scenes

It is possible to bake lighting on multiple opened scenes at the same time. This uses the Unity Scripting API function, *Lightmapping.BakeMultipleScenes:* https://docs.unity3d.com/ScriptReference/Lightmapping.BakeMultipleScenes.html

Open all of the scenes that you want to bake within the Hierarchy Window and then click the “Bake Lightmaps On All Open Scenes” button on the Scene Controller Inspector:



Scene 5 RoomE
Scene 6 RoomF
Scene 7 RoomG
Scene 8 Sun
Load New Scene: None (Scene Asset)

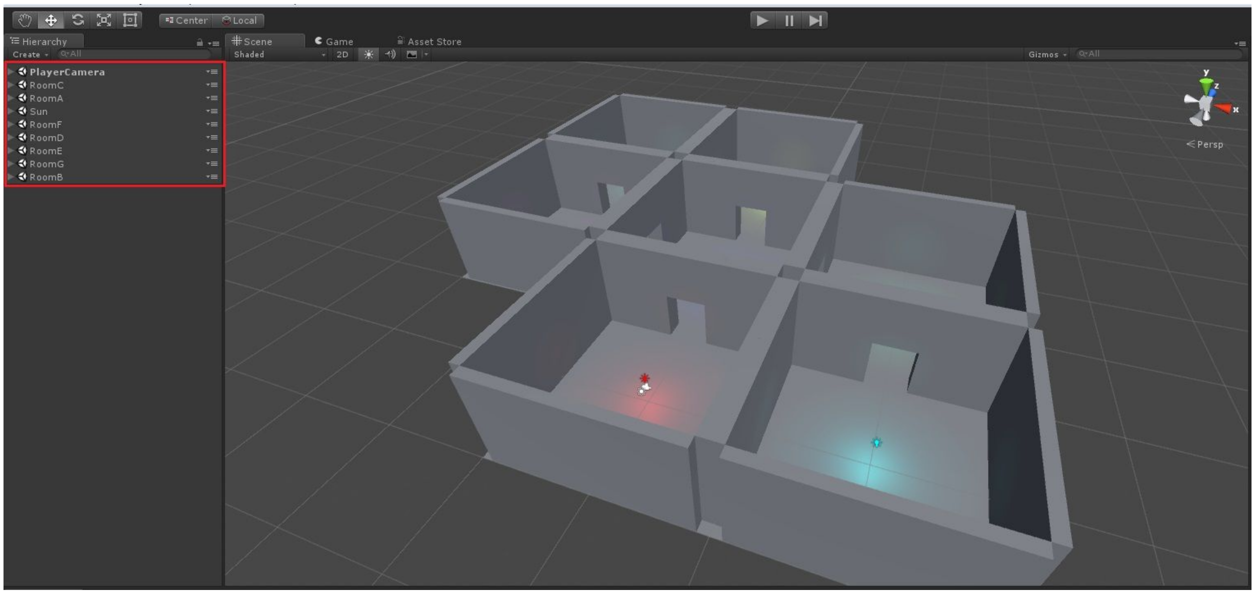
Unity will be unresponsive until the bake has finished on all scenes. This operation is currently not possible to run additively like a normal lightmap bake. I’ve opened a feature request for this, so that Unity Editor can be used while a multi-scene bake is running: https://feedback.unity3d.com/suggestions/bakemultiplescenes-asynchronously-bakemultiplescenesasync-string-paths

The forum post can be found here: https://forum.unity3d.com/threads/bakemultiplescenes-asynchronously.436790/

For now, until Unity adds a *BakeMultipleScenesAsync()* function to the Lightmapping API, be prepared to wait until the bake completely finished before you can use Unity Editor again.

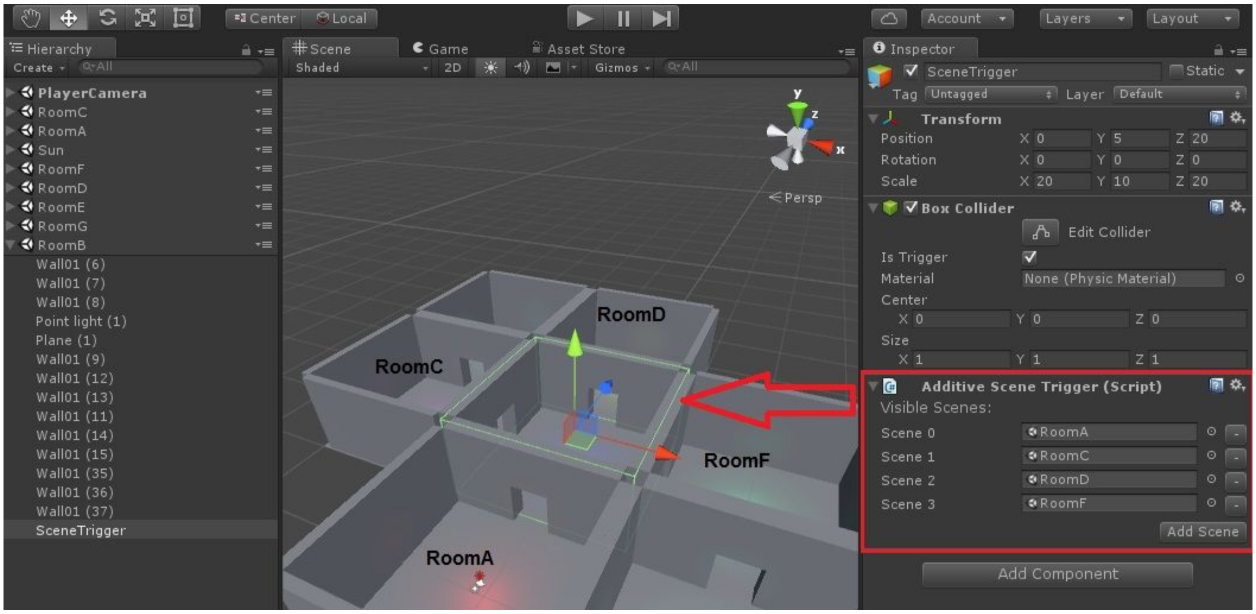
Demo Project

The included project located at */Nurface/AdditiveSceneManager/Demo/* has 9 scenes for how an example game may be set up. To see the complete project, load all 9 scenes in Unity:



PlayerCamera
RoomC
RoomA
Sun
RoomF
RoomD
RoomE
RoomG
RoomB

The idea is that the **PlayerCamera** scene and the **Sun** scene are to be loaded at all times, and each room will be loaded additively. Each room has an **AdditiveSceneTrigger** script with the visible scenes configured. In this example, we have only set the “Visible Scenes” for each room to be the room directly next to it. The following image shows how the middle room, RoomB, is configured:



PlayerCamera
RoomC
RoomA
Sun
RoomF
RoomD
RoomE
RoomG
RoomB
Wall01 (6)
Wall01 (7)
Wall01 (8)
Point light (1)
Plane (1)
Wall01 (9)
Wall01 (12)
Wall01 (13)
Wall01 (11)
Wall01 (14)
Wall01 (15)
Wall01 (35)
Wall01 (36)
Wall01 (37)
SceneTrigger

☑ SceneTrigger □ Static
Tag Untagged Layer Default

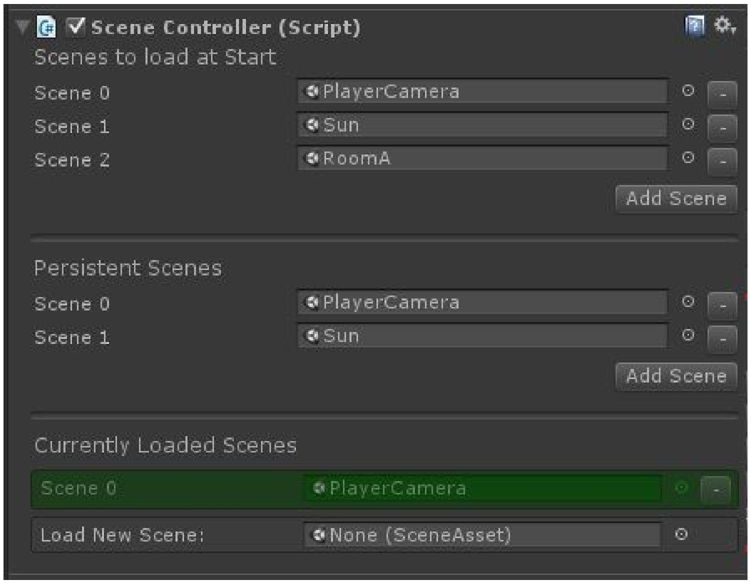
Transform
Position X 0 Y 5 Z 20
Rotation X 0 Y 0 Z 0
Scale X 20 Y 10 Z 20

☑ Box Collider
Is Trigger ☑
Material None (Physic Material)
Center
X 0 Y 0 Z 0
Size
X 1 Y 1 Z 1

Additive Scene Trigger (Script)
Visible Scenes:
Scene 0 RoomA
Scene 1 RoomC
Scene 2 RoomD
Scene 3 RoomF
Add Scene

Open the **PlayerCamera** scene and select the **GameController** gameobject.

- **Scenes To Load At Start** should be set to: PlayerCamera, Sun, RoomA
- **Persistent Scenes** should be set to: PlayerCamera, Sun



Scene Controller (Script)

Scenes to load at Start

Scene 0 PlayerCamera
Scene 1 Sun
Scene 2 RoomA

Persistent Scenes

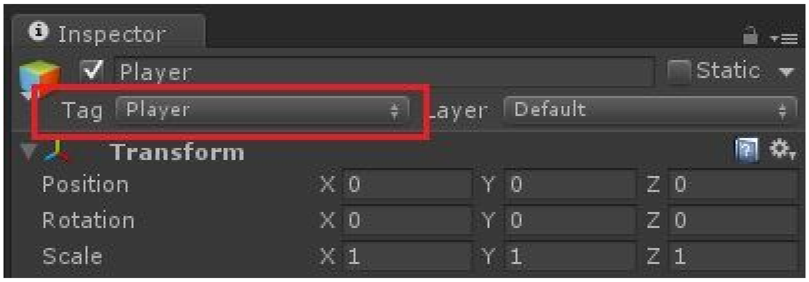
Scene 0 PlayerCamera
Scene 1 Sun

Currently Loaded Scenes

Scene 0 PlayerCamera

Load New Scene: None (SceneAsset)

- **Important!** Ensure the Player gameobject has the **Player** tag set:



☑ Player ☐ Static

Tag Player Layer Default

Transform

Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

You are now ready to play the game, and each room will be loaded additively and asynchronously and any scene that is not persistent nor visible will be automatically unloaded.

- In the demo, hold “Alt” to rotate the camera and click to start or stop movement.

Additional Support

For a video tutorial related to this asset, please click here:
Current Video v1.4:
https://www.youtube.com/watch?v=y_D0O2Ove_0

Older Versions:
https://www.youtube.com/watch?v=dbDAmuTH5sw
https://www.youtube.com/watch?v=IhCPBILR5mg

Join our VR community here for VR tutorials and videos:
https://www.youtube.com/nurfacegames/

For any questions or support, please email:
nurfacegames@gmail.com