1. Record Nr. UNINA9910437972203321

Autore	Thorn Alan
Titolo	Learn Unity for 2D game development / / Alan Thorn
Pubbl/distr/stampa	New York : , : Apress, , 2013
ISBN	1-4302-6230-3
Edizione	[1st ed. 2013.]
Descrizione fisica	1 online resource (xxv, 287 pages) : illustrations (some color)
Collana	Technology in action
Disciplina	794.8028551
Soggetti	Computer games - Programming
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Includes index.
Nota di contenuto	Contents at a Glance; Chapter 1: Unity Basics for 2D Games; Unity Projects, Assets, and Scenes; Project Wizard and Project Panel; Assets and Project Files; Scenes; Navigating Scenes and Viewports; GameObjects, Transforms, and Components; Cameras; Meshes and Geometry; Scripting and the Unity API; MonoDevelop; Components; Performance, Profiling, and the Stats Panel; The Profiler; The Stats Panel; Editor Add-Ons; Unity Interface Configuration; Summary; Chapter 2: Materials and Textures; Using Materials and Textures; Getting Started with Materials; Mesh Renderers; Shaders; Working with Textures Materials for 2D GamesMethod 1: Use White Ambient Light; Method 2: Use Light-Immune Shaders; Creating Textures; Rule #1: Power-2 Dimensions; Rule #2: Retain Quality; Rule #3: Expand Alpha Channels for Transparency; Creating Alpha Textures in Adobe Photoshop; Step 1. Separate Foreground From Background; Step 2. Duplicate and Defringe; Step 3. Expand Edges; Step 4. Merging and Alpha Channels; Creating Alpha Textures in GIMP; Importing Textures into Unity; Importing an Alpha Texture into Unity; Step 1. Import Alpha Texture; Step 2. Create Alpha Compliant Material; Step 3. Create a Textured Quad SummaryChapter 3: Quick 2D Workflow; Getting Started at Making "2D Alien Invasion"; Adding the Player and Enemies to the Scene; Implementing Player Movement; Implementing Player Weapons with Prefabs; Creating an Ammo Prefab Object; Implementing the Ammo Trajectory; Creating the Prefab Ammo Object; Defining the Cannon Point; Coding the Firing of Ammo; Implementing Moving Enemies and

Collision; The EnemyController.cs Script; Setting the BoxCollider as a Trigger Volume; Adding a RigidBody Component; Adding a Level Background; Moving Forward and Project Limitations; Summary
Chapter 4: Customizing the Editor with Editor ClassesEditor Classes; Getting Started with Batch Rename; BatchRename.cs; Creating a Folder for Editor Extensions; Adding Batch Rename to the Application Menu; The CreateWizard Function; Testing the Batch Rename Menu Option; Reading Object Selections in the Scene; Making Use of Selection in BatchRename.cs; Testing Object Selections in Scene; Adding User Input to the Batch Rename Window; Completing the Batch Rename Feature; Summary; Chapter 5: Procedural Geometry and Textured Quads; Getting Started with the CreateQuad Feature
Setting the Quad's Anchor PointSpecifying the Asset Path; Generating the Quad Mesh; Step 1-Create Vertices; Step 2-Create Quad as an Asset; Step 3-Instantiate Quad in Scene; Testing the Quad Mesh Generator; Summary; Chapter 6: Generating Atlas Textures; Getting Started with Atlas Textures; Configuring Texture Inputs; Atlas Textures and UVs; Generating an Atlas - Step 1: Optimizing Texture Inputs; Generating an Atlas - Step 2: Atlas Generation; Generating an Atlas - Step 3: Saving the Atlas Prefab; Testing the Atlas Texture; Summary; Chapter 7: UVs and Animation; Creating a Dockable Editor
Starting an Editor GUI - Selecting an Atlas

Sommario/riassunto

2D games are everywhere, from mobile devices and websites to game consoles and PCs. Timeless and popular, 2D games represent a substantial segment of the games market. In Learn Unity for 2D Game Development, targeted at both game development newcomers and established developers, experienced game developer Alan Thorn shows you how to use the powerful Unity engine to create fun and imaginative 2D games. Written in clear and accessible language, Learn Unity for 2D Game Development will show you how to set up a step-by-step 2D workflow in Unity, how to build and import textures, how to configure and work with cameras, how to establish pixel-perfect ratios, and all of this so you can put that infrastructure to work in a real, playable game. Then the final chapters show you how to put what you've already made to work in creating a card-matching game, plus you'll learn how to optimize your game for mobile devices.