

1. Record Nr.	UNINA9910955458003321
Autore	Creighton Ryan Henson
Titolo	Unity 4.x game development by example beginner's guide // Ryan Henson Creighton
Pubbl/distr/stampa	Birmingham : , : Packt Publishing, , 2013
ISBN	9781849695275 184969527X
Edizione	[3rd ed.]
Descrizione fisica	1 online resource (572 p.)
Collana	Learn by doing: less theory more results
Disciplina	794.81526
Soggetti	Video games - Design Video games - Programming
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Bibliographic Level Mode of Issuance: Monograph
Nota di bibliografia	Includes bibliographical references and index.
Nota di contenuto	Intro -- Unity 4.x Game Development by Example Beginner's Guide -- Table of Contents -- Unity 4.x Game Development by Example Beginner's Guide -- Credits -- About the Author -- About the Reviewers -- www.PacktPub.com -- Support files, eBooks, discount offers and more -- Why Subscribe? -- Free Access for Packt account holders -- Preface -- A word about the third edition -- Your future as a game developer -- What this book covers -- What you need for this book -- Who this book is for -- Conventions -- Reader feedback -- Customer support -- Downloading the color images of this book -- Downloading the example code -- Errata -- Piracy -- Questions -- 1. That's One Fancy Hammer! -- Introducing Unity 3D -- The engine, the tool, and the all-weather tires -- Unity takes over the world -- Why choose Unity? -- Why burn this book and run away screaming? -- Browser-based 3D - welcome to the future -- Time for action - install the Unity Web Player -- Welcome to Unity 3D! -- What can I build with Unity? -- FusionFall -- Completely hammered -- Should we try to build FusionFall? -- Another option -- Off-Road Velociraptor Safari -- Fewer features, more promises -- Maybe we should build Off-Road Velociraptor Safari? -- I bent my Wooglie -- Big Fun Racing -- Diceworks -- Lovers in a Dangerous Spacetime -- Showcase Showdown -- Unity Technologies - "Made with Unity" page -- Kongregate -- The iOS App Store -- Walk before you can run (or double jump) -- There's

no such thing as "finished -- Stop! Hammer time -- Fight Some Angry Bots -- The wonders of technology! -- The Scene window -- The Game window -- The Hierarchy panel -- The Project panel -- The Inspector panel -- Heads Up? -- Layers and layout dropdowns -- Playback controls -- Scene controls -- Don't stop there - live a little! -- Summary -- Big ambition, tiny games -- 2. Let's Start with the Sky. That little lightbulb -- The siren song of 3D -- Features versus content -- A game with no features -- Mechanic versus skin -- Trapped in your own skin -- That singular piece of joy -- One percent inspiration -- Motherload -- Possible additional features: -- Heads up! -- Artillery Live! -- The skinny on multiplayer -- Bang for your buck -- Pong -- The mechanic that launched a thousand games -- Have a go hero - redesign your favorite games -- Toy or story -- Pop Quiz - finding that singular piece of joy -- Redefining the sky -- Summary -- Let's begin -- 3. Game #1 - Ticker Taker -- Kick up a new Unity project -- Where did everything go? -- 'Tis volley -- Keep the dream alive -- Slash and burn! -- The many faces of keep-up -- Creating the ball and the hitter -- Time for action - create the Ball -- What just happened - that's all there is to it? -- A ball by any other name -- Time for action - rename the Ball -- Origin story -- XYZ/RGB -- Time for action - move the Ball Into the "Sky -- Time for action - shrink the Ball -- Time for action - save your scene -- Time for action - add the Paddle -- What's a Mesh? -- Poly wants to crack your game performance? -- Keeping yourself in the dark -- Time for action - add a light -- Time for action - move and rotate the light -- Have a go hero -- Extra credit -- Are you a luminary? -- Who turned out the lights? -- Darkness reigns -- Cameramania -- Time for action - test your game -- Let's get physical -- Add physics to your game -- Understanding the gravity of the situation -- More bounce to the ounce -- Time for action - make the Ball bouncy -- Have a go hero -- Summary -- Following the script -- 4. Code Comfort -- What is code? -- Time for action - write your first Unity Script -- A leap of faith -- Lick it and stick it -- Disappear me! -- What just happened? -- It's all Greek to me. You'll never go hungry again -- With great sandwich comes great responsibility -- Examining the code -- Time for action - find the Mesh Renderer component -- Time for action - make the ball re-appear -- Ding! -- Time for action - journey to the Unity Script Reference -- The Renderer class -- Have a go hero - pulling the wings off of flies -- What's another word for "huh"? -- It's been fun -- Time for action - unstick the Script -- Gone, but not forgotten -- Why code? -- Equip your baby bird -- Time for action - create a new MouseFollow Script -- What just happened? -- A capital idea -- Animating with code -- Time for action - animate the Paddle -- What just happened - what witchcraft is this? -- Why didn't the Paddle animate before? -- Pick a word - (almost) any word -- Screen coordinates versus World coordinates -- Move the Paddle -- Worst. Game. Ever. -- See the matrix -- Time for action - listen to the paddle -- A tiny bit o' math -- Tracking the numbers -- Futzing with the numbers -- Time for action - Log the New Number -- She's a-work! -- Somebody get me a bucket -- Time for action - declare a variable to store the Screen midpoint -- What just happened - we've gone too var -- Using all three dees -- Time for action - follow the y position of the mouse -- A keep-up game for robots -- Once more into the breach -- Time for action - revisit the Unity Language Reference -- Our work here is done -- Time for action - add the sample code to your Script -- One final tweak -- What's a quaternion? -- Wait, what's a quaternion? -- WHAT THE HECK IS A QUATERNION?? -- Educated guesses -- More on Slerp -- Right on target -- Have a go hero - time to break stuff -- Keep it up -- Beyond

the game mechanic -- C# Addendum -- 5. Game #2 - Robot Repair -- You'll totally flip -- A blank slate -- You're making a scene -- Time for action - set up two scenes.

No right answer -- Time for action - prepare the GUI -- The beat of your own drum -- Time for action - create and link a custom GUI skin -- What just happened? -- Time for action - create a button UI control -- What just happened? -- Have a go hero - no sense sitting around on your button -- Want font? -- Cover your assets -- Time for action - nix the mip-mapping -- Front and center -- Time for action - center the button -- What just happened - investigating the code -- The waiting game -- The easiest button to button -- To the game! -- Time for action - add both scenes to Build List -- Set the stage for robots -- Time for action - prepare the game scene -- The game plan -- Have some class! -- Time for action - store the essentials -- A matter of great import -- Building a better bucket -- How big is your locker? -- Start me up -- Going loopy -- The anatomy of a loop -- To nest is best -- Seeing is believing -- Time for action - create an area to store the grid -- Have a go hero - don't take my word for it! -- Build that grid -- What just happened - grokking the code -- Now you're playing with power! -- C# addendum -- 6. Game #2 - Robot Repair Part 2 -- From zero to game in one chapter -- Finding your center -- Time for action - centering the game grid vertically -- What just happened? -- Time for action - centering the game grid horizontally -- What just happened - coding like a ninja -- Down to the nitty griddy -- Do the random card shuffle -- Time for action - preparing to build the deck -- Let's break some robots -- Time for action - building the deck -- What just happened - dissecting the bits -- Time for action - modifying the img argument -- What just happened? -- What exactly is "this"? -- Have a go hero - grokketh-thou Random.Range()? -- Random reigns supreme -- Second dragon down -- Time to totally flip.

Time for action - making the cards two-sided -- Time for action - building the card-flipping function -- Time for action - building the card-flipping function -- What just happened - dissecting the flip -- Pumpkin eater -- What just happened? -- Stabby McDragonpoker rides again -- Game and match -- Time for action - ID the cards -- What just happened? -- Time for action - comparing the IDs -- What just happened? -- On to the final boss -- Endgame -- Time for action - checking for victory -- What just happened? -- Have a go hero - extra credit -- Endgame -- Bring. It. On. -- C# Addendum -- 7. Don't Be a Clock Blocker -- Apply pressure -- Time for action - preparing the Clock Script -- Time for more action - preparing the clock text -- Still time for action - changing the clock text color -- Time for action rides again - creating Font Texture and Material -- Time for action - what's with the tiny font? -- What just happened - was that seriously magic? -- Time for action - preparing the clock code -- What just happened - that's a whole lotta nothing -- Time for action - creating the countdown logic -- Time for action - displaying the time on-screen -- What just happened - what about that terrifying code? -- Picture it -- Time for action - grabbing the picture clock graphics -- What just happened - you can do that? -- Time for action - Flex those GUI muscles -- What just happened - how does it work? -- The incredible shrinking clock -- Keep your fork - there's pie! -- Pop quiz - how do we build it? -- How they did it -- Time for action - rigging up the textures -- Time for action - writing the pie chart script -- What just happened? -- Time for action - commencing operation pie clock -- What just happened - explaining away the loose ends -- Time for action - positioning and scaling the clock -- Have a go hero - rock out with your clock out.

Unfinished business.

Sommario/riassunto

This is a practical and light-hearted guide to get to grips with creating your first games, with easy-to-follow, step-by-step tutorials using the award winning Unity engine. If you've ever wanted to enter the world of independent game development but have no prior knowledge of programming or game development, then this is the book for you. Game developers transitioning from other tools like GameMaker and Flash will find this a useful tool to get them up to speed on the Unity engine, as will anyone who has never handled the Unity engine before.
