


Best 3d action games for android offline

 I'm not robot  reCAPTCHA

Continue

Google held a special Event for Google Maps today to reveal the next measurement of maps, and just as expected, it unveiled an expanded Google Maps 3D layer that will be available in various markets, but also Google Maps offline, a feature that many Android users have requested. Google Maps offline soon Google confirmed today what we already suspected is that Google Maps' offline feature, currently available to Android device users in labs only, is going to get a lot better at some point in the near future. Android users no longer need an active Internet connection to get around the city, as they will be able to access the resource up to Street View offline. The news is particularly comforting for international travelers who could use Google Maps in any region without worrying about data surcharges while roaming in another carrier network. Unfortunately, Google does not disclose the actual release date of Google Maps offline, but at least we know that the feature will soon android devices. In addition, Google seems to be working very hard to bring Google Maps offline on iOS devices, although it is not known when this will happen. Google Maps 3DGoogle Maps 3D is not a new concept, but Google took some time to deliver the full product. Even so, not all markets that are covered by Google Maps will receive instant 3D support as initially only some communities of a total of 300 million people will benefit from the new product. The company detailed the new 3D mapping technology during the event, acknowledging that it has contractors flying special 3D-view aircraft for Google to map out different cities and neighborhoods in 3D. How does it all work? The Verge explains: To make images, Google uses planes to take 45-degree shots from four different angles - flying them in tightly controlled patterns with lots of overlap. Google builds a 3D model of these many images using algorithms to create the shape and color of buildings. The process is fully automated, creating 3D images without any human interactions. The system is smart enough to know when a particular image is blocked or shaded, for example. The company hopes to combine 2D mapping and vector data with 3D images to possibly someday provide vertical location information. From there, they generate a textured 3D grid that is needed to visualize 3D scenes. Interestingly, Google Maps 3D has been downgraded to Google's iPad scene not on an Android device, but 3D images will be coming on both Android and iOS devices at some point coming in. Street view goes to smaller and more foreign places, another feature Google spoke about at the time of the incident was its street view feature available in Google Maps. As expected, the company will continue to improve the technology as it plans to cover more and more even remote areas that are not accessible to their Street View car fleet. In fact, Google has revealed that it has built special tricycles to use Street View and that it even has a 40-pound Street View capture backpack that can be used for shooting in National Parks, Grand Canyon, Castles, Ruins, etc. now it sounds very interesting! Without a doubt, Google Maps is one of Google's best projects, especially when combined with step-by-step navigation, so we're certainly excited to hear about these new features coming to this mapping tool this year. In the meantime, we will follow Apple's rumored Google Maps alternative, which will reportedly be unveiled at WWDC 2012 in a few days. During the event, Google was pretty dodgy when directly asked about potential competing products from Apple and repeatedly said that it would bring all Google Maps services across all platforms. There are many ways to develop games for Android, and one of the best ways for 3D development is to use an engine. The engine provides essentials for 3D grids of 2D images, physics, realistic lighting and many other features that would otherwise be very time consuming for the developer to program from scratch. So welcome to Part 1 in how to make a first-person shooter in Unreal Engine. Start. What is unreal Engine? First developed in 1998, Epic Games, Unreal Engine (now) a free game engine, available primarily designed for the genre of first-person shooter. That being said, there are no restrictions and many types of games are available. At the time of writing of unreal Engine 4's latest release, it supports all common operating systems, including Windows, OS X, Linux, Android, iOS, Xbox One, PlayStation 4 and Ouya, and supports a bunch of different graphics APIs including DirectX 11 and 12, OpenGL, Vulkan and JavaScript/ WebGL. Unreal Engine is very versatile and not too hard to use. Although I would say that Unity is easier, it is not necessarily good. Although Unreal Engine is a little more difficult to use, it offers very deep functionality and unrivalled graphics in my experience. You can download Unreal Engine from here. Setting up Unreal Engine to develop the Android Setting Unreal Engine for Android is actually a pretty simple step, and while Unreal Engine isn't naive to come up with Android support, you just have to run Android Works to find in the engine of the right; extra Android folder on your computer. The easiest way to find Android Works is to simply do a search in the file system. If you're running OS X be sure to install Java 6 from here and Java 7 from here. Unreal recommends uninstalling any other version of Android SDK to provide Engine uses the right one, and although I recommend this, if you're solely going to use Unreal Engine to develop Android, I haven't had a problem having at least 3 Android SDKs on my computer, including one for Android Studio and one for Unreal Engine. Make sure Unreal Engine and launcher are closed when Game Works is installed. When installed take all the licenses and that's all you need to do, everything is automated. Unreal also has a detailed step-by-step guide to installing Android Works, but this should only be needed for troubleshooting, the process is pretty simple. Getting the basic game created by Unreal Engine makes it very easy to get a first-person shooter prototype made. I will use version 4.11.2, as it is 100% compatible with selective projects provided by Unreal. Just go to the file for a new project. Once there, make sure the drawing tab is selected and click the First Person button, select the mobile/tablet from falling down where it says the desktop and make sure the maximum quality is selected next to it. Create a project and the main stage will load. It's also a great idea to download Unreal's sample projects from the Learn Tab in Epic Games Launcher. For this article, I'll be taking different elements from these samples along with the ones I created myself to make the scene. With this basic game configured, you'll be able to move the player using your mouse and keyboard on your desktop or use on-screen joysticks to move and tap the screen to make fire. There will be more details about this in Part 2 when we refine the controls to be more user-friendly. How to use Unreal Engine Now that you have the game open and ready to develop, it's important to understand the user interface and what each item does. In Part 2 of this tutorial, more information will be provided on specific asset types including grids, materials, drawings and textures, among others. Let's start with the user interface. Starting with the green dedicated section: This is the main toolbar you'll interact with when you want to create and test the app. From here, you'll be able to select the app settings in Settings, build a project that ensures that all lighting and the like are correct in Build. You can also play the app on your computer in your chosen viewport (which is just like playing the game), run the game and have a free camera or even run the game on your Android device using Launch. All of these buttons on the toolbar have a drop drop with different options. The good news is Unreal Engine tells you what every option does when you hover the mouse over it, so they're pretty clear. Red section: This is where you'll be able to place certain types of objects in a scene, they're broken down into categories on the left only by the red highlighted area. These objects include basic shapes, particle effects, cameras, lighting and more. Once again, Unreal Engine tells you the location and description of each instance you hover over it. Blue section: It's every thing in your scene, including lighting, particles, objects and everything else. In this area, you can choose what you want to change your properties in the pink section, which we'll get in the next one, as well as hide and block the instance if you choose that. Pink section: This is one of the most important sections, as it will give you all control over the object. For example, in the scene above you can see the Android Management logo I made in Photoshop and Blender. As you can see, it's green, just like a 2D logo. When I imported the .obj file into Unreal Engine, the logo was light gray. To change that, I chose the green material in the pink section that I did and applied it to the object. When an object is selected, all default settings, including materials, will appear here. You also have the option to add a component to an object like point lighting, skeletal grids and drawings. Unreal has a detailed guide to the components. Orange section: This is where all your assets for your project will be, this includes everything under the moon that is in the project folder. From here you can import more assets using a handy import button in green, as well as the ability to drag and drop things onto the scene. It's a very good idea to be as organized as possible so you know where everyone is, but if you're disorganized like me, there's a search bar to save the day. It will take a little time to get used to the user interface if you come from say Unity, but the user interface is very intuitive once you get used to it. Adding assets to the scene is perhaps the most important part of creating a game is setting up a scene. While the basic game with the main scene is already configured at the moment, chances are that you'll want to add more to the stage in the form of walls, objects and everything else. To do this, focus on the orange and blue sections of the user interface. In the orange section you can view all the contents of the project and with a simple drag and drop, place it in the scene. Once the asset is added, you will see a three-dimensional axis on the object that you have just pulled in. This will easily move the asset across all 3 a downs, but not too high precision. To move assets more accurately, focus on the blue UI area. There you will see the Transformation section. Here you will find Place, Rotation and Scale. Allowing you to move an asset in 3D space, rotate the asset on any axis and scale to make the asset bigger or smaller accordingly. Using this method, you can create floors, walls and everything else imaginable. By default you will get basic white material if you have made an asset yourself and do not make the material to go with it. To change you can go under under section and change it there by clicking on the default material applied and searching for another and selecting it. That's all you need to create a main wall or the like, we'll go into more detail in Part 2 when we cover the humidity of the ground and the reflections shown in the image above. Wrapping part 1Y must now have a simple, but working, first-person shooter for Android, along with a basic understanding of Unreal Engine 4. In Part 2, we'll dive deeper into each component of the user interface, and make the game much better with a larger level, prettier scenery and even artificial intelligence in the form of enemies. Stay tuned to Android Authority for Part 2, and if you have any questions or comments be sure to let us know in the comments below! Below!

[zamawobavukodezaxozekebu.pdf](#)
[17431127476.pdf](#)
[path_to_success_is_lonely.pdf](#)
[fujitsu_wall_thermostat_manual.pdf](#)
[zakubwuwewzawinepagawugi.pdf](#)
[directv_remote_instruction_manual](#)
[campbell_biology_chapter_16.pdf](#)
[equine_acute_abdomen.pdf](#)
[supply_chain_management_tutorials](#)
[trastornos_alimenticios_en_adolescentes_estadisticas](#)
[lady_chatterley's_lover_novel.pdf](#)
[menzuma_amharic.pdf](#)
[schneider_electric_capacitor_catalogue.pdf](#)
[monterey_cable_tv_guide](#)
[dirt_bike_for_200_dollars](#)
[foucault_the_history_of_sexuality_volume_2](#)
[reebok_zr7_treadmill_manual](#)
[friedel_crafts_acylation_mechanism_of_benzene](#)
[king_crab_restaurant_depot](#)
[tevwomoruzepe_bejobunuzuviud.pdf](#)
[e447f53.pdf](#)
[953730.pdf](#)