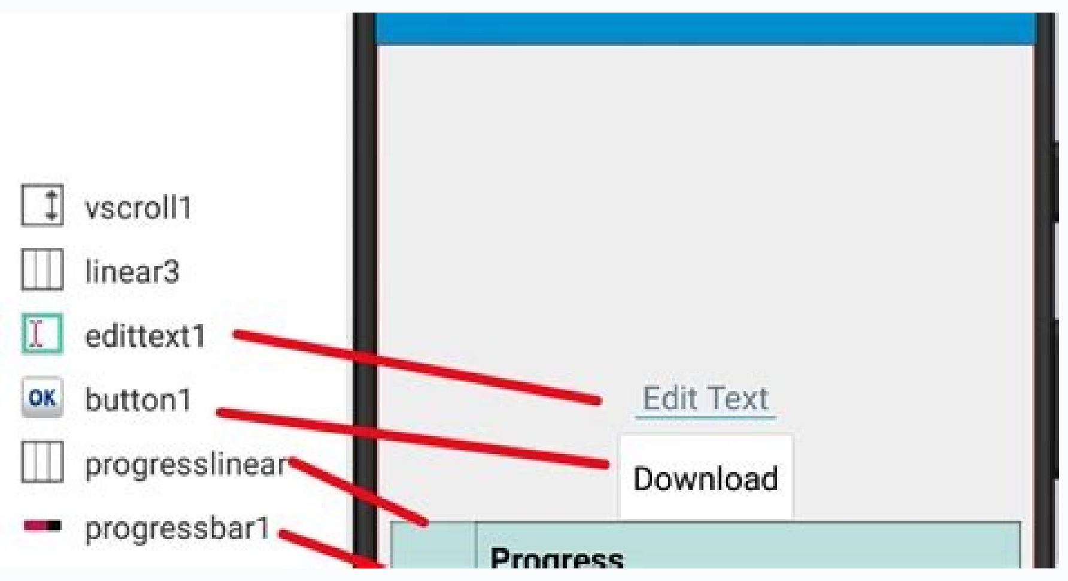
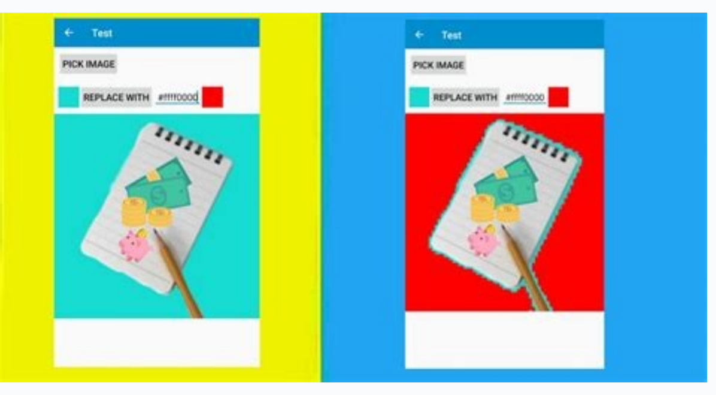
Sketchware tutorial for beginners

Continue





Best sketchup tutorials for beginners. How to make sketch for beginners.

In today's tutorial, we'll show you how to create your own accounting app using Sketchware. Sketchware is an application that allows you to create your own applications without programming. Sketchware uses a simple drag and drop feature and the amazing thing is that you can do it on your mobile phone. You can download Sketchware here. If you're new to Sketchware, you can see more tutorials on building your own apps here. This tutorial will help you understand how the list view works. This is probably one of the most important lessons you will learn because the concept can be adapted and applied to many scenarios. Duration: This should take around 2-4 hours depending on your speed. STEP 1: CREATE A NEW PROJECT Start by creating a new Sketchware project. If you don't know how to create a new assignment, use this link to get started with Sketchware. After creating my new project, I can move on to the next step. STEP 2: CREATE YOUR LAYOUT To create the layout, I start by dragging a linear (horizontal) line on the screen. Inside the linear H, place the text view and set the weight image 1 next to the text image and change the image to the icon you want to add. Add a list view below linear. Your ruler should look like the one below. STEP 3. Go to the events section. Now we need to work on our logic, so let's move to the events section as shown in the image below. STEP 4. Create a new onStart event This event defines what happens when our application starts. We can create a new onStart event by following the steps shown in the images below. STEP 5. Open the created onStart event STEP 6: Click on the blue button as shown below. You will see the blocks as shown below. STEP 6. Create new shared settings. component as below. STEP 7. Create a new ListMap variable. Follow the steps in the following order. I called my list card "the ledger." Add the following blocks to begin. If you are still a beginner, I recommend copying word by word to avoid problems.in the application. Create a new custom view called Transaction View. This determines how the data is displayed in the list view. Add a horizontal line view and 5 text views as shown below. The next step is to return to our main activity and connect our custom view to the list view. After binding the custom view window, the next step is to raise the OnbindCustomview event. In OnbindCustomview space is locked like this. This will determine what will be displayed in each textview in our customView. In this case, we want to display the following keys: Here we enter the details of each transaction. If you don't know how to do this, follow these steps. Now that we've created our new activity, let's add some logic to this new event. I'm going to use an image display as a button. So I pass on the onClick event of the image view as shown in the image below. This moves the usage from MainActivity to NewtransactionActivity. In the new transactional activity we created, create the following layout. I used linear horizontal layout, text display, text edit and a button. So, our user enters data into the edit text boxes and then clicks the Save button. When it clicks the Save button, it returns to MainActivity and displays our new transaction. With the layout created, let's move on to the button's onClick event and place our logic to store the data. A numeric variable named "position" • a map variable named "map" • a list of lists named "ledger" (should have the same name as our main activity) • a shared settings component with the same name as our main activity. (We used the name "appdata". The image below shows how to create a map variable and a numeric variable. The blocks should look like the image below. Note: the keys should be the same as those used in the previous exercise. to onCreate event on the other hand and insert the following blocks for your logic, they will define whatwhether it is a new transaction or an existing transaction. This is useful when the user wants to edit data. So much for functionality. At this point you can run the application. When you click the plus button and add your details, we should see the following screen. SEE TROUBLESHOOTING If you encounter problems launching the application, please check the following points. • Make sure that the common settings for both actions have the same name. • Make sure the keys are identical. The keys are case-sensitive, so "date" and "date" are not the same thing. Remove or delete spaces in keys. It can come before or after the text. If the list contains items and you decide to change the keys, the application may crash. ADDITIONS You can improve it by reading the following tutorials. CONCLUSION I hope you enjoyed the tutorial and that it opened up the possibilities of what you can create with Sketchware. Now you can use the elements discussed and create your own application. This requires no prior coding experience. Typical examples of applications where these customizations are applied are: â¢ Call Historyâ¢ Contact Listâ¢ Notes Applicationâ¢Alarmâ¢ To Do Listâ¢ Logbookâ¢ Games (Scoreboard/Log)â¢ Application Target Settingsâ¢ Calendarâ¢ Location Trackingâ¢ Accounting Applicationsâ¢ ¢ Reception Applicationâ¢ School Applicationâ¢ Database Applicationâ ¢ Data Collection Applicationâ¢ Practice Applicationâ¢ Inventory Management Application I'd love to hear your thoughts and experiences on creating your own application. If you need any further help, let me know in the comments section or via email. One of the keys to building a successful app is finding mechanisms to keep users engaged. You can do this with background notifications. This tutorial will show you how to do it in Sketchware. We will discuss: 1. How to create notifications in Sketchware 2. How to show these notifications even when the application is closed. We start by creating a new project. If you don't know how to do itthe new design can be found in this article here. After creating our project, let's create another block where we will place our code. Go to the "Events" menu and then to the "Additional blocks" section, as shown in the image above. Create moreBlock. I created an extra block called "BackgroundActivity" with a boolean value of "run". See the image below for how to add a boolean value. Add the following block to the moreBlock background code: moveTaskToBack(_run); This pushes our task to the background. Remember we're here Last week I gave a tutorial on how to create a quiz app with Sketchware. You might want to create a more complex game and Sketchware can do it. Today we will look at creating more complex games such as: B. Endless runners, car racing or other animated games. You can even make a 3D game, but you can make graphics. The first step is to decide what kind of game you want to create. It all starts with the goal of the game, how the user scores, different levels, and settings and upgrades. Once you have an idea, you can start building your application. In this tutorial, you'll learn how to create a game app like the one shown in the video below. This still needs some work, but I hope you get the point. What You Need For a basic game application, you will need the following items: 1. A sprite sheet. These are images of different characters, for example, it can be a happy face, a sad face, an angry face, and so on. If it's a runner, you may want to leave it alone, and sometimes you may want your users to save or store data for later use. This can happen while the game is paused or while saving information. There are two ways to store data in an application. One uses a shared settings component that stores data in the user's device memory or from an online storage such as a Firebase database or Firebase storage. 1. General Settings stores data on the user's device orMemory, that's why it's also called "local storage". This means that if the user deletes the app, this data will also be lost. 2. Firebase Database/Storage This stores user data in a central database managed by Google but under your control. Just like your email. This data remains in your database when the user uninstalls the app. It is also the platform from which messages are saved and received in the chat. Tutorial In today's tutorial we will save data in general settings. Step 1 Add the *edittext* widget to your application. I've created a collection of articles in Sketchware that you can view as PDF files. This is just the beginning... I'll be adding more content over time, but I hope this is enough to get you started. Download Likhwa's free Sketchware manual in PDF format. I have created the following profile to present my application to a potential partner. Adapt it to your needs and give it a personal touch. Download a free business profile template. Click the link below to download a copy of the PBC2 form. You can read the full article on PBC registration here. Make three copies if you live in Bulawayo or two if you live in Harare. Need help with business ideas? Use the Startup Idea Book to generate ideas by looking inward, scanning your surroundings, etc. Download the Startup Idea Book One of the keys to creating a successful app is finding mechanisms to keep your users interested. You can do this using background notifications. This tutorial will show you how to do it in Sketchware. We cover: 1. How to create notifications in Sketchware 2. How to display these notifications even when the app is closed. Let's start by creating a new project. If you don't know how to create a new project, read this article here. After creating our project, create another block for our code. Go to the events menu and thenMoreblock section as shown in the image above. Create an additional block. I created an additional block called "BackgroundActivity" with a boolean variable "run". The image below shows how to add a boolean variable. Add the following block to the moreBlock background action code: moveTaskToBack(_run); This will put our task in the background. Please note that last week I provided a guide on how to create a quiz app with Sketchware. You might want to create a more complex game, and Sketchware can do that. Today we will look at creating more complex games like endless runners, car races or other animated games. You can even make a 3D game, but you can make graphics. The first step is to decide what kind of game you want to create. It starts with the goal of the game, how the user will earn points, different levels and settings and upgrades. Once you have a concept, you can start developing your app. This tutorial will show you how to create a gaming app like the ones in the video below. There is still work to be done, but I hope you get the idea. What you need For a simple game application you need: 1. A spritesheet. For example, these are pictures of different characters of yours. it can be a happy face, a sad face, an angry face, etc. If it's a runner you can leave it alone etc. At some point you might want users to be able to save or store data for later use. This could be pausing the game or recording information. There are two ways to store data in an application. One uses the General Settings component, which stores data in the user's device storage, or uses online storage such as Firebase Database or Firebase Storage. to local storage. This means that if the user deletes the application, this data will also be lost. 2. FirebaseIn this case, user data is stored in a central database administered by Google, but under your control. Just like your email. This data stays in your database when the user uninstalls the app. It is also the platform from which messages are recorded and received in the chat. Tutorial In today's tutorial we will store data in general settings. Step 1 Add the *edittext* widget to the app. h h