

LESSON 4 NOTES
&
WORKBOOK

Mobile App Development Part I



shawacademy

Software Development Life Cycles

There are various approaches to Software Development. These include the approaches to defining and designing the project, which methods are employed during development and the overall timeline for a working prototype to be completed. Each process model follows a particular life cycle in order to ensure success in process of the software under development. Two of the most Popular Models are as follows:

Waterfall Model

The Waterfall Model was the first Process Model to be introduced. It is a linear-sequential life cycle model. It's simplicity means that each stage is completed and a review takes place to ensure the project is on track to be a success or if the project is behind, if it should be discarded.

Advantages

Simple and Easy to Understand & Use

Easy to Manage due to the rigidity of the Phases in the Model, Reviews are also held after each specific deliverable has been completed.

Each phase is processed and completed one at a time, Phases do not overlap.

Small Projects where requirements are understood benefit hugely from this Model.

Disadvantages

Once a Project has reached the Testing Phase in the later stages of the App's Life Cycle, it is very difficult to go back and make

changes for components that have not been well-thought out during conception.

No working prototype until late in the life-cycle.

Huge amount of Risk & Uncertainty

Not recommended for complex projects and object-oriented projects.

Poor Model for long and ongoing projects.

Not suitable for projects where requirements have a high risk of changing.

Agile Model

Agile Development is a type of Incremental Model. Software is developed in incremental, rapid cycles. This results in small releases or changes that build on previous prototypes during development. Each release is thoroughly tested to ensure quality is maintained throughout development. Tasks are defined each week using a Backlog of Tasks to be Completed. A Sprint is then executed where the tasks for the week are discussed and implemented before the week's end.

Advantages

Customer Satisfaction from rapid, continuous delivery

People & Interactions are heavily emphasized rather than an after thought

Working software is delivered within weeks (MVP)

Communication is clearer – Face-to-face with all parties involved.

Continuous attention to technical excellence and good design throughout.

Late changes in Requirements are welcomed as well as regular adaption to changing circumstances.

Disadvantages

Can be difficult to assess the effort required in the beginning stages of Development

Lack of emphasis on necessary design and documentation

Project is easily taken off-track from unclear outcomes during design.

App Inventor 2 – Dice App

There are many ways to learn how to include variables, decisions and procedures into programming. The easiest way, is to do lots of examples until it is second nature to you! In the following example, we'll create a Dice-Rolling App that Randomly Generates a Number from 1 – 6 by shaking the Device. (WARNING: An Android Device is required to Test the Accelerometer in this App)

Screen1 Properties: **AlignHorizontal:** Center

Title: Dice Rolla

Icon: RoyaltyFreelImage.png

ScreenOrientation: Portrait

BackgroundColor: Black

Components:

Label InstructionLabel **FontBold:** Selected

Text: "Shake to Roll the Dice"

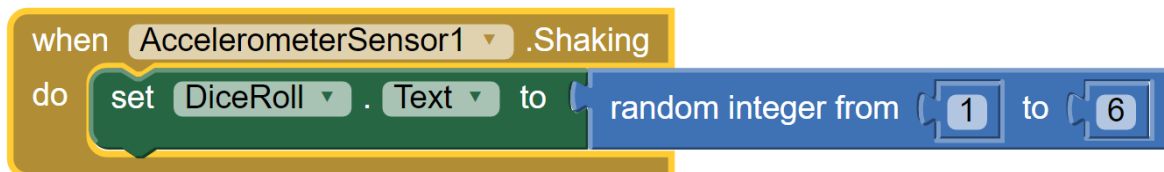
TextAlignment: Center

Label	DiceLabel	FontSize: 100
		FontBold: Selected
		Text Color: Green
AccelerometerSensor	Sensor1	Enabled

Step 1: Adding the Blocks

The Random number or **integer** as it's known in programming terms is located in the "Built-In" Blocks under the Math Blocks. Plug in the Integer Blocks for 1 & 6 and the block will generate a randomly selected number from 1 – 6.

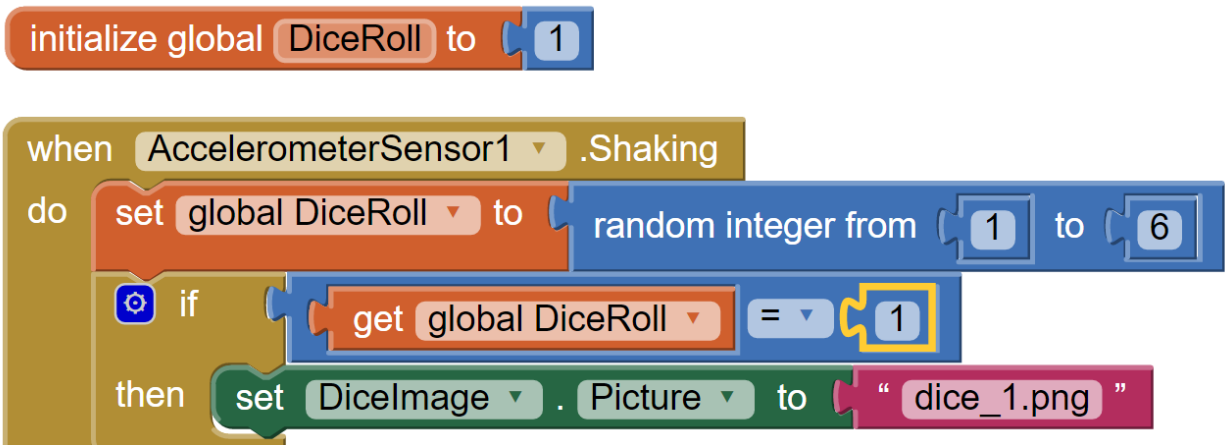
Next, we can set the blocks to jump into action **When** the Phone is shaken by adding the block from the accelerometer blocks in the "Component-Specific" Section. Select the DiceRollLabel block to change the text to the Random Integer.



Step 2: Taking it Further

Try Adding Images in to the Project to make the Dice seem more realistic. Here we can use a Variable to Store the Random Integer Number when the Phone is Shaken.

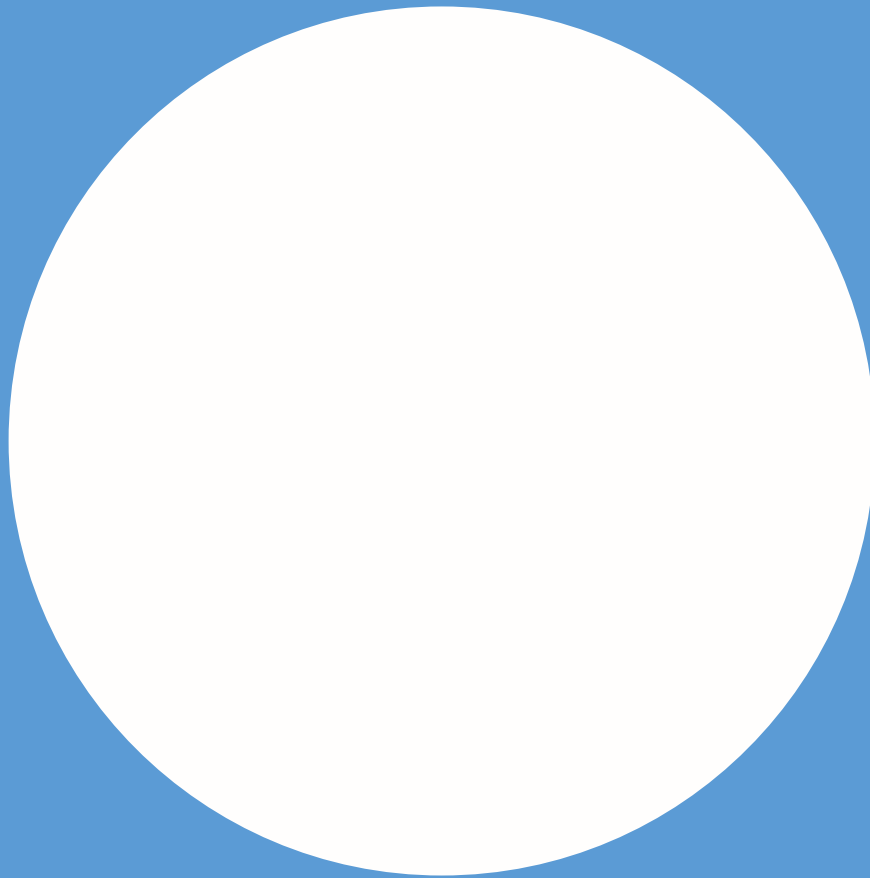
Then, we can use our If Then Blocks to set the DiceImage to the correct image if the random number is equal to it.



Repeat the If Then Blocks for each image.

Step 3: The Logical Way

In Programming Terms – Code is always modular. That means it is concise and to the point. There's no code used that could be done differently faster. While this is a great way to experiment with if Statements to see the outcomes, can you think of a better way to change the image according to the random integer when the phone is shaken?



mobile.app@shawacademy.com



shawacademy