VentureDive

Flutter Training

ANIMATIONS

Agenda

- Types of Animation in Flutter
- Implicit Animation
- Practical example for Implicit Animation
- Custom Implicit Animation
- Practical demonstration for Implicit Animation
- Essential Animation Concepts
- Explicit Animation
- Practical Examples for Explicit Animations
- Lottie/Flare animations

Types of Animation in Flutter

- In flutter there are two types of animations
- Implicit Animation
  - Predefined animation widgets provided by flutter library are known as implicit animation widgets.
  - These widgets contains the properties of animation with which user can play.
  - In flutter all implict animated widgets inherits the class ImplicitlyAnimatedWidget.
- Explicit Animation
  - There are many situations in which we require custom animations, these user defined animation are called explicit animation which are implemented by exploiting animation library..

Implicit Animation:

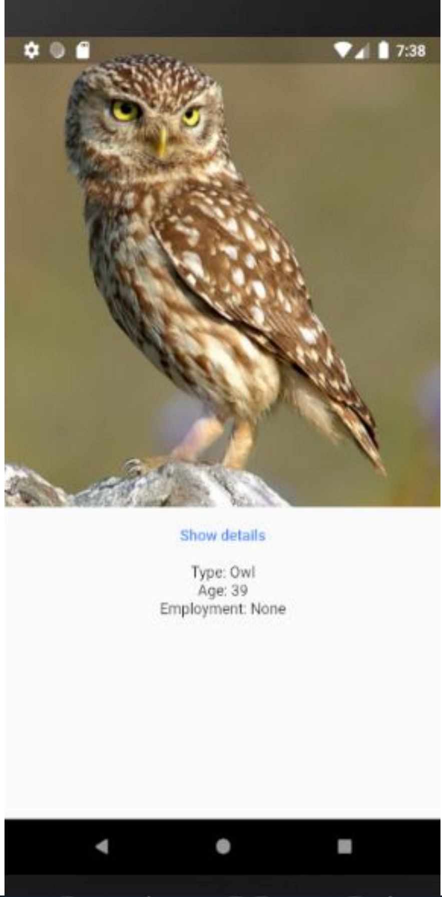
- One of the most easiest part of the animations
- User to perform basic animations by using widgets given by flutter class.
- When to use
  - When animation is only in one direction. That is either forward or backward.
  - Animation does not repeat.
  - Animation is not continuous or infinite.
  - The fluter already provide the animated version of the required widget.

Implicit Animation: Widgets Vs Animated Widgets

- Align = AnimatedAlign
- Container = AnimatedContainer
- Opacity = AnimatedOpacity
- PhysicalModel = AnimatedPhysicalModel
- Positioned = AnimatedPositioned
- DefaultTextStyle = AnimatedDefaultTextStyle

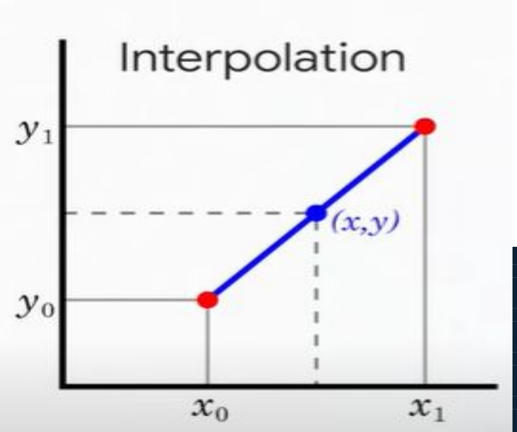
Implicit Widgets Demo

- All implicit widgets has animation properties which user can exploit
- The process of transforming from old to new state of position or angle values are called interpolation.
- We will check this properties in the demo.



Show details

Type: Owl
Age: 39
Employment: None



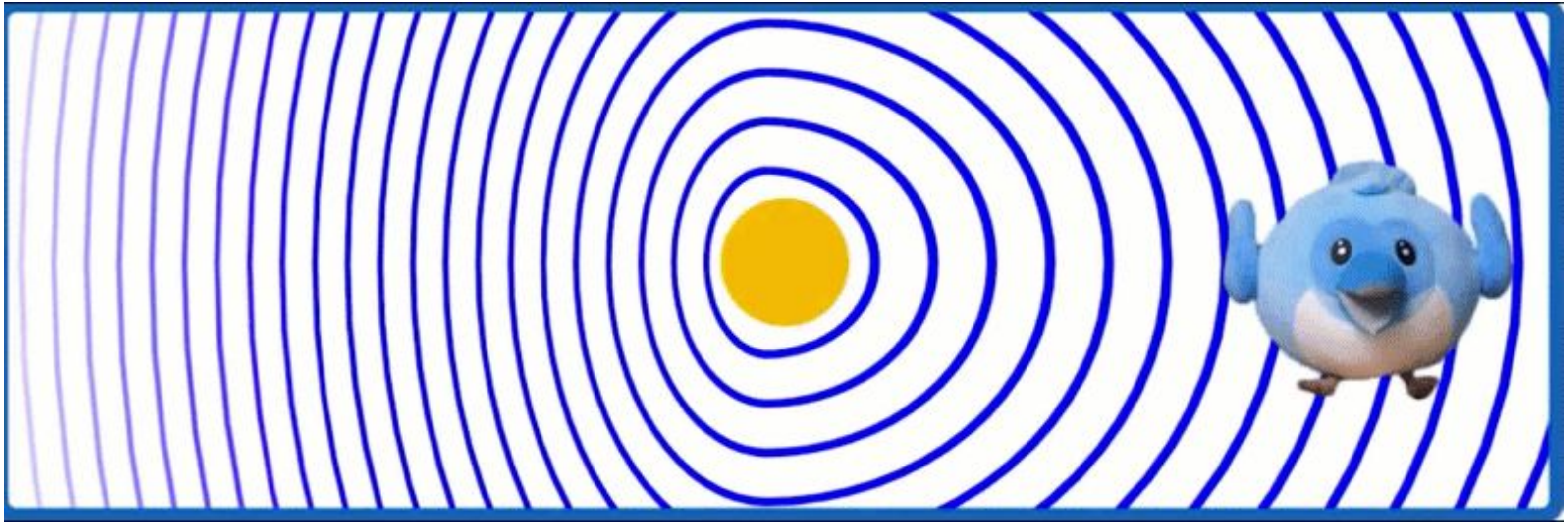
Custom Implicit Widgets

- you want to create a basic animation
- animation that doesn’t repeat forever
- animation that is not infinite or continuous
- It is just one widget or widget tree
- You can not find it in implicit animations given by flutter
- Then you can create custom implicit widget by using flutter class TweenAnimationBuilder.

Custom Implicit Widgets TweenAnimationBuilder

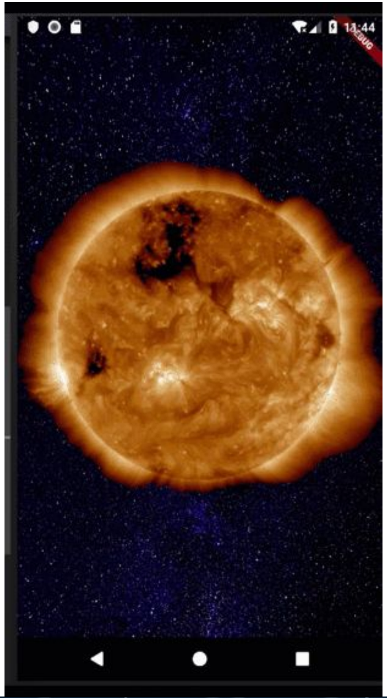
- Takes three parameter
- duration parameter, Time duration for rendering the given animation.
- The range of values to interpolate named as Tween parameter
- The builder parameter, which returns what my animated widget will look like at a given moment in time
    1. This builder function takes a parameter that is the same type as your Tween values, which basically tells Flutter what the current animation value is at a given moment.

Custom Implicit Widgets Example: Doppler Effect



Custom Implicit Widgets Example: Doppler Effect

- We will first give a practical demonstration for using TweenAnimationBuilder class of flutter.
- Then finally we will present the concept of doppler effect and code an animation that will animate the colors of solar flares.



Essential Animation Concepts

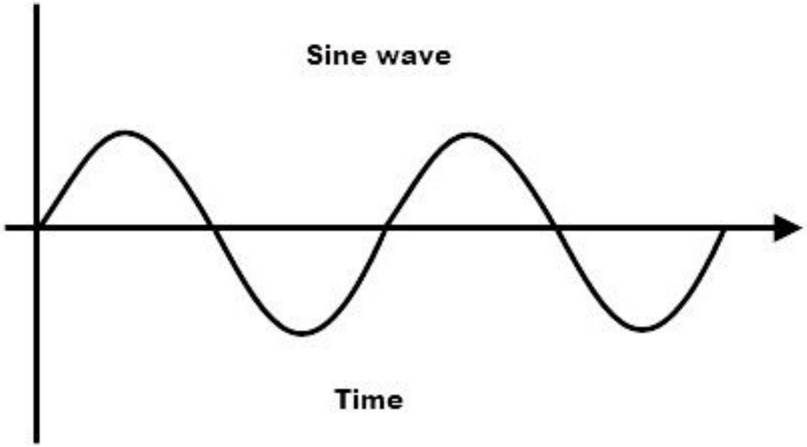
- Animation: a core class in Flutter’s animation library, interpolates the values used to guide an animation.
- Animation Controller: a subclass of Animation. It has a job of managing the behaviour of animation.
- Tween: interpolates between the range of data as used by the object being animated
- Listeners: to monitor animation state changes.

Animation

- knows nothing about what is onscreen.
- abstract class that understands its current value and its state (completed or dismissed)
- An Animation object sequentially generates interpolated numbers between two values over a certain duration.
- The value can be of any user specified type example:
    - Double
    - Color

Example: Curved Animation a sub type of Animation

- Apply Non linear transformation by generating values in non linear sequence.
- `animation = CurvedAnimation(parent: controller, curve: Curves.easeIn);`
- `class ShakeCurve extends Curve {`
- `  @override`
- `  double transform(double t) => sin(t);`
- `}`



More Examples of Curved Animation

- https://api.flutter.dev/flutter/animation/Curves-class.html

Animation Controller

- AnimationController derives from Animation<double>
- Generates a new value whenever the hardware is ready for a new frame.
- By default, an AnimationController linearly produces the numbers from 0.0 to 1.0 in a specified duration.
- `Example:`
-
  ```
  controller = AnimationController(duration: const Duration(seconds: 2), vsync: this);
  ```
- The presence of vsync prevents offscreen animations from consuming unnecessary resources.

Tween

- A Tween is a stateless object that takes only begin and end.
- The sole job of a Tween is to define a mapping from an input range to an output range
- The input range is commonly 0.0 to 1.0,
- By default, the AnimationController object ranges from 0.0 to 1.0. If you need a different range or a different data type, you can use a Tween to configure an animation to interpolate to a different range or data type.
- For example, the following Tween goes from -200.0 to 0.0:
- `tween = Tween<double>(begin: -200, end: 0);`

Listeners

- An Animation object can have Listeners and StatusListeners.
- A Listener is called whenever the value of the animation changes.
- The most common behavior of a Listener is to call setState() to cause a rebuild.
- A StatusListener is called when an animation begins, ends, moves forward, or moves reverse.

When to use Explicit Animation

- When the animation is repeated
- When the animation is continuous
- When the animation is complex and cannot be contained in a custom widget.

Code Walk Through

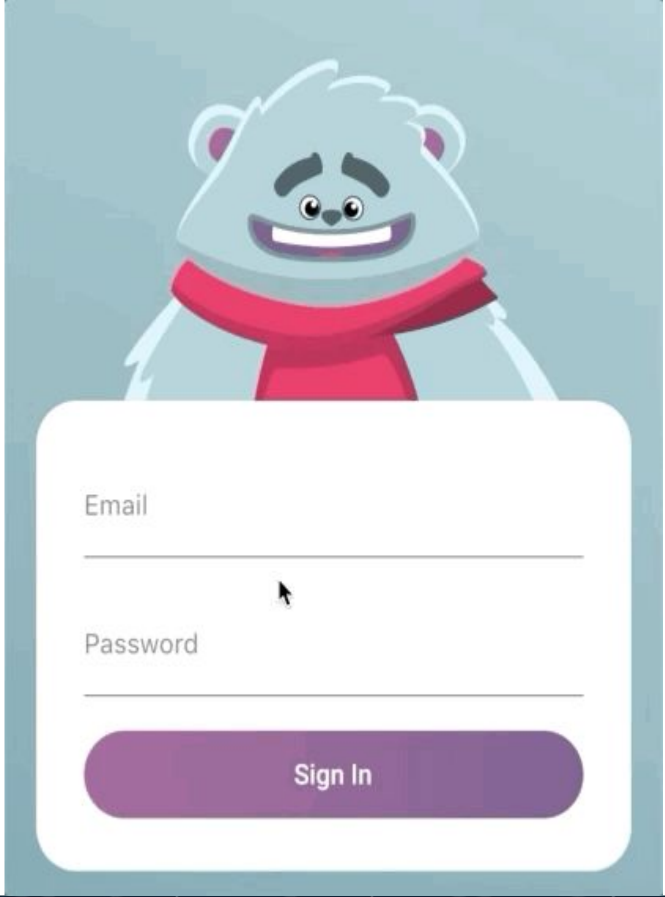
- In the demo we will use all of the four class defined above to produce some interesting animations.

Lottie Animations Demo

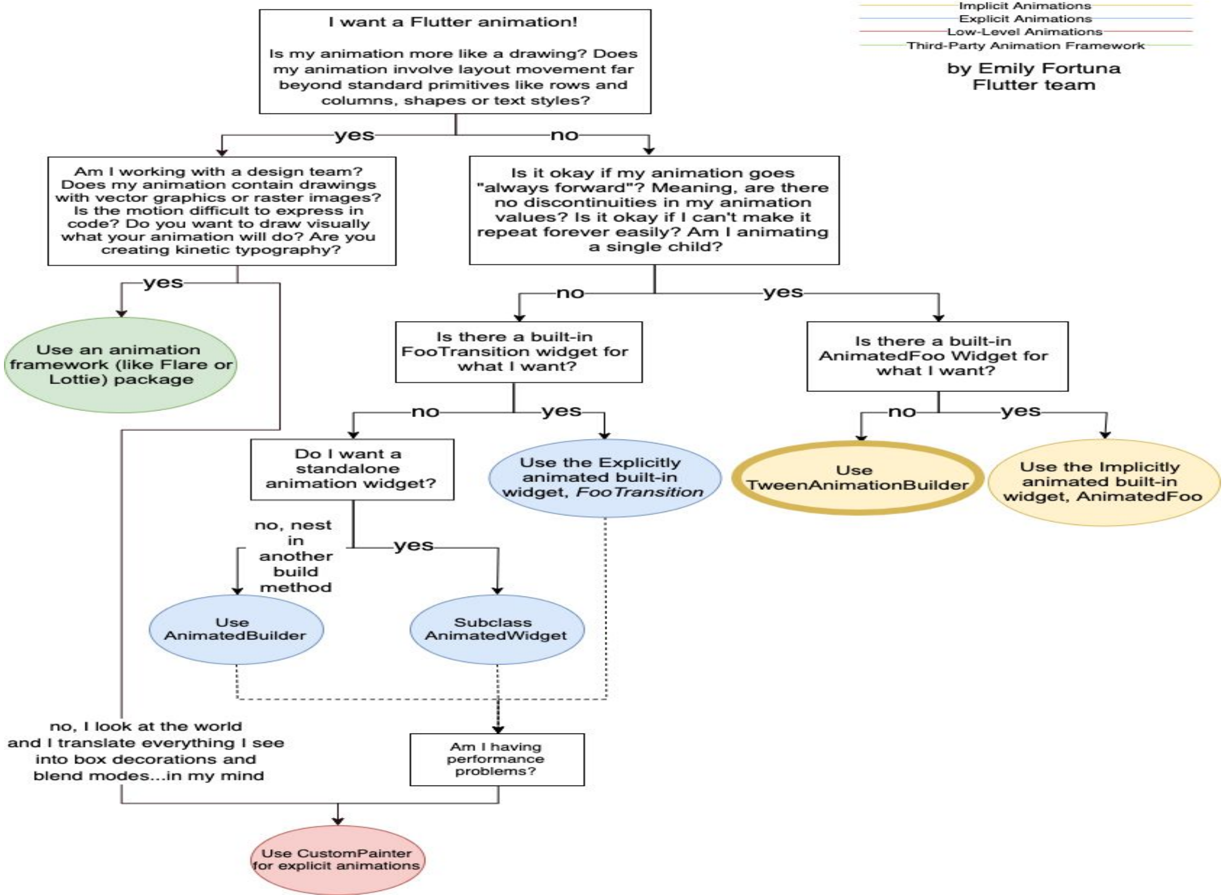
- In the demo we will use use a Lottie animation file for complex animations.

Rive/Flare Animations Demo

- User for advance controls over animations inside flutter app.
- src: Flutter Support (Updated) - Rive
- Video src:Use Rive and Flutter for dynamic, interactive, & animated experiences (Flutter Interact '19)
- Design Teddy flare file in rive: Teddy design and animation, made in 2Dimensions Flare by user JcToon
- Teddy flare file in rive: Teddy



When to choose which type of Flutter Animation



Implicit Animations
Explicit Animations
Low-Level Animations
Third-Party Animation Framework

by Emily Fortuna
Flutter team

I want a Flutter animation!

Is my animation more like a drawing? Does my animation involve layout movement far beyond standard primitives like rows and columns, shapes or text styles?

yes

Am I working with a design team? Does my animation contain drawings with vector graphics or raster images? Is the motion difficult to express in code? Do you want to draw visually what your animation will do? Are you creating kinetic typography?

yes

Use an animation framework (like Flare or Lottie) package

no, I look at the world and I translate everything I see into box decorations and blend modes...in my mind

no

Is it okay if my animation goes "always forward"? Meaning, are there no discontinuities in my animation values? Is it okay if I can't make it repeat forever easily? Am I animating a single child?

no

Is there a built-in FooTransition widget for what I want?

no

Do I want a standalone animation widget?

no, nest in another build method

Use AnimatedBuilder

yes

Subclass AnimatedWidget

yes

Use the Explicitly animated built-in widget, *FooTransition*

Am I having performance problems?

Use CustomPainter for explicit animations

yes

Is there a built-in AnimatedFoo Widget for what I want?

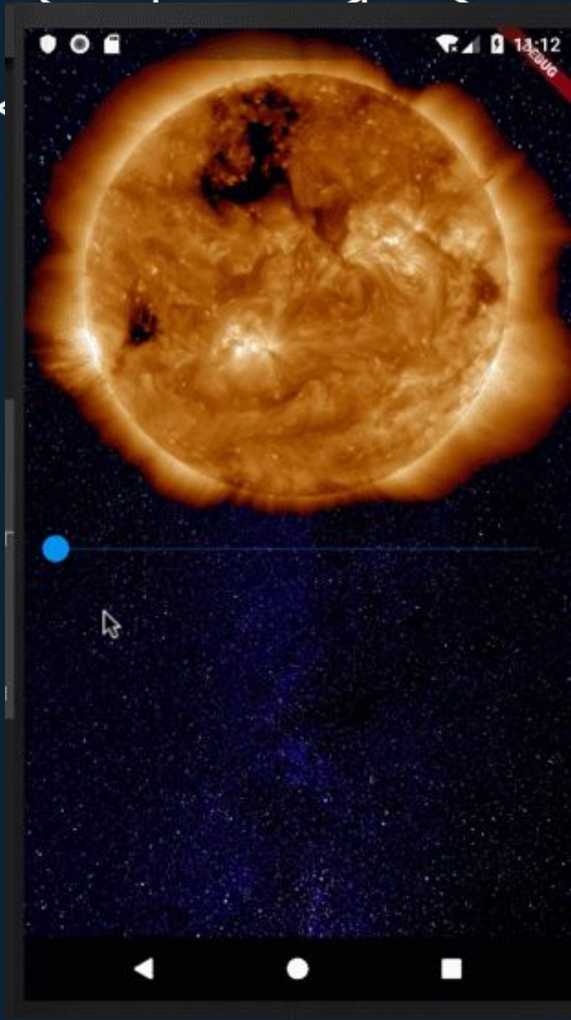
no

Use TweenAnimationBuilder

yes

Use the Implicitly animated built-in widget, AnimatedFoo

- Take home assignment
  Complete all Todos marked in the project.
- BONUS ASSIGNMENT:
  - Create a custom Implicit animation shown on right hand side to incorporate adaptive slider in animation which dynamically update the color animation relative to slider’s position
    - For reference you can check official flutter blog and video to solve this exercise.
      - [Custom Implicit Animations in Flutter…with TweenAnimationBuilder]()
      - [Creating your own Custom Implicit Animations with TweenAnimationBuilder]()



Assignment submission

Upload your code on github and submit it’s link on the Google chat group.

References

- https://www.youtube.com/playlist?list=PLjxrf2q8roU2HdJQDjJzOeO6J3FoFLWr2
- https://flutter.dev/docs/codelabs/implicit-animations
- https://pub.dev/packages/lottie#-installing-tab-
- https://flutter.dev/docs/development/ui/animations/tutorial#essential-animation-concepts-and-classes

VentureDive

End