

Our First Component

Ball

Draw itself

Starts at lower left
of screen

Moves

diagonally

across screen

Bounces off edge

Dies after
10
bounces

A Component in Flame

- The game engine interacts with your component by calling methods that you write inside that component.

A Component in Flame

- The game engine interacts with your component by calling methods that you write inside that component.
- The game engine runs the show. It decides when it wants the component to do something and calls the appropriate method.

A Component in Flame

- The game engine interacts with your component by calling methods that you write inside that component.
- The game engine runs the show. It decides when it wants the component to do something and calls the appropriate method.
- The code inside your methods determines how your component will behave.

Component Methods

- `resize()`

The game engine tells the component how big the screen is.

- `render()`

The game engine tells the component to draw itself.

It is called for every frame – up to 60 times per second

- `update()`

This is the time for your component to update itself for the next frame

- `destroy()`

The game engine asks your component if it's time to die.

Component Methods

- `resize()`

The game engine tells the component how big the screen is.

- `render()`

The game engine tells the component to draw itself.

It is called for every frame – up to 60 times per second

- `update()`

This is the time for your component to update itself for the next frame

- `destroy()`

The game engine asks your component if it's time to die.

Component Methods

- `resize()`

The game engine tells the component how big the screen is.

- `render()`

The game engine tells the component to draw itself.

It is called for every frame – up to 60 times per second

- `update()`

This is the time for your component to update itself for the next frame

- `destroy()`

The game engine asks your component if it's time to die.

Component Methods

- `resize()`

The game engine tells the component how big the screen is.

- `render()`

The game engine tells the component to draw itself.

It is called for every frame – up to 60 times per second

- `update()`

This is the time for your component to update itself for the next frame

- `destroy()`

The game engine asks your component if it's time to die.

Method Flow

That's it.

The game engine will continually call these methods, and your component continually responds in an appropriate way for the game.

Let's make our first component!

Make a new Flutter project and call it flutter_ball

- In folder test, delete the file widget_test.dart
- In folder lib, edit main.dart and delete everything in the file
- In folder lib, make file flutterball_game.dart
- In folder lib, make a subfolder called components
- In folder components, make file ball.dart
- Edit file pubspec.yaml

This file tells Flutter what libraries and assets are used

pubspec.yaml

```
name: flutter_ball  
description: A new Flutter application.  
version: 1.0.0+1
```

```
environment:  
  sdk: ">=2.2.0 <3.0.0"
```

```
dependencies:  
  flutter:  
    sdk: flutter  
  flame: ^0.14.0
```

```
dev_dependencies:  
  flutter_test:  
    sdk: flutter
```

```
flutter:  
  uses-material-design: true
```

Most of the lines will already be there. Make sure the flame dependency is there.

After you edit this file, click on tools → flutter → package get.
This will pull in the flame game engine library.

Next Up – Program our Ball