CI to the Danger Zone

Shipping ~~Fixes~~ Updates Fast with Flutter & CI

Maksim Lin
Developer Relations Engineer
Codemagic

A bit about me...


Long time Android developer now doing Flutter

Developer Relations Engineer @ Codemagic

Flutter & Dart GDE,
Melbourne GDG co-organiser

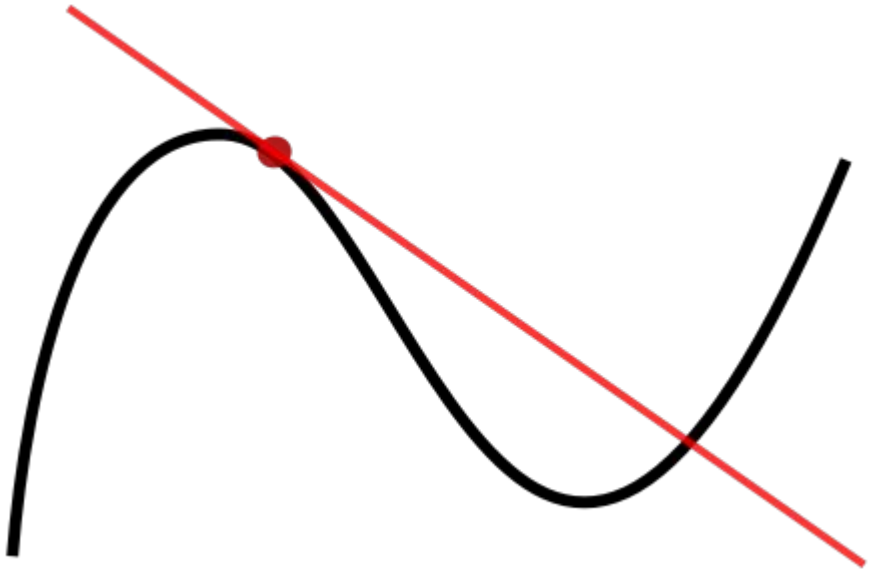

codemagic


ci_dangerzone_app
CI to the Danger Zone
DEBUG
Start Game

Flame Engine!

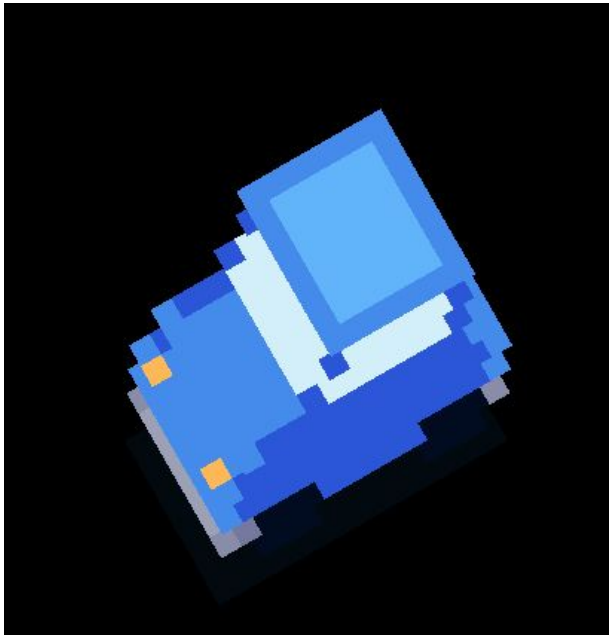
- *Flame is:* a **2D** game engine for Flutter
- *Easy to use:* a set of Dart packages
- Integrates closely with Flutter: all widgets

A little tangent...



Sprite Stacking!

- Sprite stacking is a technique of “fake 3D pixel art”
- Uses 2d painting of sprite “slices”
- Used for the car/motorbike player elements
- For more details see my upcoming blog post...



Game Logic

- The “business rules” of the game
- Behaviours (speed, acceleration, etc)
- Obstacles: how big, how often, etc
- Path finding
- Enemy AI
- etc

Game Logic in Dart

```dart
@override
Future<void> onLoad() async {
  sprite = await Sprite.load('racetrack.png');
  _spawnObstacles();
}

Future<void> _spawnObstacles() async {
  await add(WitchesHat(position: Vector2(350, 365)));
  await add(WitchesHat(position: Vector2(600, 465)));
  await add(WitchesHat(position: Vector2(200, 150)));
  await add(WitchesHat(position: Vector2(600, 70)));
}
```

Shipping Updates!

- Bugs Happen!
- Improve game play/balance
- New Game Levels
- Seasonal Variations

  (Characters, Levels)

Shipping Updates: App Updates 😞

Dart Code changes means...

- Build new app distribution artifacts
- Users need to be notified of new version
- Users need to download new version (except... on web)

Shipping Updates: App Updates

Lots of work!

Scripting Game Logic

- Many! games use scripting languages to provide parts of the game logic.
- Scripts can be used internally only (eg. “level design”)
- Scripts can allow for end-user customisation

ref:
https://en.wikipedia.org/wiki/Category:Lua_(programming_language)-scripted_video_games

Scripting Game Logic

- Scripting also useful outside of shipping code updates
- Don't just take my word for it:
- *"...they did alot of lua scripting ...allowed the designer to experiment alot...without bothering the programmers all the time"*
- Conference Talk

  Rovio: Angry Birds: Behind The Scenes (Flash GAMM Kyiv 2012)

Lua

- *Lua is:* a scripting language/VM
- *Lua is:* very popular for embedding in games
- *Easy to embed:* Simple **Stack-based** embedding API
- LuaDardo is a **pure** Dart package implementing Lua 5.3

Loading and running Lua in Flutter

```
void createObstacles() {
  state.loadString(script);
  // eval the Lua chunk before we can call functions in it
  state.call(0, 0);

  state.register("mkObstacle", _makeObstacle);

  const funcName = "doObstacles";
  // now get the Lua script function
  final t = state.getGlobal(funcName);

  // check we found the Lua function we expected
  if (t != LuaType.luaFunction) {
    print("type err, expected a function but got [$t] ${state.toStr(-1)}");
    return;
  }

  // and now run the Lua function
  final r = state.pCall(0, 0, 1);
  if (r != ThreadStatus.lua_ok) {
    print("Lua error calling $funcName: ${state.toStr(-1)}");
    return;
  }
}
```

Game Logic in Lua

```
function doObstacles()
  -- mkObstacle is defined by the host Dart env
  mkObstacle(350, 365, visible());
  mkObstacle(600, 465, visible());
  mkObstacle(200, 150, visible());
  mkObstacle(600, 70, visible());
end

function visible()
  return randomBool(0.6)
end
```

Lua calling into Dart (Stack-based API)

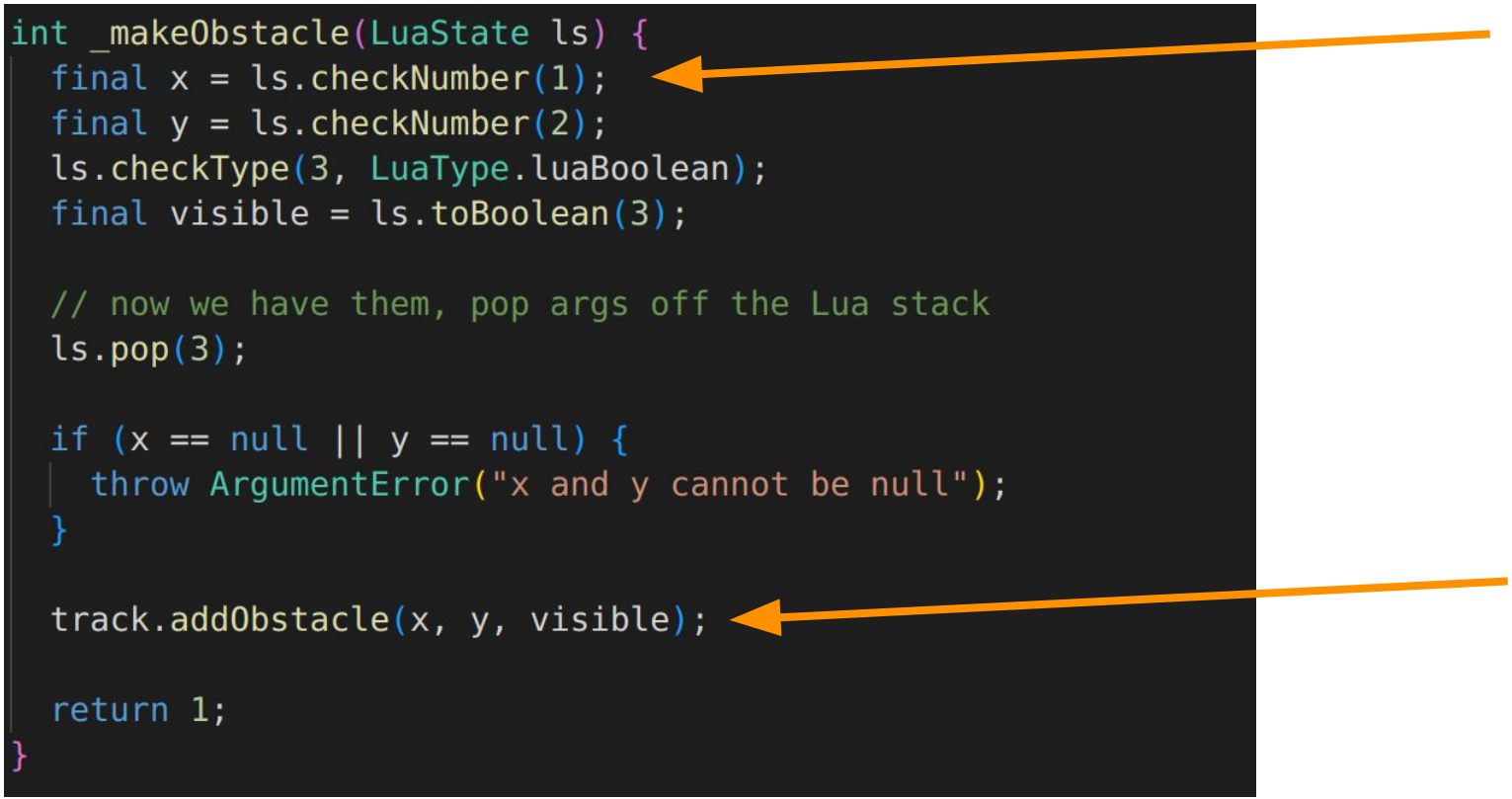
```dart
int _makeObstacle(LuaState ls) {
  final x = ls.checkNumber(1);
  final y = ls.checkNumber(2);
  ls.checkType(3, LuaType.luaBoolean);
  final visible = ls.toBoolean(3);

  // now we have them, pop args off the Lua stack
  ls.pop(3);

  if (x == null || y == null) {
    throw ArgumentError("x and y cannot be null");
  }

  track.addObstacle(x, y, visible);

  return 1;
}
```



Loading scripts from the network

```
Future<String> loadScriptFromNetwork() async {
  const scriptUrl = "http://localhost:8080";
  late final String script;

  final response = await http.get(Uri.parse("$scriptUrl/obstacles.lua"));
  script = response.body;
  return script;
}
```

App Updates without rebuilding!


ONE DOES NOT SIMPLY
EXECUTE UNTRUSTED CODE
imgflip.com

Code Signing*

(PUBLIC KEY CRYPTOGRAPHY)

Code Signing DIY: Dart Scripting!

- Use Dart package: *crypton*
- Generate Public + Private KeyPair:

```
main  ~/work/codemagic/ci_dangerzone_app $ tools/generate_keys.dart
generated key pair files
```

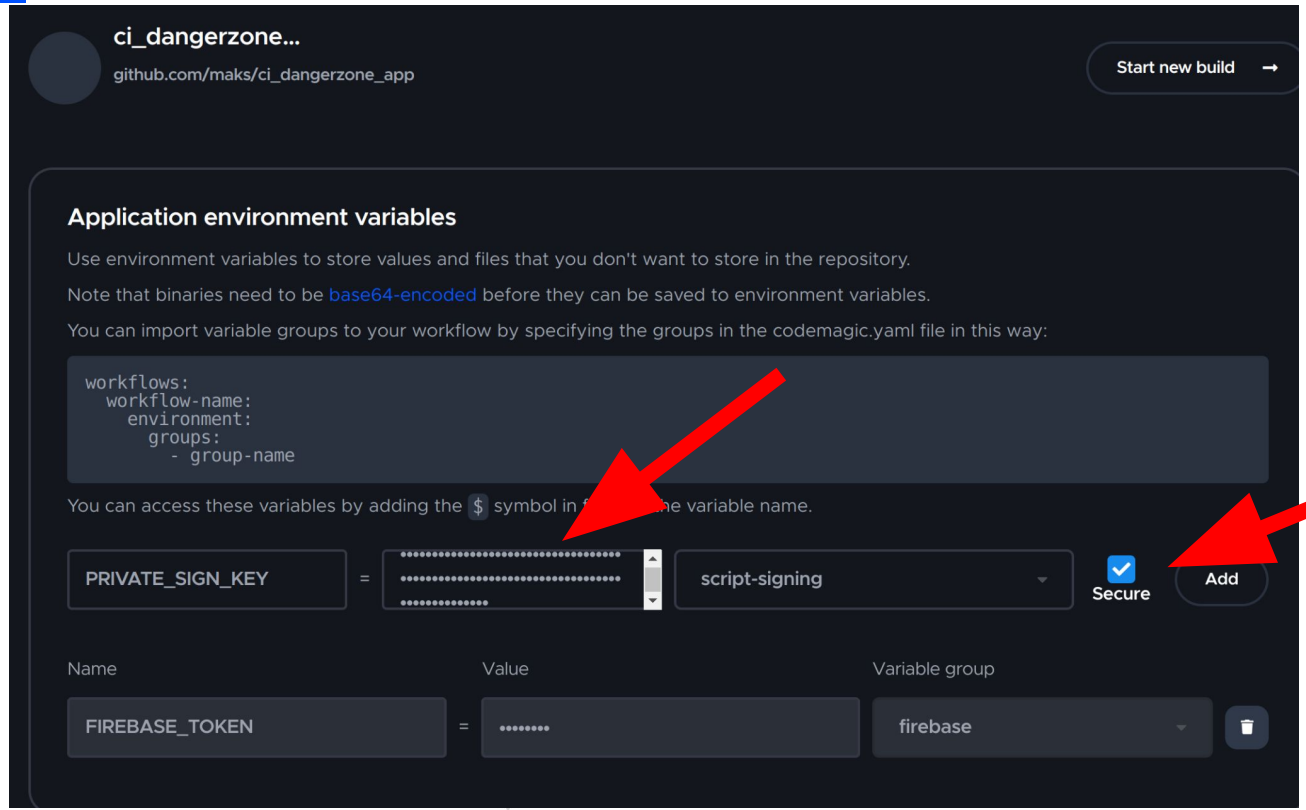
- Simple Dart CLI tool script to sign:

```
main  ~/work/codemagic/ci_dangerzone_app $ tools/sign_script.dart assets/scripts/obstacles.lua
script: assets/scripts/obstacles.lua
```

Code Signing DIY: On our CI

- Store Private Key as a **Secure (Encrypted!)** Environment Variable in Codemagic
- In Codemagic: Sign Lua script in CI, ship signed script + signature
- Upload both the signature file and Lua script file to static site hosting
- Public Key is fine to embedded as asset in our app

Code Signing DIY: Secure Env Vars in Codemagic



Upload from our CI

- Upload both the *signature* and *Lua script* to **Firebase Hosting**
  see:
  https://docs.codemagic.io/yaml-publishing/firebase-hosting/

Firebase

Code Signing DIY: On our CI

The Codemagic YAML:

```
    # build web
    flutter build web --release
    cd build/web
    7z a -r ../web.zip ./*
- |
    # sign lua script file
    tools/sign_script.dart game_scripts/obstacles.lua
    cp game_scripts/obstacles.lua build/web/
    cp signature build/web/
    name: Publish to Firebase Hosting
    script: |
      firebase deploy --token "$FIREBASE_TOKEN"
```

Code Signing DIY: On the Client

- Use same Dart *crypton* package in Flutter to verify the downloaded script content + signature 🚀

```dart
final response2 = await http.get(Uri.parse("$scriptUrl/signature"));
final signatureString = response2.body;

final publicKeyString = await rootBundle.loadString("assets/public_key");
final publicKey = RSAPublicKey.fromString(publicKeyString);
final signatureFromFile = base64.decode(signatureString);
final verified = publicKey.verifySHA256Signature(
  utf8.encode(script) as Uint8List,
  signatureFromFile,
);
```

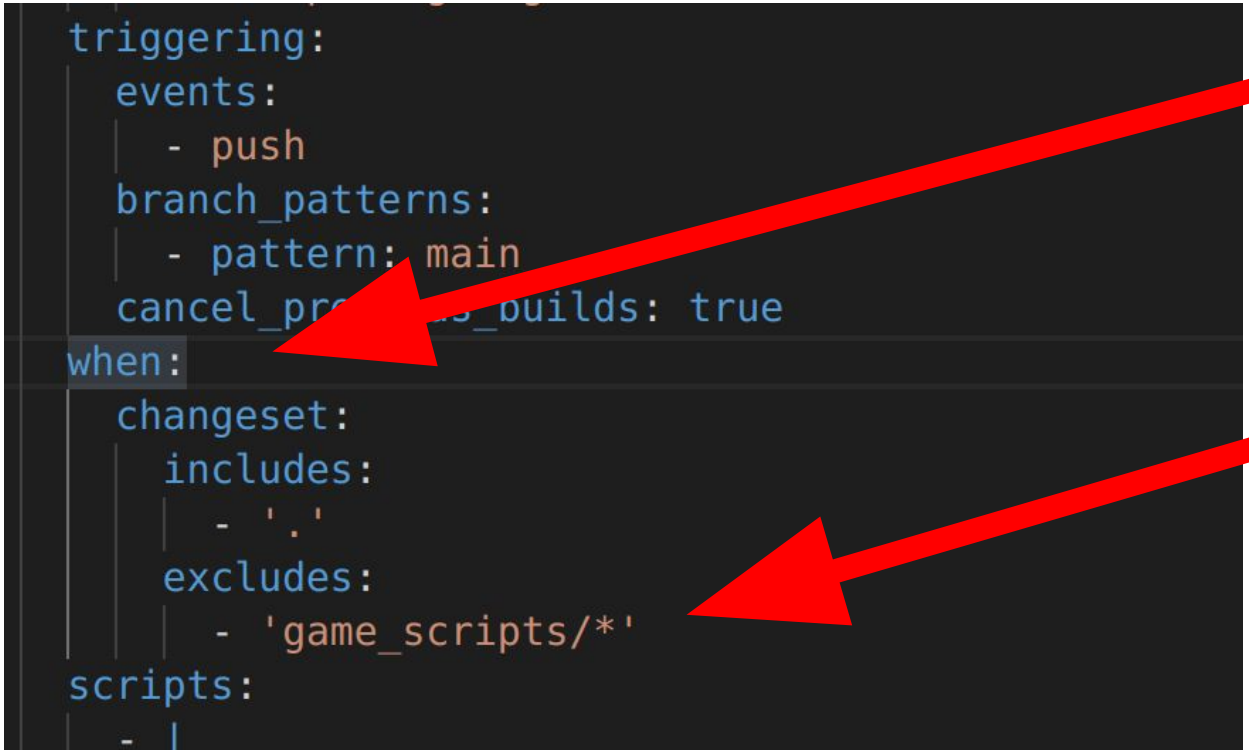
Code Signing DIY: On the CI

Just **one** more thing...

Code Signing DIY: On the CI

- No need to rebuild your app if only your script changes!
- But your CI deployment (the **CD** part!) workflow will normally trigger on every push to a branch (eg. *`main`*)
- Codemagic can help...
- Use ***conditional build triggers***...

Changeset filtering on triggers



Not just game logic!

- This technique could of course be used for many use-cases *outside* of game development
  Examples:
- More complex app configuration than supported by Firebase Remote Config
- A/B testing
- Promotional UI (sales, new products, etc) within apps
- Which leads us to...

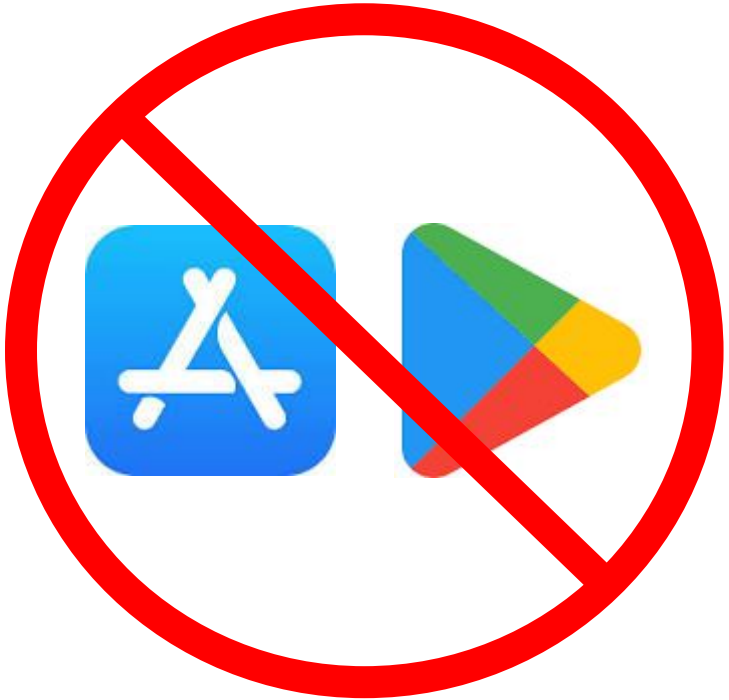
What about Widgets?

RFW, FTW!

- The **R**emote **F**lutter **W**idgets package
- From Hixie (Ian Hickson) Google Flutter Tech Lead
- RFW allows defining your UI via UI descriptions that can be obtained (**downloaded!**) at runtime.
- More details see: https://pub.dev/packages/rfw

Danger! Will Robinson, *Danger!*

Apple App* store and Google Play policies may *not* allow code push to apps!



[Alt stores for games: Steam, Itch.io, GoG, etc]

* eg. https://developer.apple.com/forums/thread/69039

Scripting: Putting it all together

- Fix bugs quickly: no waiting for app deploys to users
- Can be used for fixing UI bugs or smaller UI updates (**if** allowed by distribution platform)
- Not just for prod: internal testers, stakeholders
- Not just for prod: game/level design vs app dev
- Scripting: allows end-user scripting which is **very** popular, eg. Minecraft, Roblox
- Not just for games: useful for regular Flutter apps

Credits

- @EduSilvArt: Car, Motorbike Sprite assets from Itch.io
- @somepx: FutilePro font
- HeartBeast@YouTube: GameMaker Studio 2 - 3D Racecar - Sprite Stacking
- @wolfenrain (aka Jochum van der Ploeg) for advice on sprite stacking in Flame
- @spydon (aka Lukas Klingsbo) for advice on Flame widgets in overlays

Example game git repo

https://github.com/maks/ci_to_dangerzone

Thank You!
Questions?

fluttercommunity.social/@maks

@mklin

maks