

8種類の対応デバイスを持つ
Unity製SaaSのUnityバージョンを
アップデートしたときに行った
回帰テストの自動化手法

STYLY株式会社 QAエンジニア

井之上 心也

情報系の工業高校を卒業後、大手メーカーのソフトウェアおよびファームウェア品質保証部門に就職。

日々のテスト業務の傍ら、プログラミングの知見を活かしてGUIのテスト自動化システムの構築などを行う。

その後、「STYLY」に一人目QAとして転職。





STYLY

STYLYとは？



XRの制作配信プラットフォーム

ARやVRの作品をWeb/Unityを介して制作し

VRゴーグル、スマートフォン、Webサイトで閲覧できる

Unityをベースに開発している

UnityでできることはほとんどSTYLYでも動きます

C#スクリプトは動かないが、代わりにノードベースでプログラムを組めるライブラリに対応

ローンチしてから6年

Unityアップデートは3回目だが、自分がjoinしてからは初

XRとは？



XRとは？

XR

MR
(複合現実)



AR
(拡張現実)



VR
(仮想現実)



XR
作品制作



Webブラウザ
Unityからの
アップロード



XR
作品配信



VRゴーグル
スマホ
Webサイト



STYLY のUnityバージョンは2019 LTS

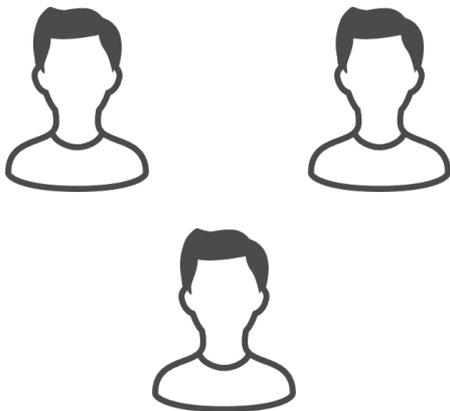
Unity 2019 LTS は2022年にサポートが切れ た

バージョンアップしないと
新しいライブラリを入れられない！

STYLYの開発体制



Unityエンジニア
3人



PdM 1人



EM 1人



※バックエンドと兼任

フロントエンド
1人



QA 1人



STYLY のサポートデバイス



役割	プロダクト識別名称	デバイス名
XR作品閲覧	STYLY VR	Meta Quest2/3
		Pico 4/4 E
	STYLY Mobile Viewer	Android
		iOS
SATCH X	Android	
	iOS	
	STYLY WebClient	Chrome Safari FireFox
XR作品制作	STYLY Studio	Chrome Safari FireFox

リリースするときに作成しないといけないビルドの数は 8個

STYLYにアップロードされている作品は
11万作品に及ぶ

社内の要望で後方互換性はなるべく
保ちたい

他チームが平行して制作している 案件も多数存在



2024-08-09

にじさんじ 夜見れな 3D新衣装披露ライブ配信「6つの星に願いをプロジェクト」



2024-05-19

AIR RACE X 2024

社内からの要望で 案件の進行に影響を出したくない

状況を整理しましょう

現状整理

開発体制	Unityエンジニア3名 QA1名
アップデート対象	8ビルド
テスト範囲	ベースとなるミドルウェアの アップデートなので 全ての機能が対象
後方互換性を保ちたい 作品数	11万作品が存在 ※過去2年間に作られた作品は保ちたい
そのほか考慮事項	平行で案件/イベントが存在 ※社内からの要望で極力影響を出したくない

限られた工数の中で
後方互換性を保ちながら
全機能のリグレッションテストを行う
必要がある

無理です

ソフトウェアテストの原則より

2. 全数テストは不可能

すべてをテストすることは、ごく単純なソフトウェア以外では非現実的である。全数テストの代わりに、テスト技法、テストケースの優先順位付け、リスクベースドテストを用いて、テストにかかる労力を集中すべきである。

条件を絞りました

条件を絞る

開発体制	Unityエンジニア3名 QA1名
アップデート対象	8ビルド
テスト範囲	ベースとなるミドルウェアの アップデートなので 全ての機能が対象
後方互換性を保ちたい 作品数	11万作品が存在 ※過去2年間に作られた作品は保ちたい
そのほか考慮事項	平行で案件/イベントが存在 ※社内からの要望で極力影響を出したくない

条件を絞る

開発体制	Unityエンジニア3名 QA1名
アップデート対象	8ビルド ⇒ 1stリリースは4ビルドに (Mobile Android/iOS、WebClient、Studio)
テスト範囲	後方互換性のテストに関しては Webサイトでフィーチャーしている XR作品160件を回帰テストする
後方互換性を保ちたい 作品数	基本的なSTYLY機能のみ全デバイスで確認
そのほか考慮事項	平行で案件/イベントが存在 ※社内からの要望で極力影響を出したくない

どのように回帰テストを行うのか？

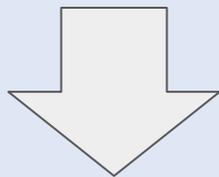
Unity 2019 LTS ⇒ Unity 2022 LTS

Unityの3年分の変更点が一気に押し寄せる！



アップデートした影響による過去のXR作品への影響が不明

アップデートした影響による過去のXR作品への影響が不明



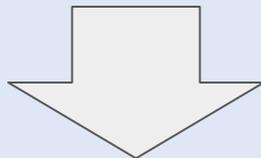
繰り返し実行できて
変化をわかりやすくトラッキングできるようにしたい

二つの自動回帰テストを導入しました

1. モバイルアプリのGUI自動テスト
 - a. アプリとしての機能を担保する
2. モバイルアプリ/WebClientでのXR作品自動閲覧テスト
 - a. XR作品への影響を継続的にトラッキング

1. **モバイルアプリのGUI自動テスト**
 - a. **アプリとしての機能を担保する**
2. **モバイルアプリ/WebClientでのXR作品自動閲覧テスト**
 - a. **XR作品への影響を継続的にトラッキング**

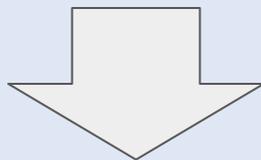
モバイルアプリが一番機能が多彩であり
弊社としても一番案件での使用頻度が高い



XR作品閲覧に関わらないGUI部分(ログイン/ユーザー作成など)を
回帰テストとしてCI上で行おう！

モバイルアプリのGUI自動テスト

Unityアプリなのでネイティブアプリと違って
GUIの描画がDOMなどで表現されない
画面いっぱいのCanvasしかない



Seleniumやそれを使った自動テストフレームワークは使えない

自動テストやるなら...画像認識しかないのか...

**画像認識はレンダリングの関係で
うまく認識されない場合が多い...**

Airtest + pocoならいける！

Airtestとは？



NetEase社より提供されているテストツール
Android、iOS、Webなど、様々なプラットフォームに対応
使用するスクリプトの言語はpython

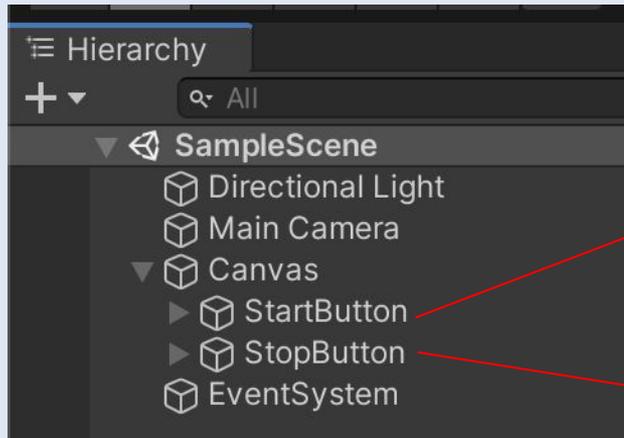
Pocoとは？



Airtestと共に提供されている

ゲーム開発プラットフォームでのUI周りのテストを記述するためのフレームワーク
Unity、Unreal Engineなどにも対応している

UnityのGameObjectの名前を指定することでUI制御ができる！



StartButtonを押したいときは
`poco("StartButton").click()`

StopButtonを押したいときは
`poco("StopButton").click()`

オブジェクト名指定

```
poco("StartButton").click()
```

任意の文字が表示されているオブジェクトを指定

```
poco(text="Start!!").click()
```

任意のテクスチャ名のオブジェクトを指定

```
poco(texture="icon_start_button").click()
```

オブジェクト名が被っていてもUI部品を特定できる

構築したフロー

1. PRが立ったタイミングでそのPRの自動ビルドを行う
2. 1で作ったビルドでGUIの回帰テストを回す
3. テスト結果がNGだった場合、PRのコメントに残す。

モバイルアプリのGUI自動テスト

フロー①開発者がPRを立てる



STYLYVWR-818 iOS16以上でARシーンを横画面で動画撮影した際に歪む不具合への対応 #3577

Merged hacha merged 6 commits into develop from ozawa/STYLYVWR-818_FixIOSMovieCaptureBug on Oct 3

Conversation 12 Commits 6 Checks 20 Files changed 6 +93 -70

hacha commented on Sep 29

下記の条件を満たす場合に動画撮影が歪む不具合があります。

- iOS16以上
- ARシーン
- シーン内メニューの画面方向が自動

厳密な原因は不明ですが、どうもiOS16にバグがあるらしいです。
<https://developer.apple.com/forums/thread/710354>
これに起因してiOSではUnityEngine.Screen.orientationが正しい値を返さないことがあるバグがあります。
<https://issuetracker.unity3d.com/issues/ios-screen-dot-orientation-returns-wrong-values-when-switching-from-landscape-orientation-to-autorotation-at-runtime-on-ios-16>
<https://blog.kyubuns.dev/entry/2023/05/06/135242>

色々試したところ、撮影時の画面回転ロックで明示的にScreen.orientationを指定することでLandscapeでの動画撮影を正常に行えるようになりました。
但し副作用として撮影時に画面回転をロックしているはずなのに、たまに画面回転してしまう現象が発生しています。

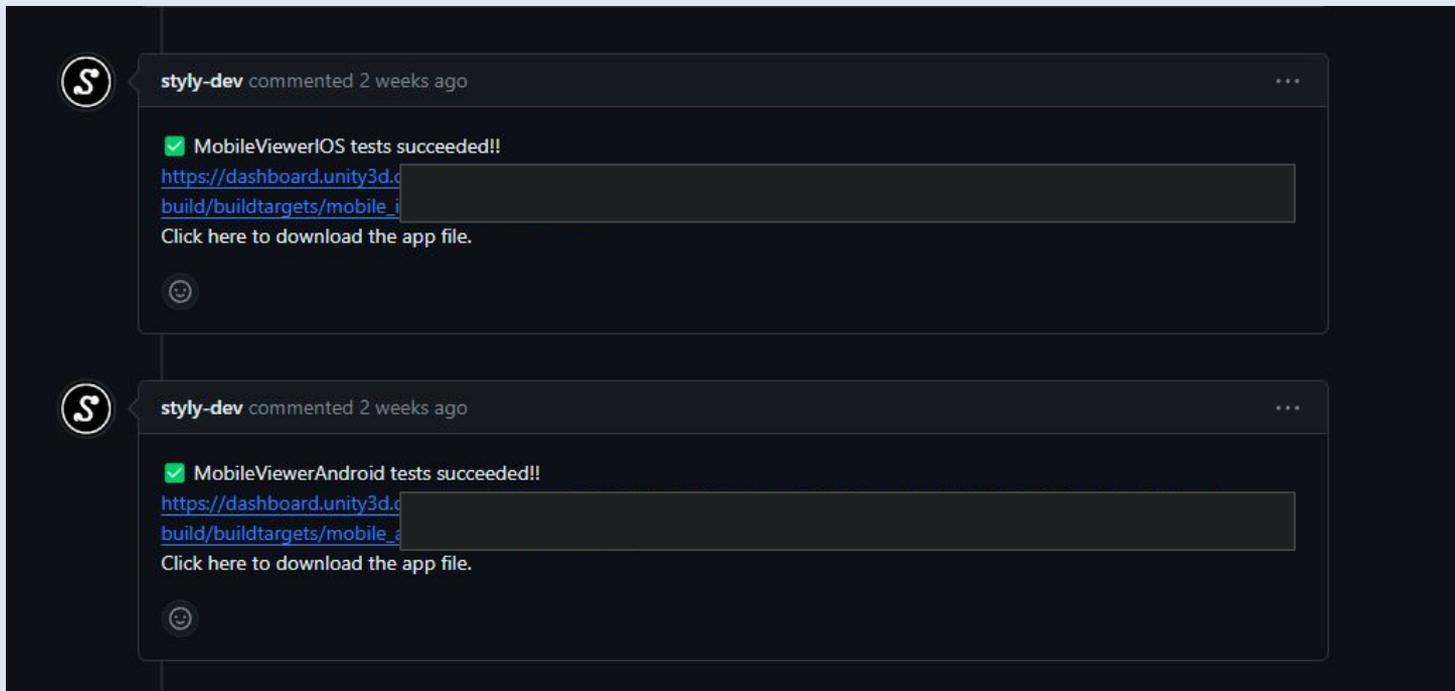
[関連スレッド](#)

Buildをdeploygateにアップしてあります。
iOS [#195](#)
Android [#160](#)

Reviewers: mhama
Assignees: No one
Labels: None yet
Projects: None yet
Milestone: No milestone
Development: Successfully merging this pull request may close these issues.
None yet

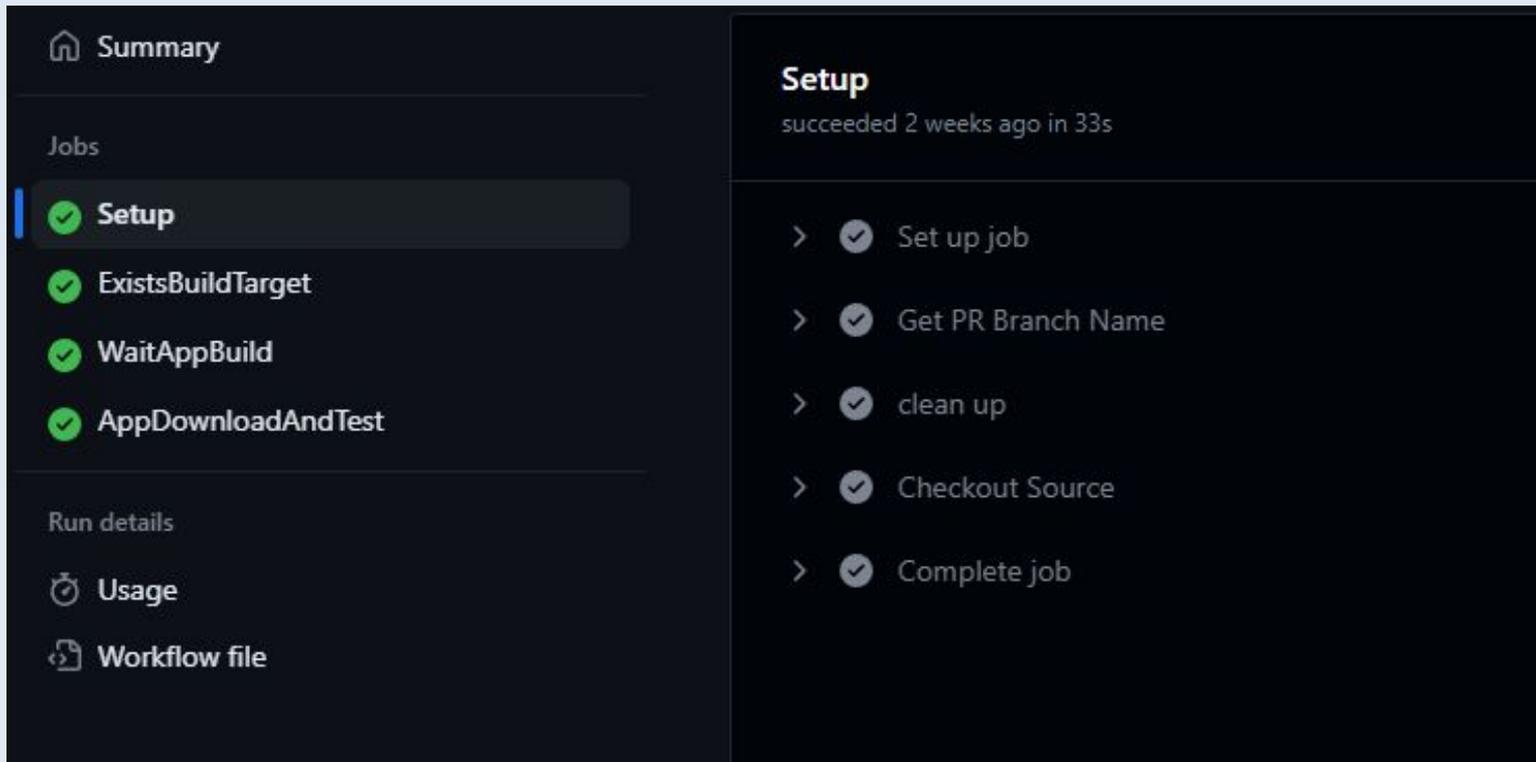
モバイルアプリのGUI自動テスト

フロー② 自動的にテスト用ビルドを作成



モバイルアプリのGUI自動テスト

フロー③ テスト用ビルドを用いて自動GUIテストを動かす



The screenshot displays a dark-themed interface for a CI/CD pipeline. On the left, a sidebar under the heading 'Summary' contains a 'Jobs' section with four items, each marked with a green checkmark: 'Setup', 'ExistsBuildTarget', 'WaitAppBuild', and 'AppDownloadAndTest'. Below this is a 'Run details' section with three items: 'Usage', 'Workflow file', and 'Workflow file'. The main content area on the right is titled 'Setup' and shows the job 'succeeded 2 weeks ago in 33s'. A list of steps follows, each with a chevron and a checkmark: 'Set up job', 'Get PR Branch Name', 'clean up', 'Checkout Source', and 'Complete job'.

Summary

Jobs

- ✓ Setup
- ✓ ExistsBuildTarget
- ✓ WaitAppBuild
- ✓ AppDownloadAndTest

Run details

- Usage
- Workflow file
- Workflow file

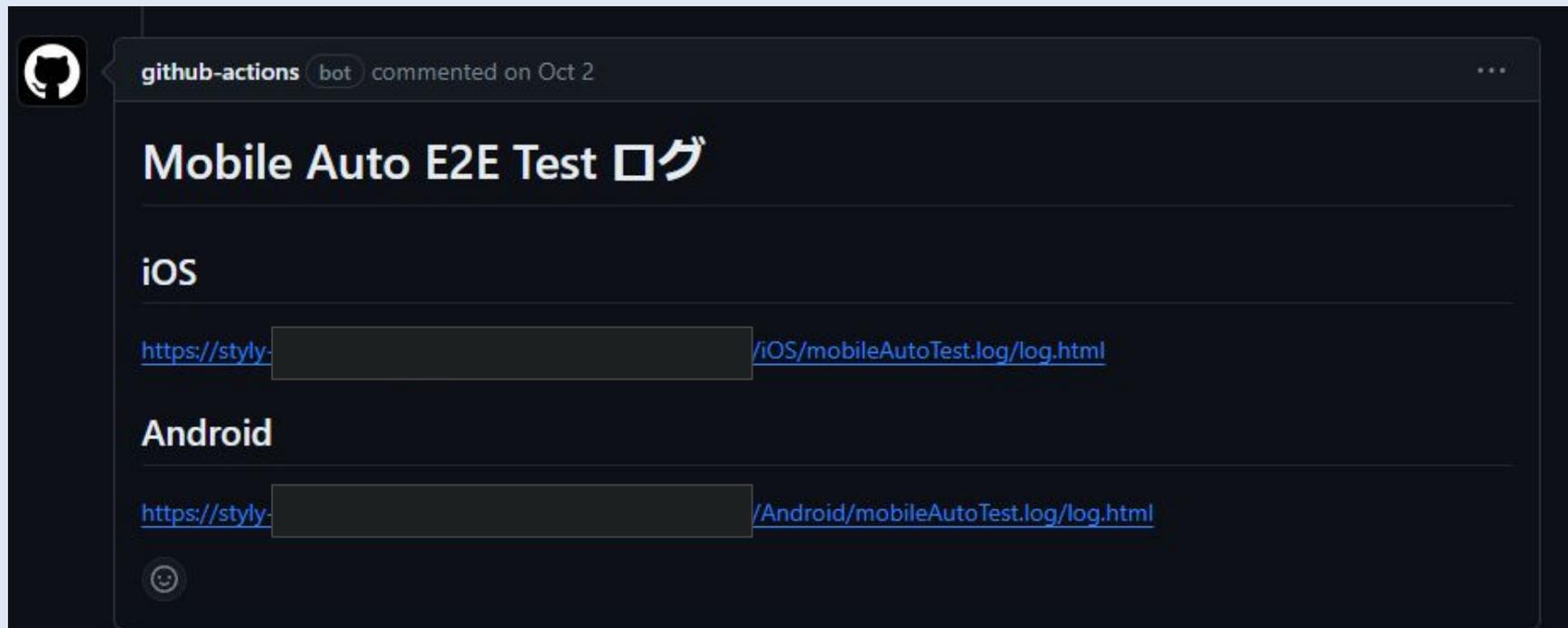
Setup

succeeded 2 weeks ago in 33s

- > ✓ Set up job
- > ✓ Get PR Branch Name
- > ✓ clean up
- > ✓ Checkout Source
- > ✓ Complete job

モバイルアプリのGUI自動テスト

フロー④ 失敗した場合、テストレポートをコメントする



モバイルアプリのGUI自動テスト

システム概要



※Cloud Build

Unityで作られたアプリをビルドするための環境
Unity公式が提供している

※self-hosted runner

Github Actionsの実行をクラウドではなく
自分が用意した環境で実行できる仕組みのこと

モバイルアプリのGUI自動テスト

システム概要



※Cloud Build

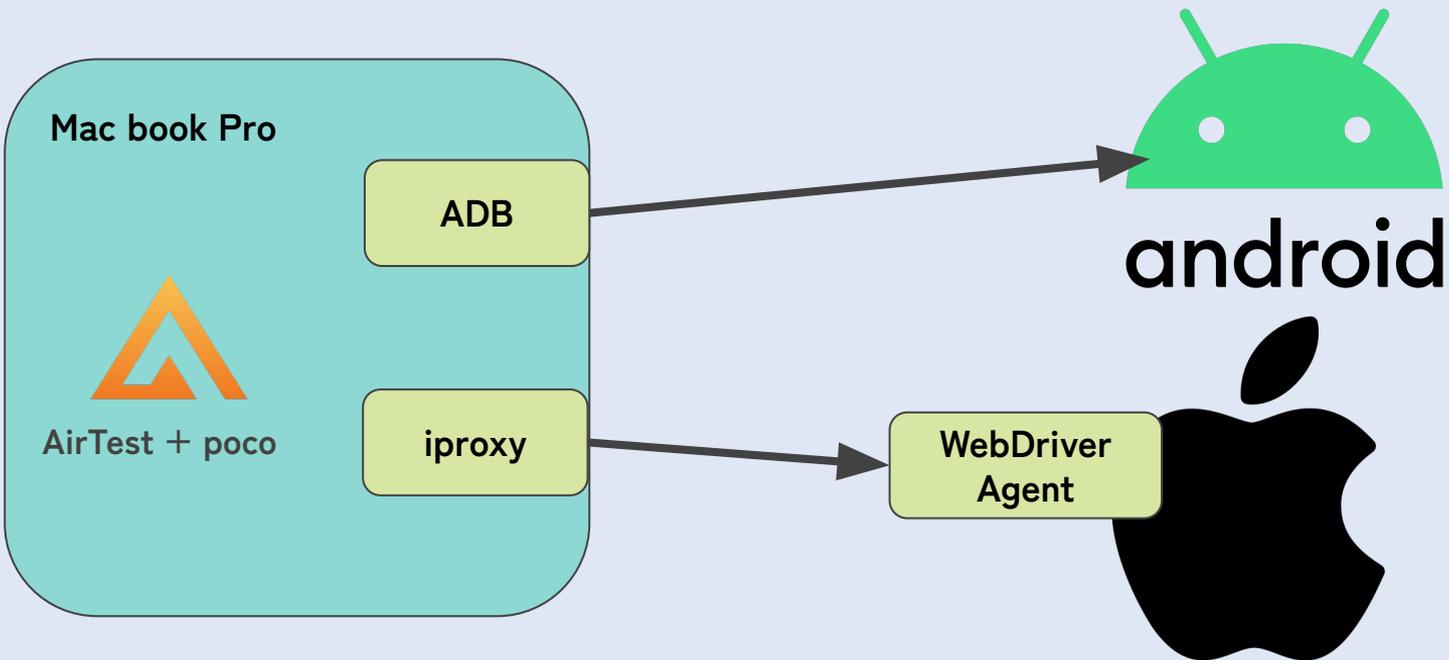
Unityで作られたアプリをビルドするための環境
Unity公式が提供している

※self-hosted runner

Github Actionsの実行をクラウドではなく
自分が用意した環境で実行できる仕組みのこと

モバイルアプリのGUI自動テスト

システム詳細



導入する自動回帰テスト

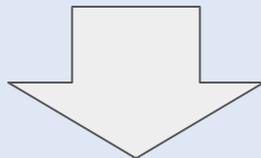
1. モバイルアプリのGUI自動テスト
 - a. アプリとしての機能を担保する
2. モバイルアプリ/WebClientでのXR作品自動閲覧テスト
 - a. XR作品への影響を継続的にトラッキング

導入する自動回帰テスト

1. モバイルアプリのGUI自動テスト
 - a. アプリとしての機能を担保する
2. モバイルアプリ / WebClientでのXR作品自動閲覧テスト
 - a. XR作品への影響を継続的にトラッキング

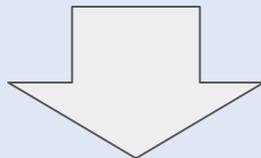
STYLYはUnityとの親和性が高い

ほとんどのUnityの機能をSTYLYで実現できる



テスト範囲があまりにも広すぎてテストしきれない！

STYLYのWebサイトでフィーチャーされている
作品を取り上げて、テストデータにしよう！



取り上げた作品たちで使われている表現手法や
機能が、STYLYで使われている機能として仮定する

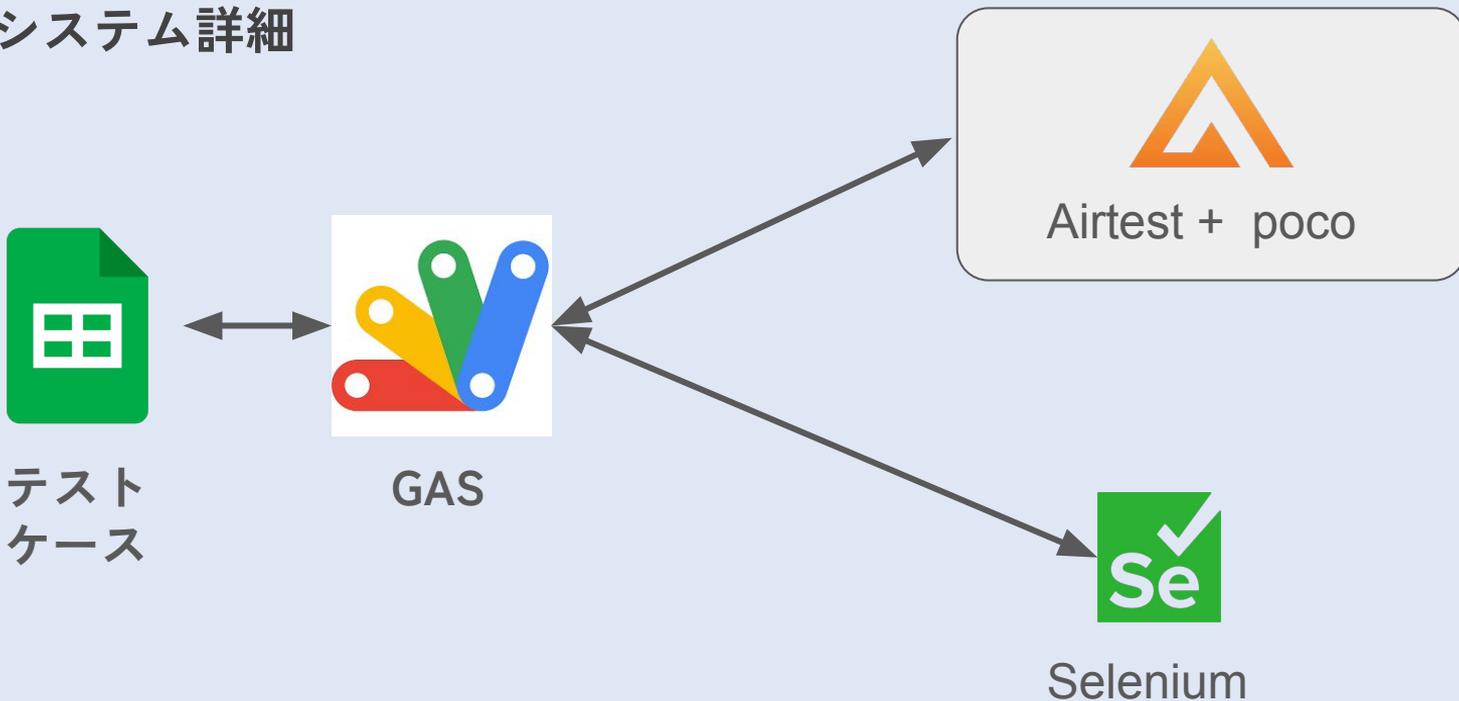
XR作品自動閲覧テスト(Mobile/Web)

システム概要

	操作に使用する フレームワーク	Assert
Mobile	Airtest + poco	アプリがクラッシュしないこと 操作不能状態にならないこと
WebClient	Selenium	クラッシュしないこと consoleにError文言が出ないこと

XR作品自動閲覧テスト(Mobile/Web)

システム詳細



WebClientはURLの変更で
任意のシーンを再生

XR作品自動閲覧テスト(Mobile/Web)



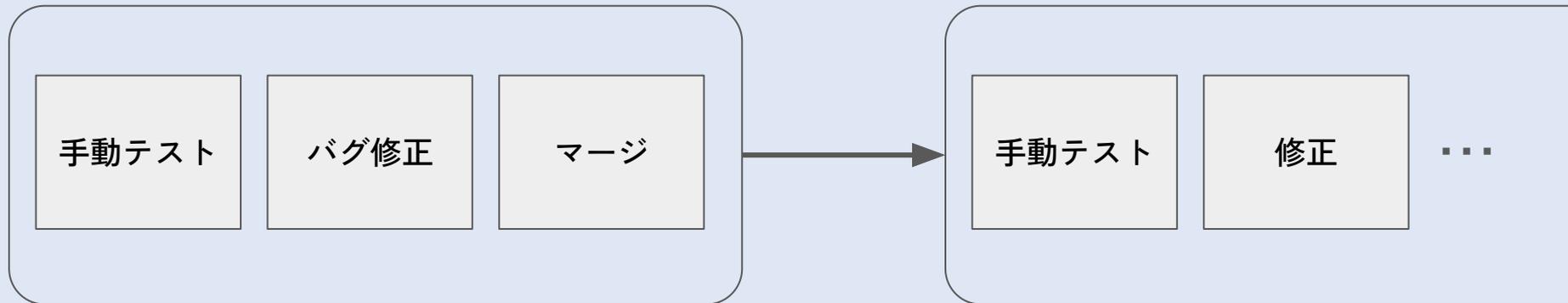
テストケース

B	C	D	E	F
		URL		自動テスト
1	6cdbbd2b-b585-45d8-93bc-5aa16063178c	https://gallery.sty	JACKSON kaki is dancing	OK
2	34d3d92e-0869-4b35-81b5-2da3266f061e	https://gallery.sty	CHIHAYABURU [size: 1m x 1m]	OK
3	a291acd8-4d4b-45fb-a7d4-ae4553386c42	https://gallery.sty	SUN AND ROOSTER	OK
4	a95439ff-bfa4-4d66-8bba-600ec1886f64	https://gallery.sty	BONDANCE_PARTY	OK
5	1bfdec06-1913-4b60-ba5e-4e6b4c32f259	https://gallery.sty	Nevermore	NG
6	dcdbf295-2eb3-496e-a098-4dbb9288cf56	https://gallery.sty	Christmas AI Date	NG
7	4cdc1a6b-687f-4f5f-97bc-f59b2c625779	https://gallery.sty	Chaos and Order : City AR for Shibuya scramble crossing	NG
1	a05b5d10-fbc3-40cb-b73a-dd8583c63344	https://gallery.sty	SHIBUYA, the city of MEGA RATs.	NG
9	b52c032c-3d80-41d1-bf85-d55812c49d5b	https://gallery.sty	AR WANTED in SHIBUYA X-crossing	NG
10	89479cf6-79cf-4223-80bd-d0722e389147	https://gallery.sty	読み込んでいます...	NG
11	2713efff-5b6d-43c3-be3e-36ffbc5220f	https://gallery.sty	2023_Ad machic TENGOKU	NG
12	7b7ce762-3ef1-4ac2-bc6b-6702c4635ca9	https://gallery.sty	Advent12/8	NG
13	2296c008-a8a8-4102-85b4-3eedc016ba50	https://gallery.sty	AWA-ODORI AR at Shibuya_Advent_Calender2023_1207_SHISHISHI	NG
14	3eda317f-e292-4b77-bc97-a60fca9674a2	https://gallery.sty	Spatial Owned Media_by STYLY_20231204	NG
15	1bb8cb21-96af-4084-9b1a-a76b883b678f	https://gallery.sty	Masahiro's Birthday_20230623	NG
16	a32f2bc7-40d7-4527-9731-ca13cce3d147	https://gallery.sty	SHIBUYA_Alien	NG
17	cb475cf3-2f51-491d-be49-0d302ae0c754	https://gallery.sty	Checking the positional displacement in Shibuya	NG
18	d456b769-0ce1-447d-bd17-ddae32d43438	https://gallery.sty	Birds of a Scape	OK
19	c0c11c95-276f-42ac-a6dc-dd58fdeb1c8d	https://gallery.sty	Nether World(updated)	OK
20	81480176-e2d4-11e9-b34d-4783bb2170d0	https://gallery.sty	REMINISCENCE	OK
21	2e341888-9b11-11e8-b34d-4783bb2170d0	https://gallery.sty	The Birth	OK
22	d1d91e20-2e78-4a73-9b73-d63d3fe14c0e	https://gallery.sty	Please get out of here! Corona (English subtitles)	OK
23	47890cc1-12f1-41e0-b34d-4783bb2170d0	https://gallery.sty	Dark Room	OK

XR作品はGUIDで管理されているのでテストしたいGUIDを羅列している

XR作品自動閲覧テスト(Mobile/Web)

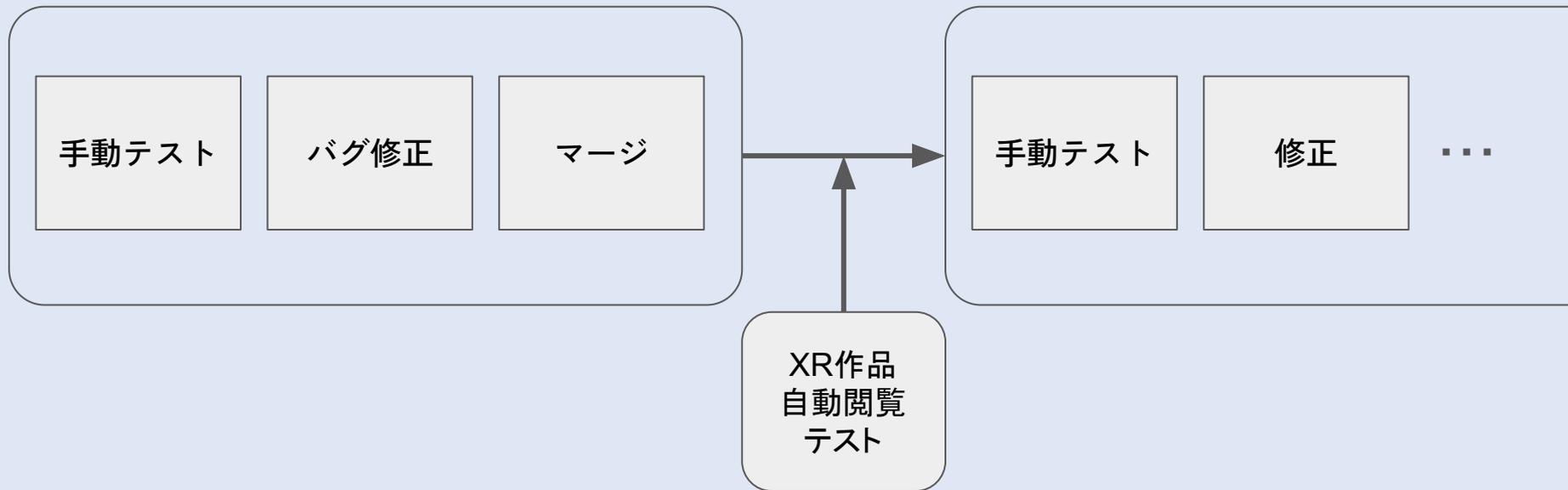
実行タイミング



テストの方針としては
バグ修正したら作業ブランチにマージして
それをまた手動テストして...を繰り返すようにしている

XR作品自動閲覧テスト(Mobile/Web)

実行タイミング



今回の修正によりどのぐらいのバグが直ったかを確認するため
マージ後に繰り返し実行していた

二つの自動回帰テストを導入しました

二つの自動テスト

	テスト対象	テスト範囲	トリガー	観点
GUI自動テスト	Mobile	ログインなどの基本的な機能	PRが作成される時に自動で	基本機能がデグレードしていないこと
XR作品自動閲覧テスト	Mobile WebClinet	XR作品を描画処理する機能	変更を入れた後に手動トリガー	XR作品再生時にクラッシュしないこと

二つの自動テスト

	テスト対象	テスト範囲	トリガー	観点
GUI自動テスト	Mobile	ログインなどの基本的な機能	モバイルアプリの根幹機能の品質を担保	
XR作品自動閲覧テスト	Mobile WebClinet	XR作品を描画処理する機能	変更を入れた後に 手動トリガー	XR作品再生時に クラッシュしないこと

二つの自動テスト

	テスト対象	テスト範囲	トリガー	観点
GUI自動テスト	Mobile	ログインなどの基本的な機能	モバイルアプリの根幹機能の品質を担保	
XR作品自動閲覧テスト	Mobile WebClinet	XR作品を描画処理する機能	クラッシュする重大な欠陥がどのぐらい修正されたかのトラッキング	

実施した効果

実施した効果



MobileアプリのGUI自動テスト

自動テストの項目数は 100項目

実施した効果

MobileアプリのGUI自動テスト

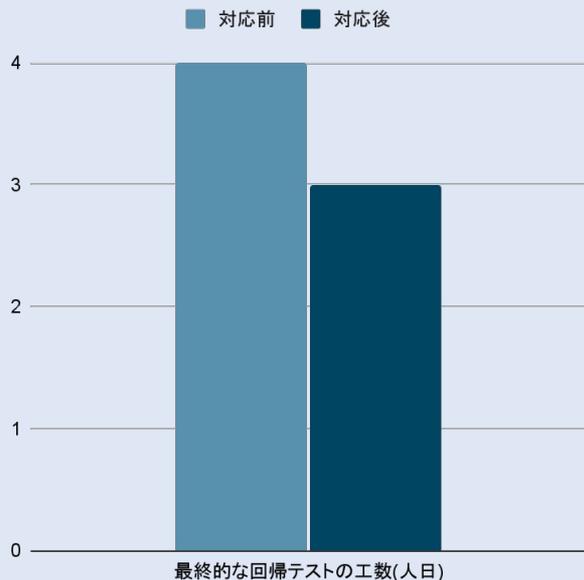
自動テストの項目数は 100項目

これはMobileアプリの回帰テストの70%を占めている

実施した効果

MobileアプリのGUI自動テスト

自動テストの項目数は100項目



リリース前の回帰テストの工数を**25%**削減！

実施した効果

XR作品自動閲覧テスト

テスト対象とした環境

Android 2機種

iOS 2機種

WindowsのChrome

MacのChrome

合計6環境で実施

実施した効果

XR作品自動閲覧テスト

削減できた工数

	一回に かかる工数	テスト 環境	回帰テストの 回数	
手動の場合 (予測)	2人日	x 6環境	x 7回実行	= 84人日
自動テスト	0.5人日	x 6環境	x 7回実行	= 21人日

XR作品の閲覧テストの工数を1/4に削減

当初予定していた
アップデート作業期間は
2年だった

1年6か月に短縮出来た！

まとめ

Unityエンジニア3名 QA1名で
テスト範囲全機能、影響範囲不明のアップデート

当初見積もりでは2年以上かかる予定だった
1年6か月でリリース出来た！

今後の展望

MobileアプリのGUI自動テスト

端末の状態によって、UIの応答が遅くなり
シナリオの途中で停止する現象が発生

XR作品自動閲覧テスト

本来はOKになるはずの項目がNGになってしまう誤判定が
頻繁に発生
→NG判定の部分の再度手動で確認する二度手間がかかっていた

MobileアプリのGUI自動テスト

端末の状態によって、UIの応答が遅くなり
シナリオの途中で停止する現象が発生

テスト実施時の安定性向上

XR作品自動閲覧テスト

本来はOKになるはずの項目がNGになってしまう誤判定が
頻繁に発生

→NG判定の部分の再度手動で確認する二度手間がかかっていた

テスト実施時の判定機能の強化

ご清聴ありがとうございました！