

Kyoto Tech Talk #4

# Playwrightを使いながら デザインや検索アルゴリズムのテスト


---

株式会社Helpfeel エンジニア

秦 友幸

@hata6502

秦 友幸 (はたともゆき)

- 株式会社Helpfeel エンジニア
  - インフラ基盤、顧客受け入れ
  - Cloudflare、EV SSL証明書、SAML認証、脆弱性対応、アクセシビリティ対応など
-  ムギユウ @hata6502
  - BlueHoodというお絵かきアプリを運営していました



情報からナレッジを作り、磨き上げ、届けるところまで、All-in-Oneで行うことができるSaaSプラットフォーム



知識を届ける  
エンタープライズサーチ



知識を磨き上げる  
アイディエーションツール



情報を知識にする  
メディアキャプチャー

# Playwrightを使ってデザインのテスト

---

- 既存システムの磨き上げもやっていきたい
  - デザイン
  - 検索アルゴリズム

Helpfeelについて調べたいことは何ですか？

gizyutu|



6,566件中 4件マッチ 0.042秒

- Helpfeelを支える技術
- チャットボットのように機械学習の技術が用いられていますか？
- OpenAI社の技術を使ってFAQを下書きする機能はありますか？
- 意図予測検索とはどのような技術ですか？

 Gyazo Support




Gyazoへ戻る  お問い合わせ  オートコンシェルジュ: [開始](#)

聞きたいことはありますか？

例: ログアウト



よくある質問

-  Gyazoの使い方
-  画像一覧ページ
-  ログインでお困りの方

## 目立たない想定の小さいデザイン修正

- フォーカスリングを付けた
- コントラスト担保のため色を変えた
  - 水色リンク ⇨ 青色リンク
- 驚かせないようにデザイン修正するには
  - デザイン崩れはないか?
  - 案件ごとのデザインカスタマイズへの影響は?
    - CSSレベルで自由に書いてるけど

## Playwrightを使ってみた

- ユーザー視点でのテストならE2Eテスト
- ヘッドレスブラウザ
- Visual Regression Test



- テスト差分を見ていくのが主な目的
  - テストは毎回落ちる前提で
- 公開された大量の実データをとにかく投入
  - ヘッドレスブラウザ × 大量データ
- CIには組み込まない
  - デザイン修正したいときだけテスト実行

- テスト結果 = デザインが変わる箇所
- 人間がデザイン変更差分を見ていく

聞きたいことはなんですか？

公開記事



6件中 1件マッチ

 [公開記事](#) タイトル

- テスト結果 = デザインが変わる箇所
- 人間がデザイン変更差分を見ていく

聞きたいことはなんですか？

公開記事



6件中 1件マッチ

 公開記事タイトル

- テスト結果 = デザインが変わる箇所
- 人間がデザイン変更差分を見ていく

聞きたいことはなんですか？

公開記事



6件中 1件マッチ

 [公開記事](#) タイトル

- テスト結果 = デザインが変わる箇所
- 人間がデザイン変更差分を見ていく

聞きたいことはなんですか？

公開記事



6件中 1件マッチ

 公開記事タイトル

- テスト結果 = デザインが変わる箇所
- 人間がデザイン変更差分を見ていく

聞きたいことはなんですか？


公開記事



6件中 1件マッチ

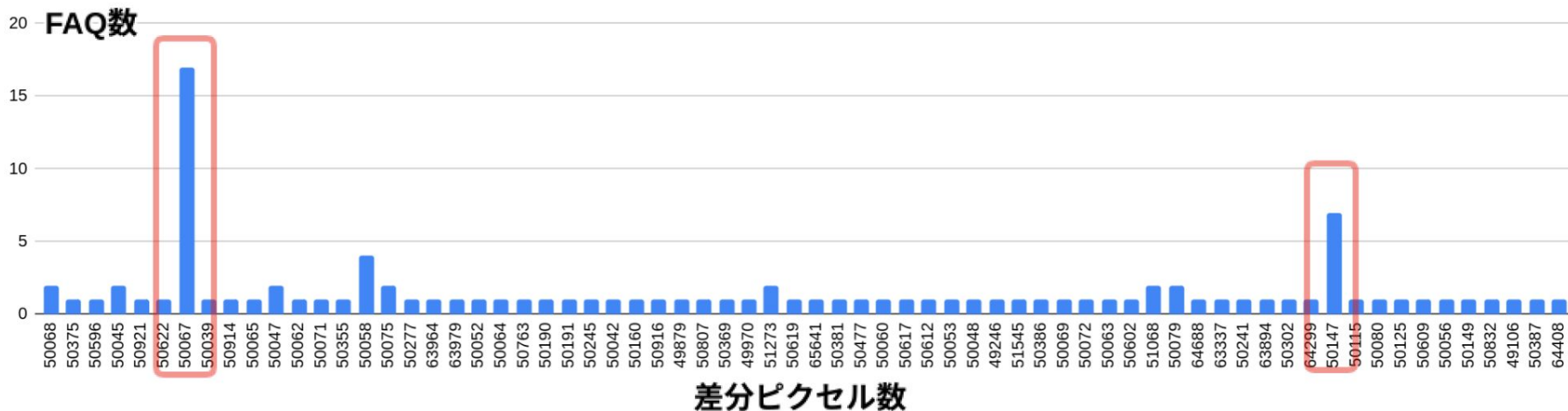
 **公開記事** タイトル

- 公開されているFAQ数百案件のCSS

- not 継続的テスト
-  がテスト結果を全部見た
- 数百案件のテスト結果でも、人間が見ていける
  - デザイン修正したいモチベーション
  - 差分ピクセル数で分類する



- FAQ案件ごとにCSSはカスタマイズされている
- テスト結果の差分ピクセル数で、FAQ案件进行分类する
- 差分ピクセル数が同じなら、デザインの変わり方も同じはず



## focus-visible状態でスクリーンショットを撮る

- CDP(Chrome Devtools Protocol)でCSS.forcePseudoStateする

```
// 全入力項目と送信ボタンにフォーカスを当てる。
const cdp = await page.context().newCDPSession(page);
const nodeIds = (
  await cdp.send("DOM.querySelectorAll", {
    nodeId: (await cdp.send("DOM.getDocument")).root.nodeId,
    selector: ":is(.form-control, .button-submit)",
  })
).nodeIds;
await cdp.send("CSS.enable");
for (const nodeId of nodeIds) {
  await cdp.send("CSS.forcePseudoState", {
    nodeId: nodeId,
    forcedPseudoClasses: ["focus-visible"],
  });
}
```


- report.jsonlから正規表現で採ってくる
- `/(\d+) pixels/`

{ } report.jsonl ×

home > hata6502 > helpfeel-visual-diff > { } report.jsonl

```
227   parentStepId": "922553d824af043ab22bbbdb8809e35f", "title": "attach > Screenshot compa Aa _ab_* ? / 792 件  ↑ ↓ ≡ × \
228   ation": 1}}}}
229   parentStepId": "922553d824af043ab22bbbdb8809e35f", "title": "attach \"Contact-Sense-visual-diff-██████████1-actual.png\"",
230   ation": 1}}}}
231   parentStepId": "922553d824af043ab22bbbdb8809e35f", "title": "attach \"Contact-Sense-visual-diff-██████████1-diff.png\"", "c
232   ation": 1}}}}
233   ation": 3517, "error": {"message": "\u001b[31mScreenshot comparison failed: \u001b[39m\n\n 50068 pixels (ratio 0.02 of all
234   title": "After Hooks", "category": "hook", "startTime": 1712116267112}}}}
235   parentStepId": "3e68e9f5d03860bdb9a2fd94252fae67", "title": "fixture: page", "category": "fixture", "startTime": 1712116267114
236   ation": 0}}}}
237   parentStepId": "3e68e9f5d03860bdb9a2fd94252fae67", "title": "fixture: context", "category": "fixture", "startTime": 1712116267
238   ation": 0}}}}

```

- 公開されているFAQのCSSを全件ダウンロード
- mainブランチでローカル環境のHelpfeelを起動
  - Playwrightで全CSSを当てながらスクショを撮っていく
- デザイン修正ブランチでHelpfeel再起動
  - Playwrightで全CSSを当てながらVisual Regression Test
  - テスト結果はエラーになる前提で
- テスト結果を差分ピクセル数で分類
-  がデザイン差分を全部見る

## デザインが崩れそうなFAQを事前に対策しておく

### 一例 borderの色がズレている

言葉のヒント

A-Z

か

は

送信

## デザインが崩れそうなFAQを事前に対策しておく

### 一例 borderの色を合わせる

言葉のヒント

A-Z

か

は

送信

## デザインが崩れそうなFAQを事前に対策しておく

### 一例 borderの色がズレている

言葉のヒント

A-Z

か

は

送信

## デザインが崩れそうなFAQを事前に対策しておく

### 一例 borderの色を合わせる

言葉のヒント

A-Z

か

は

送信



## デザイン修正していくフローが整った

- 既存顧客も巻き込んでデザイン修正
  - not 新規案件からデザイン改訂
- デザイン変更箇所を事前にテスト
- 影響箇所を見つけて事前に対策できるようになった
  - デザイン修正による驚きを最小に

# Playwrightを使って検索アルゴリズムのテスト

---

## ローマ字でも検索したい

- IME英語モードのまま日本語入力しても、検索できるようにした
- seihin ⇒ 製品

聞きたいことはなんですか？

seihin|



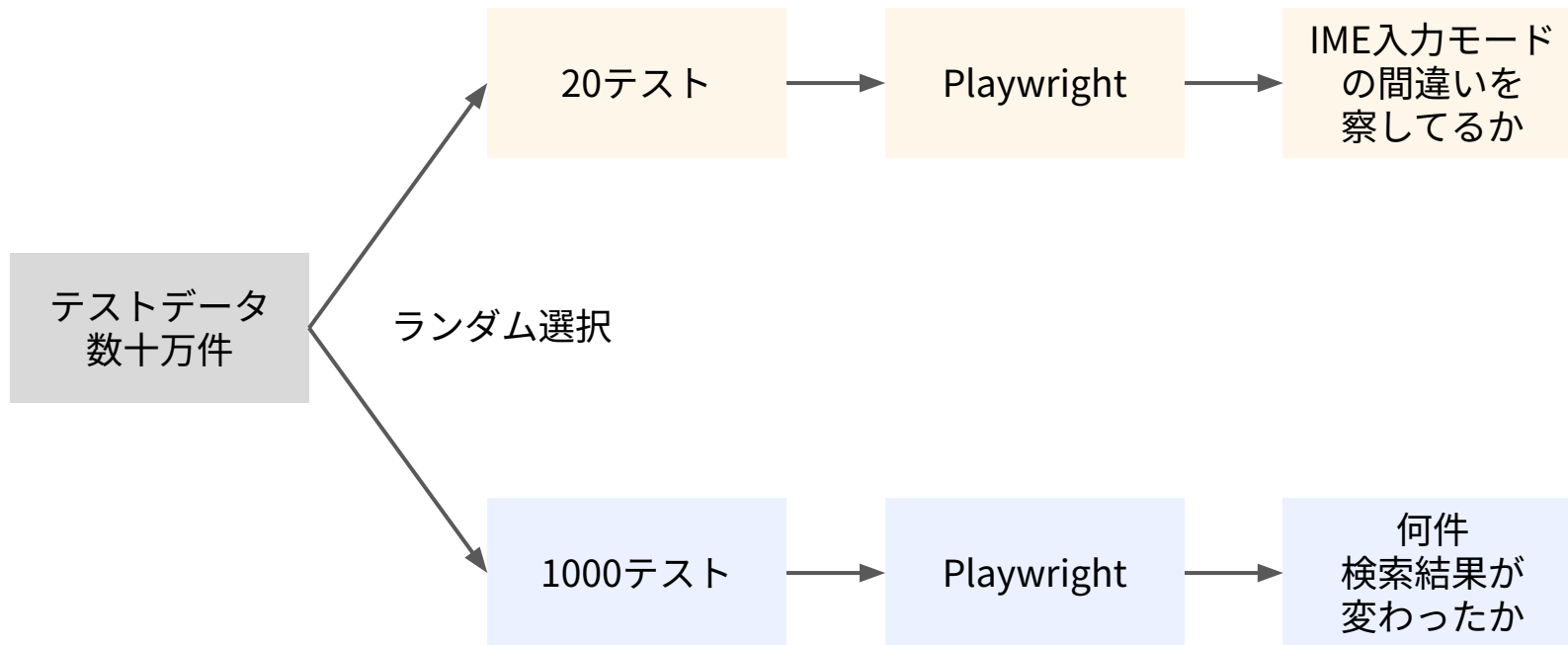
製品が壊れた

- まずは控えめに検索アルゴリズムを変更する
- 英単語に干渉するかも
  - 「IE」は、「家」か「Internet Explorer」か？
- 意図どおりの検索結果が増えるか？
  - IME入力モード間違いを察して、検索ヒットするのが目標
  - 関係ない記事が大量に検索ヒットしても無意味

## Playwrightを検索アルゴリズムのテストにも応用

	デザイン	検索アルゴリズム
テストデータ源	公開されているFAQ 数百案件のCSS	BigQueryに蓄積した 数十万件の検索クエリ※
テスト対象	全部テストする	ランダム選択
テスト結果の形式	スクリーンショット	検索結果のリスト
定性評価	デザイン崩れないか 全部見る	IME入力モードの間違いを 察してるか20テスト見る
定量評価	—	1000テスト中何テストで 検索結果が変わるか

※<https://www.helpfeel.com/landing-team>



## 重み付けランダムで頻出検索クエリを選択

- 数十万件の検索クエリ全部をテストするのは難しい
- 各検索クエリデータを、検索された回数で重み付け
- よく検索されるクエリほど、テスト対象に選ばれやすくする

- 実はテキストデータでも、スナップショットテストできる
  - スクリーンショットだけではない

index.spec.ts:75

5.5s

chromium

× Run

▼ Errors

Error: Snapshot comparison failed:

Q. rogui

A.

ログインできない

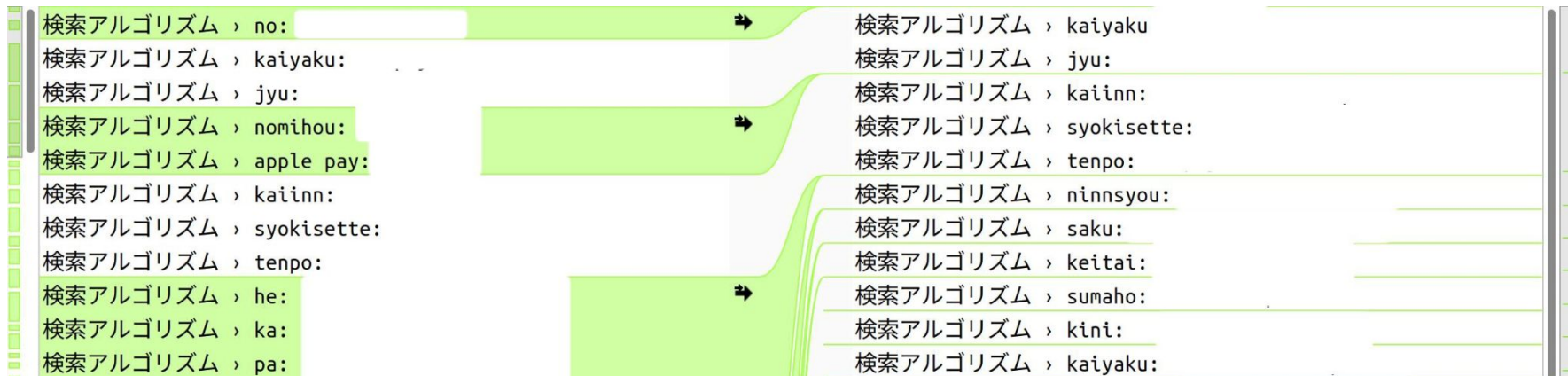
ログインID・パスワードを変更する

ログインできない



- 20件のテスト結果を、人間が見て評価する
  - 良くなった      11件
  - 変化なし      6件
  - 微妙だね      3件
- 実データを眺めると、その場で気づくこともある
  - 大文字の場合……
  - 子音が……
- 検索アルゴリズムを作り込んで、再度テストしていく

- 検索アルゴリズムを作り込みながら、影響度を調査
  - 1000テスト中何テストで、検索結果が変わるか
  - 10分~15分で機械的に結果を出すため、試行錯誤しやすい
- まずは控えめに、検索アルゴリズムを修正リリースした



## 事前に影響調査してから修正リリース


- 丁寧にプロダクト改善していけそうです
  - デザイン崩れが起きないかテスト
  - IME入力モードを間違えても検索できるかテスト

## Playwrightの使い方まとめ

	デザイン	検索アルゴリズム
テストデータ源	公開されているFAQ 数百案件のCSS	BigQueryに蓄積した 数十万件の検索クエリ※
テスト対象	全部テストする	ランダム選択
テスト結果の形式	スクリーンショット	検索結果のリスト
定性評価	デザイン崩れないか 全部見る	IME入力モードの間違いを 察してるか20テスト見る
定量評価	—	1000テスト中何テストで 検索結果が変わるか

※<https://www.helpfeel.com/landing-team>

## 個人開発祭り #3 やります

- 銀座でオフライン & 配信
- 2024年4月12日 19:00~21:30
  - 3日後、週末、金曜日です
-  も参加します
  - アルバムと地図の話をしてします
  - カレンダーの話も
- みてね

