うさちゃんのにんじんクリックハント🥕

手書き草案

1 ヒアリング

2 要件 → スパイラル型

機能要件

非機能要件

3 UML 設計図

作りたい物♡

〈楽しく学ぶクリック練習アプリ〉

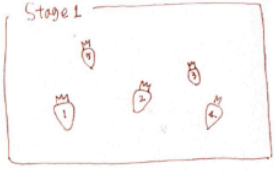
・ユーザーは うさちゃん ⇒ マウスポインターがうさちゃん

・出てくる にんじんをクリックする

・作れるなら ステージ3以降作りたいな〜

Stage1

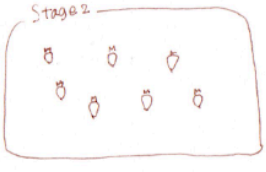
にんじん ちびちびステージ



← 大小差のあるにんじんを5つ用意
番号の順にクリック
↓
成功するとにんじんが うさぎが食べた形になる
5まで終わるとClear
次のステージに進む？今日は終わる？

Stage2

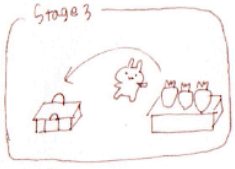
にんじん 10連ステージ



← 小さくて少しクリックしにくい(10つ)
にんじんを用意 → ダブルクリック
↓
成功するとにんじんが消える
10まで終わるとClear
次のステージに進む？今日は終わる？

カバン

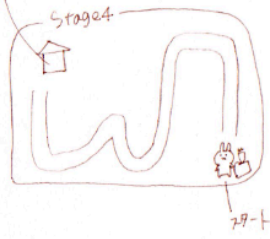
Stage3



← 右ににんじんを3つ用意
ドラッグ操作で
にんじんをうさぎちゃんが自分のカバンにしまう
↓
3つ入ったらClear
次のステージに進む？今日は終わる？

うさぎちゃん家

Stage4



← 道からはみだすとアウト
やりなおし
うさちゃんをお家へつれていく
ロングドラッグ
↓
成功したらコングラッチュレーションページへ
↓
失敗したら終了ページへ

うさちゃんのにんじんクリックハント🥕

草案

楽しくできるクリック練習アプリ制作
・マウスポインターをうさちゃん(うさぎ画像)に変更し、ユーザーがそれを動かしてクリック練習をする
・画面に表示されたにんじん(にんじん画像)をクリック、ダブルクリック、ドラッグすることで進めていく
・違う操作が行われた場合は正しい操作へ導くために「クリックじゃないとだめだよ！」など表示する
・トータルステージは４ステージ

①スタート画面
うさぎとにんじんのイラストの下にスタートボタンがあり、クリックすると１ステージへ遷移する

②**１ステージ**(にんじんもぐもぐステージ)
・画面上に大小差のあるにんじん(番号を書いたにんじんの画像を用意)を５個用意
・ユーザーは番号順にクリックしていく
・成功するとにんじんがかじられたのにんじん画像に代わる(かじられたにんじん画像用意)
・５個クリック終わるとClear画面に遷移

③clear画面は上部にClearと出て下部に横並びのボタンで次に進む？今日は終わる？と出る
・次に進むボタンが押されたら２ステージへ進む
・今日は終わるボタンが押されたら終了画面へと進む。
終了画面にはおつかれさまでしたと文字が出てその下にウサギの画像を設置して、一番下に閉じるボタンを設置し、
閉じるボタンをクリックするとアプリ画面が消えるローカルサーバーを立てた場合はサーバーも一種に切断

④**２ステージ**(にんじん収穫ステージ)
・画面上に小さめのにんじん(にんじん画像準備)を10個用意
・にんじんをダブルクリックするとにんじんが消える
・10個ダブルクリックして全部消えるとClear画面に遷移

⑤clear画面は上部にClearと出て下部に横並びのボタンで次に進む？今日は終わる？と出る
・次に進むボタンが押されたら３ステージへ進む
・今日は終わるボタンが押されたら終了画面へと進む。
終了画面にはおつかれさまでしたと文字が出てその下にウサギの画像を設置して、一番下に閉じるボタンを設置し、
閉じるボタンをクリックするとアプリ画面が消えるローカルサーバーを立てた場合はサーバーも一種に切断

⑥**３ステージ**(にんじんをカバンに入れるステージ)
・画面右ににんじんを３個用意(にんじん画像用意)画面左にかばんを用意(かばん画像用意)
・右においてあるにんじんをドラッグ操作でかばんまで移動させる
・3つ移動出来たらClear画面に遷移

⑦clear画面は上部にClearと出て下部に横並びのボタンで次に進む？今日は終わる？と出る
・次に進むボタンが押されたら４ステージへ進む
・今日は終わるボタンが押されたら終了画面へと進む。
終了画面にはおつかれさまでしたと文字が出てその下にウサギの画像を設置して、一番下に閉じるボタンを設置し、
閉じるボタンをクリックするとアプリ画面が消えるローカルサーバーを立てた場合はサーバーも一種に切断

⑧**４ステージ**(うさちゃんお家へ帰るステージ)
・画面左上にうさちゃんの家を設置(家の画像準備)右下をスタートとして少し入り組んだ家まで道を表示する
(幅はマウスポインターのうさちゃんがギリギリ通れる程度)
・道からはみ出さないようにお家までうさちゃんをドラッグする
失敗したらContinue画面へ成功したらClear画面に遷移

⑨Continue画面は上部にContinue画面と出て下部に横並びのボタンでもう一度? 今日は終わる？と出る
・もう一度ボタンが押されたら４ステージへ戻る
・今日は終わるボタンが押されたら終了画面へと進む。

⑩終了画面にはおつかれさまでしたと文字が出てその下にうさぎの画像を設置して、一番下に閉じるボタンを設置し、
閉じるボタンをクリックするとアプリ画面が消えるローカルサーバーを立てた場合はサーバーも一種に切断

うさちゃんのにんじんクリックハント🥕

要件定義

アプリ名：うさちゃんのにんじんクリックハント🥕
開発技術： Python（Flask） + HTML / CSS / JavaScript
利用環境： ローカルWebアプリ（Flaskサーバーで起動）

1.目的

・パソコン初心者の入門であるマウス操作と楽しく（クリック、ダブルクリック、ドラッグ）を学べる
・かわいいうさちゃんを操作しながら、各ステージで異なるマウス操作に挑戦できる
・ゲーム・遊び感覚で挑戦できる
・1ステージずつ操作が変化し、自然に習得できる構成とする

2.機能

・ログイン画面 ⟶ お名前とスコアという個人情報があるので簡単なログイン画面
・スタート画面 ⟶ うさちゃんとにんじんのイラストとお名前入力BOX、「スタート」ボタンを表示
・ステージ１：クリック練習 ⟶ 番号順ににんじんをクリックして食べる（クリック動作）
・ステージ２：ダブルクリック練習 ⟶ 小さなにんじんをダブルクリックで収穫（ダブルクリック動作）
・ステージ３：ドラッグ練習 ⟶ にんじんをドラッグでかばんに入れる（ドラッグ操作）
・ステージ４：ロングドラッグ練習 ⟶ うさちゃんを道から外れず家までドラッグ移動（ドラッグ＋当たり判定）
・クリア画面 ⟶ 各ステージ後に表示「次へ」「今日は終わる？」ボタン付き（ランキング表示）
・コンティニュー画面 ⟶ ステージ4失敗時に表示「もう一度」「今日は終わる？」ボタン付き
・終了画面 ⟶ 「おつかれさまでした」＋うさちゃん画像＋閉じるボタンでアプリ・サーバー終了

3.ステージ詳細（各ステージスコア記録付き）

ステージ1：にんじんもぐもぐステージ（クリック）
・にんじん画像（大小、番号付き）を5個設置
・番号順にクリックで進行
・正解でかじられた画像に切り替え
・完了後クリア画面へ

ステージ2：にんじん収穫ステージ（ダブルクリック）
・小さなにんじん画像10個をランダムに配置
・ダブルクリックでにんじんが消える
・全て消すとクリア画面へ

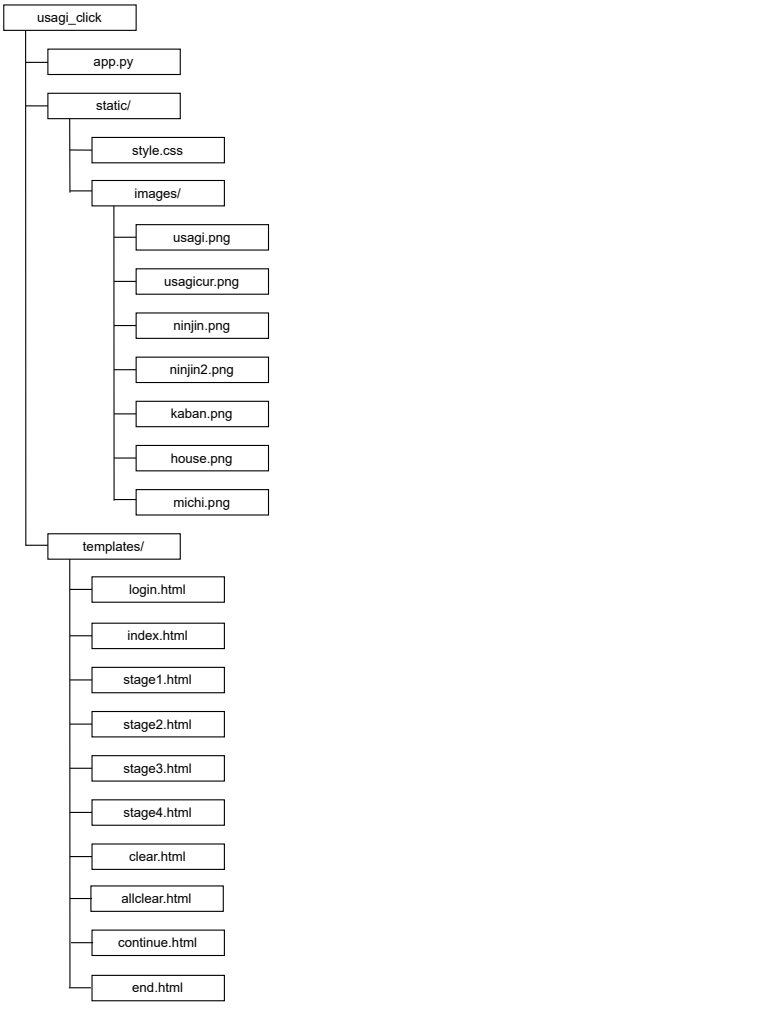
ステージ3：カバンへ収納ステージ（ドラッグ）
・にんじん画像を右側に3つ設置
・左側にカバン画像配置
・ドラッグ＆ドロップでカバンへ
・成功でクリア画面へ

ステージ4：おうちへ誘導（ロングドラッグ＋当たり判定）
・うさちゃん画像をドラッグして家（左上）へ導く
・道を通る背景画像（白い道、黄緑背景）をCanvasで描画
・道から外れたらコンティニュー画面へ
・ゴールですべてクリア画面へ

4. 技術スタック（開発において使われる技術やツール、プログラミング言語）

・サーバーサイド：Python + Flask
・フロントエンド：HTML / CSS / JavaScript

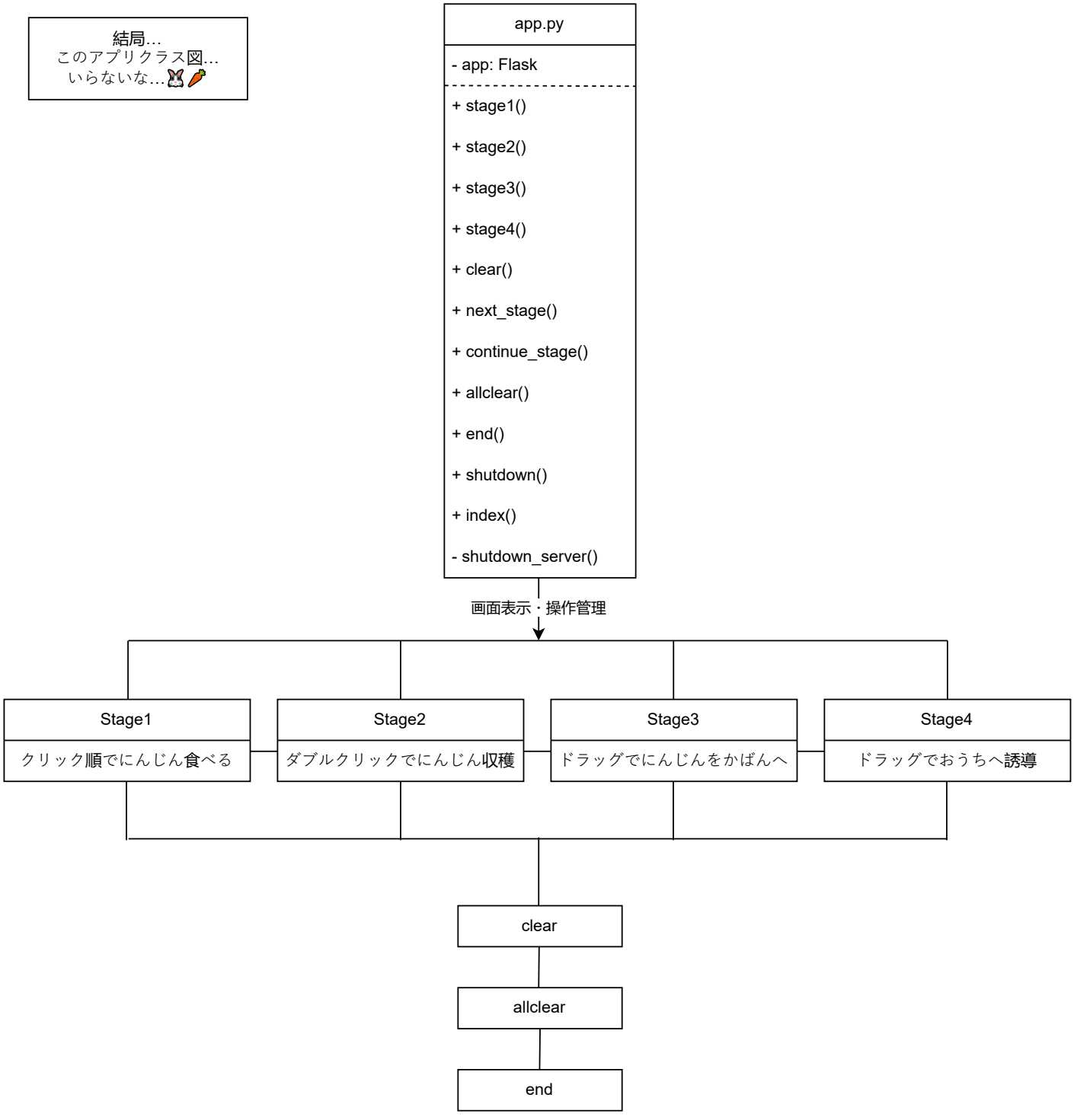
5.ファイル構成



うさちゃんのにんじんクリックハント🥕

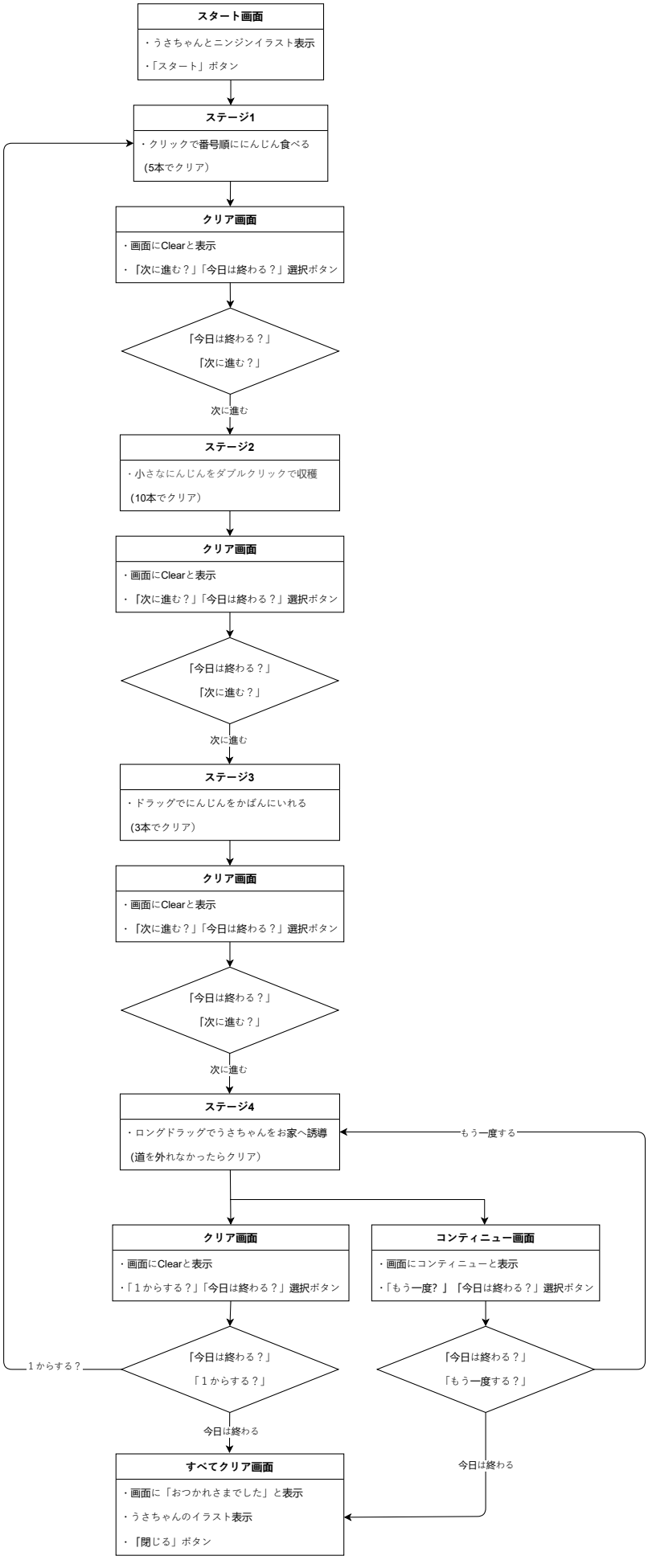
クラス図

結局…
このアプリクラス図…
いらないな…🐰🥕



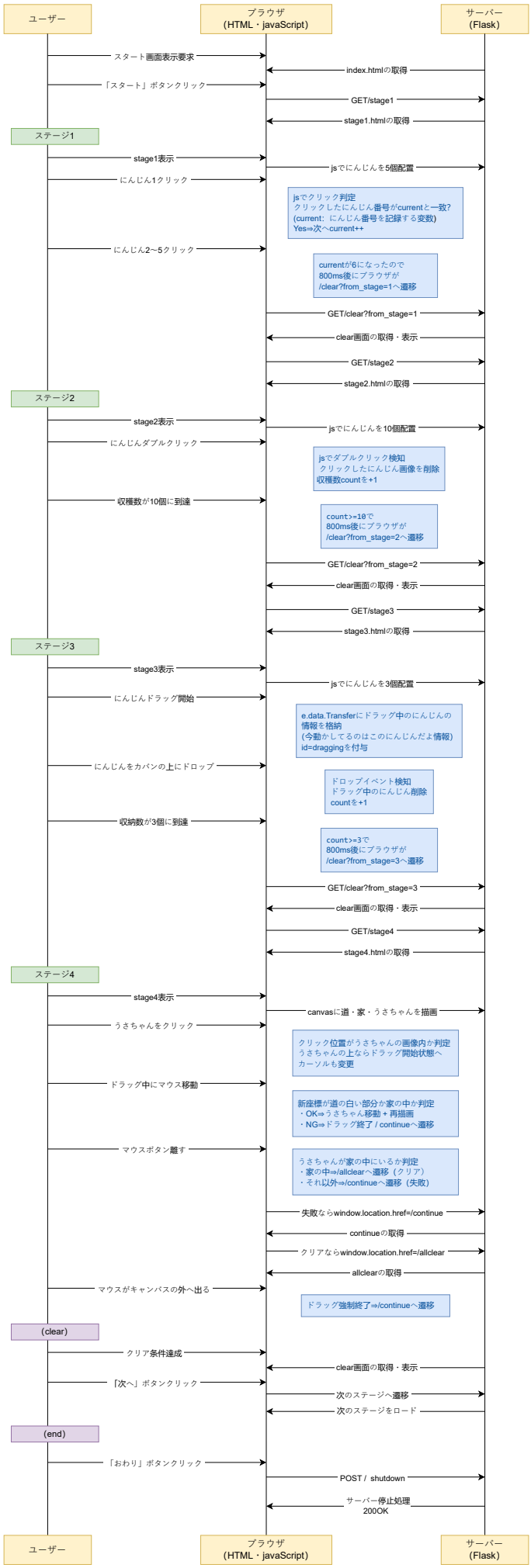
うさちゃんのにんじんクリックハント🥕

画面遷移図



うさちゃんのにんじんクリックハント🥕

シーケンス図



うさちゃんのにんじんクリックハント🥕　　Flaskとは

Flask (フラスク)とは

1.「開発用サーバー」がついている

・Flaskは自分だけで試すのに十分なサーバーを内蔵している
・Flaskだけ でWebアプリを表示、操作できる
・Flaskのサーバーは簡単に動かせる(python app.pyで動く)
・本番には向かないので、開発にだけ使用

※本番に向かない理由
・シングルスレッド(一度に1人しか対応できないからアクセス数が増えるとすぐ止まる)
・セキュリティが弱い(外からの攻撃(悪いリクエストなど)からちゃんと守れない)
・自動で再起動しない(サーバーが止まったら手で再起動しないといけない)
・機能が少ない(ログ記録や複数プロセス処理などの本格機能が足りない)

2. webアプリを作るためのツールである

・ボタンを押したらページが変わる
・名前を入力したらメッセージが表示される
・ゲームのステージが変わるなど

3. 名前の由来

「Flask」は理科の実験で使うフラスコのこと。軽くて持ち運びしやすい、つまり軽量なwebフレームワークのこと

4. Flaskでできること(一部)

【今回の使用した例】

①ページを表示できる
「ホーム画面」「ゲーム画面」「結果画面」などを作って、それぞれのページを見せることができる

```
@app.route('/') #URLのルート(トップページ / )にアクセスしたら、index.htmlを返す
def index():
    return render_template('index.html')
```

http://127.0.0.1:5000にアクセスするとindex.htmlが見えるようになる

②ボタンやリンクでページを切り替えられる
ゲームで「つぎへ」ボタンを押したらステージが変わる動きができる

```
@app.route('/stage1') #/stage1にアクセスしたら、stage1.htmlを返す
def stage1():
    return render_template('stage1.html')
```

ボタンをクリックしたら/stage1にジャンプしてstage1のページが出る

③条件によって動きを変えられる
ステージ3から次はステージ4に進める

```
#/next_stageにPOSTで現在のステージ番号を受け取り、次のステージのページにリダイレクトする
@app.route('/next_stage', methods=['POST'])
def next_stage():
    current = request.form.get('from_stage')
    if current == '1':
        return redirect('/stage2')
    elif current == '2':
        return redirect('/stage3')
    elif current == '3':
        return redirect('/stage4')
    elif current == '4':
        return redirect('/allclear')
    else:
        return redirect('/')
```

「どのステージにいるか」で進む先を変えられる

④サーバーを止めたり、特別な処理もできる

```
#/shutdownにPOSTでアクセスされたらサーバーを停止する処理を呼び出しシャットダウン中のメッセージを返す
@app.route('/shutdown', methods=['POST'])
def shutdown():
    shutdown_server()
    return 'Server shutting down...'
```

ボタンを押してサーバー終了