麻雀アルゴリズム

◾️作りたい物

天和ゲーム

■画面一覧

|  |  |  |
| --- | --- | --- |
| No | ファイル名 | 内容 |
| 1 | index.html | ローカルストレージのloginの状態を参照して、  Login.htmlかtop.htmlに振り分ける |
| 2 | login.html | ・ログインを行う（id,.pwの入力）  ・ログインエラーを出す  ・ログイン成功の場合、top.htmlに遷移する  ・signin.htmlに遷移する |
| 3 | signin.html | ・サインインを行う(id,PW,PW確認の入力)  ・サインインエラーを出す（PWミス、id重複）  ・login.htmlにもどる |
| 4 | signin\_comp.html | ・サインイン成功のフィードバックをする  ・login.htmlにもどる |
| 5 | majang.html | ・配牌する  ・理牌する  ・課金する  ・ログアウトする  ・ポイントを参照する  ・ID名を表示する |
| 6 | charge.html | ・課金を行う  ・majang.htmlに戻る  ・ログアウトする |
|  | admin.html | ・不要なLSのデータを削除する |

■ローカルストレージ設計

□ユーザー

|  |  |  |
| --- | --- | --- |
| No | key | value |
| 1 | id\_[idname] | [pw] |
| 2 | id\_ [idname]\_point | 数値　※自然数 |
| 3 |  |  |
| 4 |  |  |

□ログイン

|  |  |  |
| --- | --- | --- |
| No | key | value |
| 1 | login | [idname] |
| 2 | [haipaino]※1~14 | 数値　※自然数1~34 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

□UIパーツ

|  |  |  |  |
| --- | --- | --- | --- |
| no | パーツ名 | Class名 | 内容 |
| 1 | ボタン | btn | ・画面遷移、配牌等のアクションのトリガー  ・ホバー時にopacity:50%,cursor:pointer |
| 2 | ヘッダー |  | ・アイコン  ・ログアウトボタン |
| 3 | フッター |  | ・サービス名 |
| 4 | 共通画面 |  | ユーザー名を表示する  ポイント名を表示する |
| 5 | 可変画面 |  | 配牌表示、課金画面表示などを行う |
| 6 | 入力フォーム |  | テキストを入力する |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

◾️機能要件

・ボタンAを押すと配牌のイメージで麻雀牌が14枚表示される

Feature Distribution of hai

Given click the url

When push the [haipai]button

Then distributed hais are displayed on the ××area

・ボタンBを押すと14枚の牌を理牌できる

Feature Sorting

Given hais are distributed

When push the [sorting]button

Then sorted hais are displayed on the ××area

・ボタンCを押すと上がり判定をして、上がっていた場合天和エフェクトが出る。

Feature win judgrment

Given hais are distributed

When push the [judgement]button

Then result of judgement are displayed on the○○area

・配牌ボタンを押すと1カウント消費する

Feature Distribution of hai

Given click the url

When push the [haipai]button

Then カウンターが一減る

・課金する

Feature 課金画面表示

Given TOP画面にいる

When 課金ボタンを押す

Then 課金画面が表示される

Feature 課金

Given 課金画面が表示されている

When 金額ボタンを押す

Then 　当該金額がポイント表示画面に追加される

Feature 牌選択

Given 牌が表示されている

When 牌をクリックする

Then 牌が選択モードになり上にズレる

And 選択モードの牌が元に戻る

Feature 捨牌選択

Given 牌が選択モードになっている

When 選択モードの牌をクリックする

Then 手配から牌が消えて左にずれる

And 新しい牌が右に表示される

◾️画面デザイン

・配牌ボタン

・理配ボタン

・上がり判定ボタン

・天和出るまで繰り返すボタン

・配牌表示領域

◾️リソース

・牌画像

・天和エフェクト

◾️開発プロセス

・配牌機能(配列を表示するまで)

　固定配牌をテキストで表示するdone

→ランダムで牌を選びテキストで表示するdone

・画像表示(配列に対応した牌を表示する)

・国士無双判定

・七対子判定

◾️牌の扱い方

□配列を使う

・34種類のkeyで牌の種類を表す

・0〜4の値で牌の枚数を表す

MountArray(i:j):ディクショナリー

i:1〜9,101〜109,201〜209,301〜307

j:0≦j≦4

MyArray(i:j) : ディクショナリー

i:1〜9,101〜109,201〜209,301〜307

j:0≦j≦4

DisplayMyArray(i:j)

i:1〜14

j:牌種別

□クラスを使う

・マークプロパティ

・数値プロパティ

◾️ルール

あがりフラグ:あがり(bool)

雀頭カウント: double

刻子カウント: triple

順子カウント:trio

◾️グループ

まんず1〜9

そうず101〜109

ぴんず201〜209

字牌301〜307

ちゅんちゃんパイ

◾️配牌

以下を14回繰り返す

ランダムにiを選び

MountArray(i)が0でなければ、

MountArray(i)から1を減らして

MyArray(i)に1を加える

◾️処理フロー

「国士判定」&「あがり判定」

→「七対子判定」&「あがり判定」

→「面子手判定」&「あがり判定」

→「最終判定」&「あがり判定」

□あがり判定

あがり=trueであれば処理終了

□国士判定

ちゅんちゃんパイの枚数足し算(sum)

残りをかけ算(multiple)

if sum=0&multiple<>0 then

あがり=true

□七対子判定

雀頭判定して

if doule=7 then

あがり=true

□メンツ手判定

「字牌処理」

→「数牌判定」

□字牌処理

28〜34に

「刻子加算」「雀頭加算」「字牌1,4枚判定」

処理終了時doule>1であれば終わり

□数牌判定

まず初めに「雀頭探し処理」

処理終了時doule>1であれば終わり

if doule=1then

3種に「doule=1 の時処理」

else

3種に「doule=0 の時処理」

□ 雀頭探し処理

i全てに対して

MyArray(i)=2, MyArray(i)=0

の時「雀頭処理」

□ 「doule=1 の時処理」

case1:MyArray(i)が4の時

MyArray(i+1)+ MyArray(i+1)=8then

trio+4で終了

else

「刻子加算」「順子加算」

NEXT

case2:MyArray(i)が3の時

MyArray(i+1)>2,MyArray>2 then

「順子加算」

else

「刻子加算」

NEXT

case3 MyArray(i)が0の時

NEXT

case4 MyArray(i)が1,2の時

「順子加算」

NEXT

□ 「doule=0の時処理」

case1 MyArray(i)が0の時

NEXT

case2 MyArray(i)が1の時

「順子加算」

NEXT

case3 MyArray(i)が2の時

○MyArray(i+1)が0→存在しない

NEXT

○MyArray(i+1)が1 →「雀頭加算」

NEXT

○MyArray(i+1)が2

subcase1 MyArray(i+2)が0→存在しない

NEXT

subcase1 MyArray(i+2)が1→「雀頭処理」

NEXT

subcase3 MyArray(i+2)が2 →「雀頭処理」

NEXT

subcase4 MyArray(i+2)が3★

subcase5 MyArray(i+2)が4★

○MyArray(i+1)が3

subcase1 MyArray(i+2)が0→「雀頭処理」

NEXT

subcase1 MyArray(i+2)が1→「雀頭処理」

NEXT

subcase3 MyArray(i+2)が2 →「雀頭処理」

NEXT

subcase4 MyArray(i+2)が3 →「総当たり法」

subcase5 MyArray(i+2)が4 →「総当たり法」

○MyArray(i+1)が4

subcase1 MyArray(i+2)が0→「雀頭処理」

NEXT

subcase1 MyArray(i+2)が1→「雀頭処理」

NEXT

subcase3 MyArray(i+2)が2 →「総当たり法」

NEXT

subcase4 MyArray(i+2)が3 →「総当たり法」

subcase5 MyArray(i+2)が4 →「総当たり法」

case4 MyArray(i)が3の時

→「総当たり法」

case5 MyArray(i)が4の時

→「総当たり法」

□総当たり法

iからα(0.1)とたしていき

MyArray(i+α)=0に初めてなるまで以下繰り返す

sum+= MyArray(i+α)

string += string(MyArray(i+α))

上がりパーツデータセット(No.連続数,合計数,並び,雀頭数,刻子数,順子)と比較して一致するものがあれば以下の処理を行う

doule,triple,trioに加算、及び対象をゼロにする。

□ 最終判定

doule=1 trio +triple=4 then

あがり

◾️要素処理

□刻子加算

MyArray(i)=3の時

tripleに1を加え、MyArray(i)-3

□雀頭加算

MyArray(i)=2の時

doubleに1を加え、MyArray(i)-2

□字牌1,4枚判定

MyArray(i)=1.4の時

終了

□順子加算

i,i+1,i+2に関してMyArray(i)を取得

tiroに最小値分 追加して、MyArray(i)から引き算