

Web プログラミング最終レポート課題 README

201911419 土佐凜斗

アプリケーションの概要

本課題で作成したのは Web 上で BRACK JACK をプレイできるアプリケーションである。

具体的には対戦相手（ディーラー）対プレイヤーの一対一のブラックジャックで、簡単のため、「ダブルダウン」「インシュランス」「スプリット」などの細かいルールは無しとし、単純にプレイヤーは配られたカードとディーラーのオープンカードから、ヒットするかスタンドするかを選択し、スコアが21に近い方が勝利、バストした時点で敗北とする。

またデータベースにストラテジー表を保存し、自分の得点とディーラーのオープンカードを入力することで状況に応じたストラテジーを参照できる。

基本用語

ヒット：カードを一枚「ドロ」すること

スタンド：カードを引かずにターンを終えること

バスト：得点が 21 を超え、ゲームオーバーすること

ストラテジー表：自分の得点とディーラーのオープンカードから次にヒット、スタンドどちらが勝率が高いかを確率的に示した表で以下のようなものが知られている。

[illegible]

※D（ダブルダウン）と R（サレンダー）はヒットと換算した

ファイル間の関連

wire_flame.jpeg を参照

データ形式

```
sqlite> CREATE TABLE strategy (  
...> player INTEGER,  
...> dealer INTEGER,  
...> strategy TEXT  
...> );
```

プログラムの解説

deal.rb

ゲームスタート時

- ・スタート画面

ゲーム中

- ・ヒットまたはスタンドを選択、ストラテジー表を見る
- ・ルール通りポイントは絵札（11 以上）は 10、A は 1 または 11 とカウントできる
- ・セッションをクリアしてゲームをリスタートできる

action.rb

- ・カードを山札から引く
- ・カードは実際のトランプ同様引けばカードが減るように実装している
- ・プレイヤー、ディーラーそれぞれがスタンド、ヒットした時の表示が変更される

result.rb

- ・勝敗を表示する
- ・もう一度プレイする場合はセッションをクリアしてから deal.rb に移る

strategy.rb

- ・deal.rb からプレイヤーの得点とディーラーのオープンカードの情報を cgi を用いて受け取る

- ・データベースに接続し、状況に適したストラテジーを表示する
- ・データベース接続にエラーが生じたときは例外処理をするようにした

その他詳細な説明は各コードのコメントアウト参照

プログラムにアクセスするための URL

<http://cgi.u.tsukuba.ac.jp/~s1911419/wp/deal.rb>

感想

「Ruby がサーバーサイドで動く」ということと「JavaScript がクライアントサイドで動く」という言葉の意味がこの授業を通して正しく理解できたと思う。

そこが直感的な理解のままで曖昧だったため、初めは Ruby がどうしてこのような書き方になるのか、どうしてこれができないのか、など様々なところでつまづいたが、演習を通して理解してこれてよかった。

最終課題では、実際のようにトランプの山札を実装するところ（ドローしたらそのカードは山札から消える）機能と、ブラックジャック ならではの絵札（11 以上）は 10、A は 1 または 11 の都合の良い方にカウントする機能が難しかった。

このルールのせいで勝利判定、ディーラーがヒットするかスタンドするかを分岐するところも非常に苦労した。

また、掛け金の調整や、ディーラーの強さ設定、画像を用いたカードの視覚的な表現なども実装したかったが、そこまで至ることができなかったので、機会があれば挑戦してみたい。