

Sekappyプログラミングチャレンジ

目次

- 部門(1) コンボが初手に揃う確率の計算ツール
 - 対象者
 - 課題説明
 - 提出物
 - 審査について
- 部門(2) デッキシャッフルツール
 - 対象者
 - 課題説明
 - 提出物
 - 審査について
- 部門(3) カードゲーム版アキネイターの作成
 - 対象者
 - 課題説明
 - 提出物
 - 審査について

募集期間: 2021年8月6日(金)~8月23日(月) 23:59

結果発表: 2021年10月予定

部門(1) コンボが初手に揃う確率の計算ツール

1. 対象者

エンジニア未経験の方

2. 課題説明

TCGには複数のカードを組み合わせて勝利を目指すコンボデッキがいくつも存在します。

そのようなコンボデッキを作る・使う際に、初期手札にコンボパーツが揃う確率がどのくらいなのかは気になるところです。そこで、デッキリストと初期手札の枚数とキーパーツの種類数などを指定し、キーパーツが1枚以上ずつ初手に揃う確率を計算するツールを作成してください。

様々なTCGに対応するように、初期手札の枚数やデッキ枚数も可変としています。

キーパーツを3種類以上入力した時の確率や、キーパーツのそれぞれの必要枚数を指定したときの確率などの発展形を実装できればなお良いでしょう。

初手の引き直し(マジックでいうところのマリガン)は考慮しない。

例はMTGやSVのデッキとしますが、自身の好きなTCGでデッキを作成してください。

[入力]

- ・デッキリスト(テキストファイル等)
- ・デッキの総枚数(正の整数で入力)
- ・初期手札の枚数(正の整数で入力)
- ・初手にほしいキーパーツの名称と種類数(カンマ区切りの文字列で指定)

[出力]

確率は%表示、小数第3位を四捨五入。

- ・初手にほしいキーパーツが指定の枚数以上初手7枚に揃う確率

[例1] MTG:《ティボルトの計略》コンボの場合

[入力]

デッキリスト:テキストファイル

デッキの総枚数:60

初期手札の枚数:7

初手にほしいキーパーツの種類と枚数:{ティボルトの計略, トーモッドの墓所}

[出力]

初手に揃う確率: 18.33%

[例2] 遊戯王OCG:《封印されしエクゾディア》コンボの場合

[入力]

デッキリスト:テキストファイル

デッキの総枚数:40

初期手札の枚数:5

初手にほしいキーパーツの種類と枚数:{封印されしエクゾディア, 封印されし者の右腕, 封印されし者の左腕, 封印されし者の右足, 封印されし者の左足}

[出力]

初手に揃う確率:0.000152%

3. 提出物

①ソースコード

- ・推奨言語(PHP, Ruby, Python,C, C#, C++, Java, JavaScript)
- ・圧縮してフォームにアップロードしてください

※各種ライブラリやフレームワーク、APIなどの外部サービスの利用は自由としますが、各規約等に従って利用してください。

②再現手順・操作手順

- ・フォームに入力して提出してください。

③アピールポイント

- ・フォームに入力して提出してください。

4. 審査について

審査はSekappy社内の審査員が行います。

- ・コードの保守性
- ・利便性(UI/UX)
- ・アイデアの独自性
- ・アルゴリズムの効率良さ

以上の観点で評価し、優秀者には表彰を行います。

部門(2) デッキシャッフルツール

1. 対象者

エンジニア実務経験3年未満の方(未経験の方も可)

2. 課題説明

カードゲームはデッキ(山札)をシャッフルすることが多いです。ゲーム開始前のシャッフル、ゲーム中にデッキからカードを探した後にシャッフルなど様々なタイミングでシャッフルを行います。そこで、シャッフルの種類やシャッフルした回数に応じて、シャッフル前後でデッキ内のカード順番がどう変わったかを確認するプログラムを作成してください。

[実装機能]

・シャッフルの種類は3つ以上作成してください。

→ヒンズーシャッフル、ディールシャッフル、ファローシャッフル、リフルシャッフルなど。

・シャッフルにランダム性を持たせ、同じシャッフル・回数でも異なる結果になるようにしてください。

→ヒンズーシャッフルの場合、デッキの中央から下半分を上に移動ではなく、デッキの中央付近(前後10枚など)から下半分を上に移動するなど。

[入力]

・シャッフルの種類

・シャッフルの回数

[出力]

・シャッフル前のデッキ内容

・シャッフル後のデッキ内容

[例] 例はMTGのデッキとしますが、自身の好きなカードゲームでデッキを作成してください

[デッキ例]

※デッキはテキストファイルから読み込むなどで情報を取得しておく

出現の根本原理

ゼイゴスのトライオーム

~中略~

絶滅の契機

森

[入力]

※シャッフルの種類および回数を入力した後に、続行の可否を問いかける場合

シャッフルの種類:ディールシャッフル

回数:1

シャッフルを続けますか?:はい

シャッフルの種類:ファローシャッフル

回数:3

シャッフルを続けますか?:はい

シャッフルの種類:ヒンズーシャッフル

回数:5

シャッフルを続けますか?:いいえ

[出力]

※シャッフル前のデッキ内のカードを出力

出現の根本原理

ゼイゴスのトライオーム

~中略~

絶滅の契機

森

※シャッフル後のデッキ内のカードを出力

島

海の神のお告げ

~中略~

キオーラ、海神を打ち倒す

出現の根本原理

3. 提出物

①ソースコード

- ・推奨言語 (PHP, Ruby, Python, C, C#, C++, Java, JavaScript)
- ・圧縮してフォームにアップロードしてください

※各種ライブラリやフレームワーク、APIなどの外部サービスの利用は自由としますが、各規約等に従って利用してください。

②再現手順・操作手順

- ・フォームに入力して提出してください。

③アピールポイント

- ・フォームに入力して提出してください。

4. 審査について

審査はSekappy社内の審査員が行います。

- ・コードの保守性
- ・利便性(UI/UX)
- ・アイデアの独自性
- ・アルゴリズムの効率良さ

以上の観点で評価し、優秀者には表彰を行います。

部門(3) カードゲーム版アキネイターの作成

1. 対象者

誰でも

2. 課題説明

質問への回答をもとに実在または架空の人物・キャラクターを推測するアキネイター。そのカードゲーム版を作成してください。ただし条件としてベイズの定理を利用する絞り込みを使用してください。

[入力]

質問に対する以下の回答を複数回

回答1:はい

回答2:いいえ

回答3:わからない

回答4:たぶんそう、部分的にそう

回答5:たぶん違う、そうでもない

[出力]

入力者が想定したカード名

[例]想定したカードが<<ブルーアイズ・ホワイト・ドラゴン>>の場合

質問1:漫画にでた？

回答1:はい

質問2:クリーチャー？

回答2:いいえ

質問3:モンスター？

回答1:はい

質問4:ドラゴン？

回答1:はい

質問5:主人公が使った？

回答4:たぶんそう、部分的にそう

質問6:光属性？

回答1:はい

結論

もしかして:ブルーアイズ・ホワイト・ドラゴン

3. 提出物

①ソースコード

- ・推奨言語 (PHP, Ruby, Python, C, C#, C++, Java, JavaScript)

- ・圧縮してフォームにアップロードしてください

※各種ライブラリやフレームワーク、APIなどの外部サービスの利用は自由としますが、各規約等に従って利用してください。

②再現手順・操作手順

- ・フォームに入力して提出してください。

③アピールポイント

- ・フォームに入力して提出してください。

4. 審査について

審査はSekappy社内の審査員が行います。

- ・コードの保守性

- ・利便性 (UI/UX)

- ・アイデアの独自性

- ・アルゴリズムの効率良さ

以上の観点で評価し、優秀者には表彰を行います。