



### 【ユースケース：0.システムを起動する】

- ・概要：ユーザがシステムを起動し，メニューを表示する．
- ・事前条件：なし
- ・トリガー：ユーザがアプリケーションを起動する．
- ・基本フロー：
  1. ユーザがアプリケーションを起動する．
  2. システムはコマンドを入力するためのメニューを表示する．
- ・代替フロー：
  - 特になし．

### 【ユースケース：1.初期設定をする】

- ・概要：ユーザがポーカーゲームの初期設定を行う．
- ・事前条件：システムが起動済みでメニューが表示されていること．
- ・トリガー：ユーザがメニューから対応する数字を入力する．
- ・基本フロー：
  1. ユーザがメニューから「ゲームを始める」に対応する数字を入力する．
  2. システムはユーザの名前を入力するように求める．
  3. ユーザが名前を入力する．
  4. システムは，入力されたユーザの名前で，ユーザオブジェクトを生成する．
  5. システムは，3人のCPUを生成する．
  6. システムは，CPU（ディーラー）を生成する．
  7. システムはポーカーテーブルを生成し，5人のプレイヤ(ユーザ，CPU，ディーラー)を登録する．
  8. システムはゲームの対戦回数を5にセットする．
  9. システムはゲームをスタートさせる．
- ・代替フロー：
  - 3a. 名前が省略されたら「名無しユーザ」の名前で進める．
- ・事後条件：プレイヤがポーカーテーブルに登録され，ゲームが開始される．

## 【ユースケース：2.ポーカーゲームをする】

- ・概要：ユーザが3人のCPU，ディーラーとポーカーをする．
- ・事前条件：ユースケース「1.初期設定をする」が完了している
- ・トリガー：ユースケース「1.初期設定をする」が完了した．
- ・基本フロー：
  1. システムは，5回勝負のポーカーの開始を表示する．
  2. システムはユーザに，かけるチップの枚数を尋ねる．
  3. ユーザは，番号でチップの枚数を入力する．
  4. システムは，3人のCPUのかけるチップの枚数を決めさせる．
  5. システムは，山札をフルデッキにし，ジョーカーを1枚加え，シャッフルする．
  6. システムは，全員に5枚の手札を配る．
  7. システムはユーザに，交換するカードを数字で求める．
  8. 8-1. ユーザは対応するコマンドを入力し，番号で交換するカードを入力する．
    - 8-2. ユーザは対応するコマンドを入力し，他のプレイヤと会話する．
    - 8-2. ユーザは対応するコマンドを入力し，自分の手札をみる．
    - 8-2. ユーザは対応するコマンドを入力し，役の一覧をみる．
  9. システムは，3人のCPUの交換するカードを決めさせる．
  10. システムは，ディーラーの交換するカードを決めさせる．
  11. システムは全員のカードを交換する．
  12. システムはユーザに，フォールドするかレイズするかを尋ねる．
  13. 13-1. ユーザは対応するコマンドを入力し，フォールドするかレイズするかを決める．
    - 13-2. ユーザは対応するコマンドを入力し，ほかのプレイヤの様子を観察する．
    - 13-3. ユーザは対応するコマンドを入力し，自分の手札を見る．
    - 13-4. ユーザは対応するコマンドを入力し，用語の説明を見る．
  14. システムは，3人のCPUオブジェクトのレイズ/フォールドを決めさせる．
    - レイズ：かけたチップを倍にしてディーラーと勝負する．
    - フォールド：勝負を降りる．
  15. システムは，レイズしたプレイヤオブジェクトの手札を表示する．
  16. システムは，ディーラーの手札を表示する．
  17. システムは，レイズしたプレイヤとディーラーの手札を比較し，各勝敗を判定する．
  18. 18-1. システムは，勝ったプレイヤのかけたチップを倍にして返す．
    - 18-2. システムは，引き分けたプレイヤのかけたチップをそのまま返す．
    - 18-3. システムは，負けたプレイヤのかけたチップを没収する．
  19. システムはプレイヤの状態（名前，勝率，チップ，出した一番強い役，スコア）を表示する．
  20. 勝負の回数が5回未満なら，2.に戻る．5回終わったら，ゲームを終了する．
- ・代替フロー：
  - 3a. ユーザーのチップが0の時はパスさせる．
  - 4a. チップ0のCPUはパスさせる．全員がパスしたとき，ゲームを終了する．
  - 8a，13a. コマンドが不正の場合，再入力させる．
- ・事後条件：5回のポーカーゲームが終了し，4人のプレイヤの対戦成績が画面に表示されている．

### 【ユースケース：3.対戦成績を記録する】

- ・概要：対戦成績をファイルに記録し，メニューにもどる．
- ・事前条件：ユースケース：2.ポーカーゲームをするが完了している
- ・トリガー：ユースケース：2.ポーカーゲームをするが完了した．
- ・基本フロー：
  1. システムは，ユーザオブジェクトの対戦成績を取得する．
  2. システムは，成績記録用のファイルを追記書き込みでオープンする．
  3. システムは現在時刻を求め，yyyy-MM-dd HH:mm:ss 形式の文字列にする．  
(例： 2018-05-20 12:35:56)
  4. システムは，ファイルの末尾に，以下の形式のコンマ区切りで対戦成績を書き込む．  
[時刻，ユーザ名，スコア，チップの枚数，勝率，一番強い役]
  5. システムはファイルをクローズする．
  6. メニューに戻る．
- ・代替フロー：
  - 特に無し．
- ・事後条件： ファイルに対戦成績が追記され，メニューが表示される．

### 【ユースケース：4.過去の成績を見る】

- ・概要：ユーザーが過去の対戦成績をみる．
- ・事前条件：システムが起動済みでメニューが表示されていること．
- ・トリガー：ユーザがメニューから対応する数字を入力する．
- ・基本フロー：
  1. システムは，成績記録用のファイルを読み込みでオープンする．
  2. システムは，過去の成績を読み出して表示する．
  3. メニューに戻る．
- ・代替フロー：
  - 1a. ファイルが見つからないとき，その旨を表示してメニューに戻る．
- ・事後条件： 過去の対戦成績が表示される．

【ユースケース：5.システムを終了する】

- ・概要：ユーザーがシステムを終了する．
- ・事前条件：システムが起動済みでメニューが表示されていること．
- ・トリガー：ユーザがメニューから対応する数字を入力する．
- ・基本フロー：
  1. システムを終了する．
- ・代替フロー：
  - 特に無し．
- ・事後条件：過去の対戦成績が表示される．



