

現状のゲーム面での問題点

- 行動順が固定

現状では

(自分の手札の交換→相手の手札の交換) × 3

行動の決定 → 自分の攻撃 → 相手の攻撃

という順番が固定されており、自分が先に行動できるのでプレイヤー側に有利過ぎる状況です。

また、素早さがまったく意味のない数値になってしまっており、設定を生かし切れていません。

そこでキャラクターの素早さによって行動順番(攻撃順)を決定するシステムを導入します。

今後の改善予定(案)

- ・見辛いテキスト回り・わかりにくいシステム面

役と攻撃・守備行動を示すテキストボードが見辛く、文字のメリハリが無いので自分の決断が良かったのか悪かったのか、行動と結果が結びつきません。

テキストの重要度によって文字サイズや色合いを変更するようにして、自分の行動の何が良かったのか悪かったのかがプレイヤーに分かりやすいようにしたいと思います。

また攻撃行動・守備行動共に一般的なポーカーゲームからすると少々馴染みが無いものです。相手の役と行動を何を選んでいるかも不明瞭で、駆け引き要素は薄いものになってしまっています。

システムの分かりにくさは説明ボードを用意する・オブジェクトにマウスをホバーすると説明文が出るなどの中から効果的な方法を実装予定です。

今後の改善予定(案)

- 駆け引き回りの課題点

COMの思考ルーチンが単調で、捨て札を見れば大体予測可能です。

そのため、相手の守備行動・攻撃行動・表情(現行ニヤニヤと書かれているボード)などから相手の手札を予測する、というところまで落とし込んでいません。

守備の行動が相手の役を予測すれば有利になるシステムなので、そこをシビアなものにしていきたいです。

またせっかくポーカーなので、相手が自分の手を読んでいるのか否かについても、もっと考慮する必要があるバランスにしていきたいです。

強い役を組めたとしても相手がそれを予測していたら効果的にはならないとなれば、相手をだますための行動も必要になってきて、駆け引きがより面白くなります。

例えば捨て札が多ければ多いほど相手はこちらの情報を知っていることになるので、捨て札の枚数に応じて正確にこちらの手を予測してくるなどは現状でも可能な方法です。

今後の改善予定(案)

- ゲームが単調になりがち

仮に駆け引きが面白いものになったとしても、ずっとポーカーをしていたら**すぐに飽きてしまいます**。

また現状の問題点として、キャラクター毎の**能力格差が大きい**ことが挙げられます。

歴史上の人物を使用している関係上、イメージから来る強さや賢さに明確な差が出てくるのでそこを一律にはしにくく、キャラクター間の**不均衡は大きくなりがち**です。

対策としてキャラクター毎に**配下**を用意することを計画しております。

配下の人数や能力値で差を埋めるという方法は**歴史上の人物という特性**ともかみ合いは悪くありません。

ポーカーの合間に配下を使った別のミニゲームを挟んで、**目先を変えて**またポーカーに臨む様にすれば、飽きるまでの時間も稼げますし、問題点である能力格差も

「このキャラクターは、ポーカーの時は強いけど配下が弱いからこっちでは不利だな」

と**不公平感も減り**つつ、**人物のイメージも損なわず**に済みます。

今後の改善予定(案)

- ポーカー要素をより強く

ポーカーゲームに対戦ゲームのシステムを導入したため、システム面はゲーム要素が強くなっています。

一方でポーカー要素に追加した点は交換回数の増加のみですので、バランス的にもう少しポーカー要素を強めていく予定です。

例えばカジノでポーカーをする場合、ベッド(かけ金)という要素が有りますし、1対1ではなく複数人が参加するのが一般的なルールになります。

対戦相手としてプレイヤー対COM複数人、というのはゲーム自体を根本的に仕様変更する必要があるので「レッツゴー☆本能寺III」作成時に実装予定です。

ベッド要素はうまく落とし込めば、より面白い要素になり得ると考えています。

今後の改善予定(案)

- ベッド要素と交換回数による有利・不利の実装

ポーカーにおいてベットは額が大きければ大きいほどリターンは大きいですが、失った場合のリスクも上昇していきます。

また相手にそのまま勝負するのか、一旦降りるのかといった判断も強いるポーカーにおいて重要な要素です。

例えばベット額に応じて自分の役の強さに乗数がかかる・守備にベットすれば相手の強い役を弱い役として使えるなどメリットがある一方、使いどころを間違えると失って一気に不利になるようなシステムを実装すれば、ポーカー要素をゲーム要素に落とし込めるのではないのでしょうか。

また現状では3回交換固定ですが、交換無しや1回交換で終了するなど、交換回数を残した場合、先に行動しやすくなるなどメリットを付加すれば、駆け引き要素はさらに増えるでしょう。

まとめ

- 改善点は多い

Javascriptを使った動的かつルールに基づいたある程度複雑なプログラムを、大きく破綻無く作ることには成功したものの、プログラム記述上の問題含めまだまだ改善点は多数存在します。

今後継続して学習していく上で、現状の「レッツゴー☆本能寺II」のアップデートはもちろん、新たなプログラム言語で新規作成することなども検討していきます。