

デジタルコンテンツ 1 最終レポート

岩本祐輝 大下沙偉 松村亮直
チーム炒飯

2020 年 8 月 25 日

1 作品の狙い・アイデア

1.1 作品の狙い

自由に遊べる纏まった時間が中々取れず、遊ぶ余裕がない人向けに「短い時間でもパパッと楽しめるゲーム」を作ることが、今回の作品作りの狙いである。

1.2 作品のアイデア

短い時間で遊べるものといえば子供の遊びが考えられる。今回は名前は分らないが 1 から 5 までの数字を 1 つずつ使って大小を比べて勝った回数を競う遊びから着想を得た。

今回作成しようとしているゲームは、手札の中に実際に場に出すまでは数字が確定しないものを混ぜることで、勝つ為には実力に加えて運が必要になるようにする事により、決まった勝ち方が存在しない毎回勝つか負けるかの勝負を楽しめるものになっている。

今回我々が作るゲームの利点を 2 つ述べる。

1 つ目は、今回我々が作るゲームは 1 人用なので、自分以外に参加してくれる人を探す必要も無く、自分と他の人のスケジュールをすり合わせる必要も無く、自分のやりたいと思った時に遊ぶことが出来る。

2 つ目は、今回我々が作るゲームは、自分と相手の手札を見比べて、指定された枚数の手札を選んで勝敗を競うというとてもシンプルなルールなので、その分 1 回 1 回のゲームをやり遂げるのに必要な時間は短くなるので、今回の統一テーマの「短い時間でもパパッと楽しめるゲームを作る」を満たすことが出来る。

手札を何枚か選んで勝負するという単純なシステムに加えて、手札やターン数を設定できるため、短い時間でも楽しめるような作品である。

2 プログラムの仕様

2.1 クラス説明

2.1.1 config クラス

int hund:手札の数 int low:最小手札 int high:最大手札
int turn:ターン数 int tLow:最小ターン数 int tHigh:最大ターン数
int rLow:乱数下限値 int rHigh:乱数上限値 int rMax:最大乱数
上記の変数は実際にはグローバル変数として存在する。

void setting():設定について
void drawConffig():設定を変える時の画面表示
void drawFigure():クリックする図形の描画

2.1.2 myHand・comHand クラス

int[] type:カード（固定値）とダイス（乱数）どちらを配るかを記憶する
int[][] hand:手札を格納する
int temp:ダイス内の2つの数字の大小が逆の時に入れ替えるのに使用する
void decideType():手札としてカードとダイスのどちらを配るかを決める
void handOut():手札の数値を決める

2.1.3 printHand クラス

void printMyhand(int[] type, int[][] hand):自分の手札を画面に表示させる
void printComhand(int[] type, int[][] hand):相手の手札を画面に表示させる
void printInformation(int nowTurn, int MaxTurn, int myWinCount, int comWinCount):
ゲームをするために必要な情報を画面に表示させる

2.1.4 selectHand

int[] myused:同じ手札を選べないようにする int[] comused:同文
int n:自分が手札を選んだ枚数を記憶する
int[][] myHand:自分が選んだ手札を格納する
int[][] comHand:相手が選んだ手札を格納する
void resetUsed(int[] used):変数myused、comusedの中身をリセットする

`void printPlayButton(int n):`自分と相手の手札の比較へ移行するためのボタンを表示させる
`void setMyHand(int [][] set, int hand[][], int i):`配られた自分の手札を選択する
`void setComHand(int [][] hand):`配られた相手の手札を選択する
`void printSelect(int []used, int n):`自分が選んだ手札に順番に 1 から数字が表示されるようにする
`boolean IsClickHand(int i, int x, int y):`マウスカーソルが手札内にあるか判定する
`boolean ISFull(int n):`手札を規定枚数選んだか判定する

2.1.5 judgeHand クラス

`void judge(int player, int com, int i):`自分と相手の手札を比較する
`int Win():`勝った回数を返す
`int Lose():`負けた回数を返す

3 プログラムの実行の様子



図 1: タイトル画面

まずタイトル画面が表示される。エンターキーを押すとゲームがスタートして、Q キーを押せばコンフィグモード（手札・ターン数の設定）に入る。

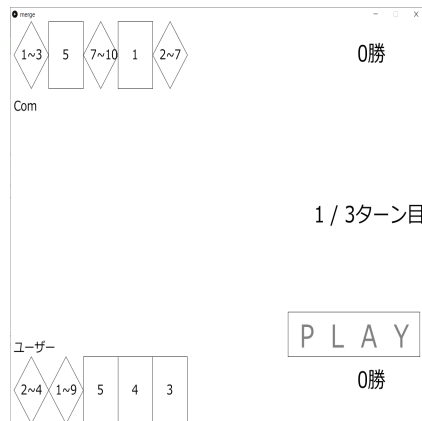


図 2: ゲーム画面その 1

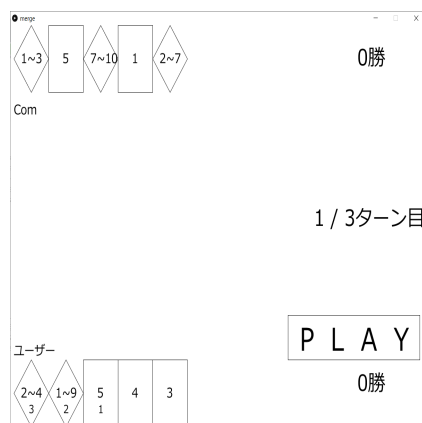


図 3: ゲーム画面その 2

エンターキーを押すと、ゲームが始まる。自分の手札は画面の下側にあるので、それをクリックして手札を選ぶ。規定枚数選ぶと灰色で「PLAY」と表示されていた文字が黒色に変わるので、それをクリックする。

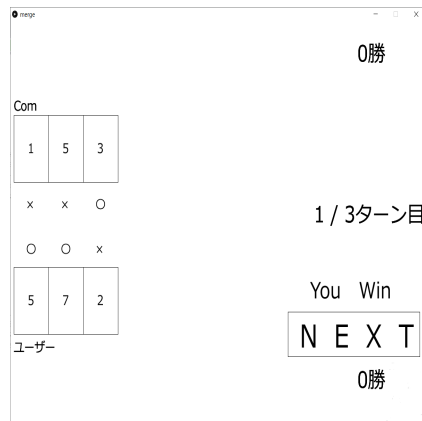


図 4: ゲーム画面その 3

クリックをすると、自分と相手の手札の比較が始まり、勝った回数が多い方が、このターンの勝者となり一点を獲得する。全てのターンが終わっていない時に「NEXT」をクリックすると、次のターンに移る。終わっている時は最終結果画面に移る。

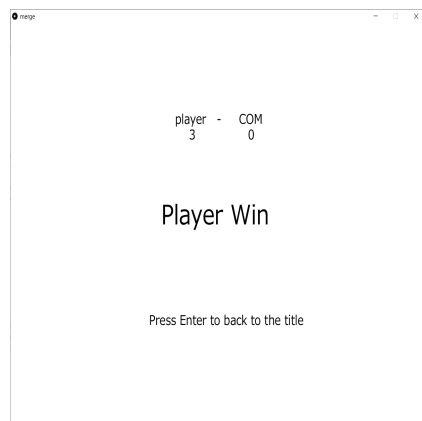


図 5: 最終結果画面

最終的により多くの点数を獲得した方が今回のゲームの勝者となる。エンターキーを押すとタイトル画面に戻る。

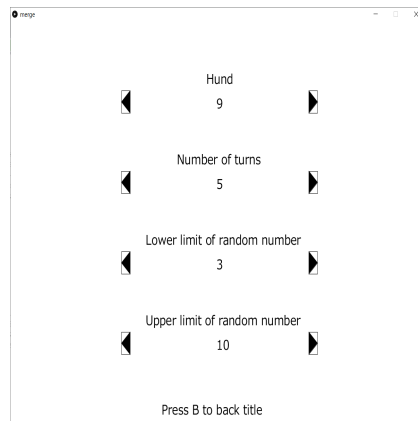


図 6: 設定画面

タイトル画面で Q キーを押すと設定モードに移り、設定画面が表示される。上から手札の枚数、プレイするターン数、乱数の下限値、乱数の上限値を管理する項目が表示されている。左右のボタンをクリックすれば数値を変える事が出来る。設定を変更した後のゲーム画面の例が以下の画像である。

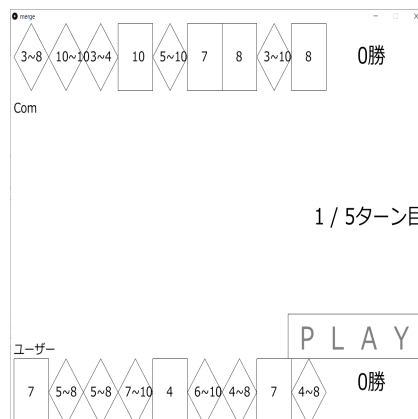


図 7: 設定を変更した後のゲーム画面

4 試用結果

手札がダイスで下限値が 10 で上限値が 10 の時は表示がずれてしまうという問題はあったが、それ以外は当初予定していた通りの挙動を取って、何の支障も無く遊ぶ事が出来た。

手札を選ぶだけなので時間を取らずに遊ぶ事が出来たので、「短い時間でもパパッと楽しめるゲーム」を作るという狙いを達成することは出来たと思われる。

5 発表時に貰ったコメント

- コンピュータはどうやって手札を選んでいるか？
- アイディアがとても面白かった。

6 今後の課題

偶にマウスカーソルがずれて違う手札を選んでしまう事があったので、選んだ手札をリセットしてもう一度選び直せる機能を付けたいと思った。また、今回は相手はランダムで手札を選ぶだけだったので、何かしらの規則に従って手札を選べるようにしたい。