Game

CreateGame( Game , fun \* CheckGameWin, int reals, int numOfPleyers)

Run(

DestroyGame( Game )

int score[4];

Player \*plyr[4];

int roundNum;

Round \*gameRound;

Creates 4 players, creates a round, adds the points to each player, calls for a check and if there is no winner then runs another round

Card

enum suit

enum rank

Deck

struct

* vector \*m\_cards

CreateDeck( Deck )

DestroyDeck( Deck )

PopCard( Deck , \*card)

ShuffleDeck( Deck )

Player

struct

* vector \*m\_cards
* int m\_IsReal
* char m\_name[128];

CreatePlayer ( )

DestroyPlayer ( Player )

GetCard ( Player , card)

ThrowCard ( Player , \*card)

FindCard

UI

PrintCard( Card )

PrintDeck( Deck )

PrintHand( \*Player)

PrintScore( \*Players, int \*arr)

Round

struct

* int m\_cards[4]
* Player\* m\_players
* int m\_score[4]
* int m\_ roundNum

create

destroy

run

static

DealCard( Deck , \*Players , roundNum) //deals the cards

GetCard ( Round, Player , card) //gets the card and put it in the right player

ShiftCards ( Round, \*Players) // Makes a card exchange according to the number of the round (m\_ roundNum). receives and redistributes

ScoreNAddress( Round, \*Players ) //Calculates the result and gives it to the appropriate player

int CheckGameWin(\*Players, int \*arr) return win/not //Called by the game and calculates whether there is a loser and a winner

Divides the deck into 4 and each player receives 13 cards. Runs 13 turns. Each turn receives a card from each player. Determines who gets the score. Calculates the score and gives the player what he deserves.

Game.c Game.h Card.h Deck.h Deck.c Player.h Player.c UI.h UI.c Round.h Round.c main.c