UML diagram

Andrew Hughes | April 4, 2021

main

+ main(): int

BlackJack

- run(): void
- blackJackGame(): void

Memory

+ size:int = 4
+ board:char[size][size]
- boardHid:char[size][size]
- cards[]:char = {A, B, C, D, A, B, C, D< E, F, G, H, E, F, G, H}

+ run(): void
- populate(): void
- display(): void

Deck

- total_Cards: int = 52
- m_deck:Card[52]
- topOfDeck: int

- getValue(): int
- shuffle(): void
- dea(): void
- printCard(): void

Snake

+ width:int
+ height:int
- wall:char
- snake[50][2]:int
- board[25][50]: int
- food[2]:int = {0, 0}
- score:int
- snakeStartLen: int
- snakeSpeedX: int
- snakeSpeedY: int
- split:int
- xmovement: int
- ymovement:int

+ keyboardhit():int
- create_board(): void
- create_snake(): void
- snake_coordinate_update(): void
- install_snake(const int val): void
- print_board(): void
- clear_board(): void
- animation_loop(): void
- sleepcp(int ms): void
- keyboard_responder(const char k): void
- food_creation(): void
- eat_food(): void
- check_walltouch(): void
- pimary_loop(): void
- rum():void

Class: Checkers

+ m_Board:char**
+ m_Player1: Checkers_Player*
- m_Player2: Checkers_Player*

+ Run():Void
- printBoard():Void
- ResetBoard():Void
- promote():Bool
- Gameover():Bool

Class: Checkers_Player

+ m_Board:Char*** = m_Board
+ isBlack:bool

+ isWhite():Bool
- movePiece():void
- validPosition(int Row, int Col):bool
- replace(int Row1, int Col1, int Row2, int Col2): bool
- validMove(int Row1, int Col1, int Row2, int Col2): bool
- validJump(int Row1, int Col1, int Row2, int Col2): bool
- jump(int& Row1, int& Col1, int& Row2, int& Col2): void
- canMove(int Row, int Col) : bool

Hangman

+ word:std::string
+ HMword:std::string
- previousGuesses:std::string
- guess: char
- winner: int = 0
- wordTracker:int = 0
- geussesLeft: int = 6
- playAgain:char

+ run(): void
- wordGenerator(): void
- display(): void
- personDisplay(): void
- isWinner(): void
- reset(): void

Card

- m_face: std::string
- m_suit: std::string
- m_value: int

- card(std::string face, std::string suit, int value): Card
- print():std::string
getValue(): int