

«enumeration»
PawnTeam

NEUTRAL = 0
BLUE = 1
RED = 2

«enumeration»

Direction

NORTH

EAST

WEST

SOUTH

-PawnType pawnType -PawnTeam pawnTeam -int power -bool isHidder -PawnTeam comparaison(Pawn pawn) -int correspondingPower() +Pawn(PawnType pawnType) +Pawn(PawnType pawnType, PawnTeam pawnTeam) +PawnTeam getPawnTeam() +PawnType getPawnType() +getPower() +hool isHidden() +void togglePawnVisibility() +void setPawn(Pawn &pawn) +PawnTeam operator*(Pawn pawn) +Pawn operator==(Pawn pawn)() +void setHidden(bool visibilty)() ostream & operator<<(ostream & out, Pawn & pawn)

Controller

-std::vector<char> readStrategy(PawnTeam team)()
-std::vector<char> fileReaderTeam(std::string fileName)()
-bool isCorrecTile(std::vector<char> pieces)()
-bool stifleSetUp()
-bool stifleSetUp()
-bool ostinueAsking(std::map<char.int>map)()
-void pawnRemains(std::map<char.int> & pawnRemains)()
-bool checkValue(char pawnChar,std::map<char.int>map)()
-void setUp()
-Position interactionWithPlayer(PawnTeam turn)
-void showBoard(PawnTeam teamTurn)
+Controller()
+Void stanGame()

ViewConsole +View() +void displayRemainsPawns() +void displayPawnChar(char pawnChar) +void displayWelcome() +void displayHeader() +void askSetUpFiles() +void fileFeedBack() +void interactiveFeedBack(string teamName) +void displayfileSetUpError() +void fileErrorFeedBack(string fileName) +void displayErrorPawn() +void askPawnReamaing(int row, char column) +void printFormat(int ligne) +void printPawn(Pawn & pawn, PawnTeam & teamTurn) +void printSymbol(string symbol) +void printEmptyLine() +Direction directionChoice() +int nbDeplacementSpy() +bool checkValidInput(string input) +bool checkDirectionInput(string dir) +void printMsg(string msg)

+void printTurn(PawnTeam turn) +void printNbPawnAlives(int nbAlivePawn) +void showEmptyBoard()

Game -PawnTeam teamTurn_ -PawnTeam winner -vector<Pawn> convertCharsToPawns(vector<char>&charPawns, PawnTeam team)() -Pawn correspondingPawn(char pawnChar, PawnTeam team) -void toggleWinner(PawnTeam turn) -void changeTurn() -vector<Pawn> convertCharsToPawns(vector<char>&charPawns, PawnTeam team)() +Game() +Game(vector<char>& teamBluePawn, vector<char>& teamRedPawn)() +PawnTeam getWinner() +bool isOver(PawnTeam turn) +vector<vector<Pawn>>& getGameBoard()() +Board & getBoard()() +getBoard() +bool canMove(PawnTeam hisTeam, Position pos, Direction direction, int deplacement =1)() +void move(Position pos. Direction direction, int deplacement =1)() +bool canFight (PawnTeam hisTeam, Position & pos, Direction direction, int deplacement)() -bool checkOppositePawn(Position firstPawnPosition, Position secondPawnPosition) -bool canFight(Position firstPawnPosition, Position secondPawnPosition) +void fight(Position pos,Direction direction, int deplacement)() +int getAlivePawn(PawnTeam pawnTeam) +PawnTeam &getTeamTurn()

Board

void toggleVisibilityTeam(PawnTeam pawTeam, bool isHidden)()

+Board(int keyAdmin) +void setPawnTest(Position position, Pawn pawn, int keyAdmin)()

+void move(Position pos. Direction direction, int deplacement =1)()

void killPawn(Position pawnPosition, PawnTeam pawnTeam)()

-bool possibleOneMove(Position pos, Direction direction, int deplacement = 1)

-bool possibleMultipleMove(Position pos, Direction direction, int deplacement)
-void changePawnSpecifities(Position firstPosition, Position secondPosition)

+Board(vector<Pawn>& pawnsRedTeam, vector<Pawn>& pawnsBlueTeam)(

void fill(vector<Pawn>&pawnsRedTeam,vector<Pawn>& pawnsBlueTeam)()

+bool canMove(PawnTeam hisTeam, Position pos, Direction direction, int deplacement =1)()

+bool isMovebalePawn(vector<vector<Pawn>> & board, Position pos, PawnTeam pawnTeam)()

+bool canFight (PawnTeam hisTeam, Position pos, Direction direction, int deplacement)()
+void fight(Position pos, Direction direction, int deplacement)
Position newPositionAtterMove(Position pos, Direction dir, int deplacement = 1)

-bool checkIsPositionOccuped(Position pos, Direction direction, int deplacement = 1)

-bool canSpyAttack(PawnTeam team, Position pos, Direction dir, int deplacement)

int getDirectionValue(Direction direction)()

-void toggleVisibilityStatus(Pawn & pawn)()

+Pawn & getPawn(Position pos)(Position pos)

bool isBlocked(Position pos, PawnTeam turn)

-int ownAlivePawn

+Board()

-int oppositeAlivePawn

+bool isInside(Position pos) +vector<vector<Pawn>> &getBoard()()

+int& getRedAlivePawn()()

+int& getBlueAlivePawn()()

+int getBoardSize()