

objPos
<ul style="list-style-type: none"> - x: int - y: int - number: int - prefix: char - symbol: char
<ul style="list-style-type: none"> + objPos() + objPos(xp:int, yp:int, num:int, pref:char, s:char) + objPos(&o: const thisPos) + operator=(&o: const thisPos) objPos& + getX() int const + getY() int const + getNum() int const + getPF() char const + getSym() char const + getReward() int const + setX(xp:int) void + setY(yp:int) void + setNum(num:int) void + setPF(pref:char) void + setSym(sym:char) void + copyObjPos(&thisPos: objPos) + isOverlap(thispos: objPos* const) bool const + printObjPos() void const

<<virtual>>

objPosList

- + ~objPosList()
- + getSize(): int
- + isEmpty(): bool
- + insertHead(thisPos: objPos): void
- + insertTail(thisPos: objPos): void
- + insert(thisPos: objPos, index: int): void
- + getHead(): objPos
- + getTail(): objPos
- + get(index: int): objPos
- + getNext(): objPos
- + resetReadPos(): void
- + set(thisPos: objPos, index: int): void
- + removeHead(): objPos
- + removeTail(): objPos
- + remove(index: int): objPos
- + printList(): void

objPosArrayList
<ul style="list-style-type: none"> - list : objPos* - listSize : int - arrayCapacity : int - currentPos : int
<ul style="list-style-type: none"> + objPosArrayList() + ~objPosArrayList() + getSize() : int + isEmpty() : bool + insertHead(&thisPos : objPos const) : void + insertTail(&thisPos : objPos const) : void + insert(&thisPos : objPos const, index : int) : void + getHead() : objPos const + getTail() : objPos const + get(index : int) : objPos const + getNext() : objPos + resetReadPos() : void + set(&thisPos : objPos const, index : int) : void + removeHead() : objPos + removeTail() : objPos + remove(index : int) : objPos + printList() : void const

GameMechs

- gameBoard: char**
- boardSizeX: int
- boardSizeY: int
- exitFlag: bool
- loseFlag: bool
- delayConst: int
- cmd: char
- playerList: Player**
- playerCount: int
- setExitFlag() void
- setLoseFlag() void
- setInitBoard() void
- checkInput() bool

- + GameMechs()
- + ~GameMechs()
- + getBoardSizeX() int const
- + getBoardSizeY() int const
- + getExitFlag() bool const
- + getLoseFlag() bool const
- + getCMD() char const
- + getBoardRef() char** const
- + getPlayerListRef() Player** const
- + getPlayerCount() int const
- + processInput() void
- + applyDelay() void
- + addPlayer(pl: Player*) void
- + setGameLost() void

ItemBin
<ul style="list-style-type: none"> - myItem: objPos* - gmRef: GameMechs* - boardRef: char** - undrawItem() void - DigitToChar(result: char[]) void
<ul style="list-style-type: none"> + ItemBin(gm: GameMechs*) + ~ItemBin() + drawItem() void + generateItem() void + getItem() objPos

Player
<ul style="list-style-type: none"> - gmRef: GameMechs* - myPos: objPosList* - boardRef: char** - binRef: ItemBin* - myScore: int - killable: bool - myDir: Direction - Direction: enum { STOP, UP, DOWN, LEFT, RIGHT } - undrawPlayer() void - checkCollision() bool - checkSelfCollision() bool - updatePlayerFSM() void
<ul style="list-style-type: none"> + Player(x: int, y: int, symbol: char, gm: GameMechs*, bin: ItemBin*) + ~Player() + drawPlayer() void + movePlayer() void + increaseScore() void + getScore() int const + getPlayerPos() objPosList* const

ScreenDrawer
<ul style="list-style-type: none"> - gmRef: GameMechs* - playerRef: Player* - binRef: ItemBin*
<ul style="list-style-type: none"> + ScreenDrawer(gm: GameMechs*, player: Player*, binRef: ItemBin*) + ~ScreenDrawer() + Draw() void const + DrawEndGame() void const + DrawAverageComputationTime(data: double) void const

<<enumeration>> Direction
STOP UP DOWN LEFT RIGHT