CRC CARDS

Class:Main	Responsibilities	Collaborators
Main	Define the main entry point for the application.	Game
	Create an instance of the Game class.	Game
	Call methods to set up and run the game.	Game

Class: Board	Responsibilities	Collaborators
Board	Initialize the game board with cells, walls, estates, and doors.	Estate, Cell, Person, Player
	Build and manage the list of estates on the board.	Estate
	Manage the players on the board and update their positions.	Person, Player
	Check if a move is valid for a player.	Person, Cell
EmptyCell	Represent an empty cell on the board.	Cell
Wall	Represent a wall cell on the board.	Cell
Estate	Build and represent estates on the board.	Cell
	Provide information about estate properties and cells.	
Door	Represent a door cell associated with an estate.	Estate, Cell
Cell	Define common behavior for cells on the board.	

Method	Description

List<Estate> getEstateList()	Returns the list of estates on the board.
void BuildPeople(HashMap<Integer, Player> playerMap)	Converts a map of player objects to a map of person objects and adds them to the board.
void updatePeopleOnBoard()	Updates the positions of players on the board.
String displayBoard()	Generates and returns a string representation of the current state of the board.
boolean isMoveValid(int playerNum, int moveX, int moveY)	Checks if a move is valid for a player based on proposed coordinates.

Interface: Cell	Responsibilities	Collaborators
Cell	Define common behavior for cells on the board. Provide a default implementation for deleting cells. Provide a default implementation for checking walkability. Provide a default implementation for checking if the cell is a door. Provide a default display character for the cell.	

Method	Description
default void delete()	Provides a default implementation for deleting the cell.
default boolean isWalkable()	Provides a default implementation for checking if the cell is walkable.
default boolean isDoor()	Provides a default implementation for checking if the cell is a door.
default String getDisplayChar()	Provides a default display character for the cell.

Class: Door	Responsibilities	Collaborators
Door	Represent a door cell associated with an estate on the game board.	Cell, Estate
	Store a reference to the associated estate.	Estate
	Provide methods to set and get the associated estate.	Estate
	Indicate if the door is an entrance to an estate.	
	Provide a display character for the door.	
	Implement the Cell interface.	Cell
	Implement the toString method to provide a string representation of the door.	
	Manage associations with multiple estates.	Estate

Method	Description
boolean setEstate(Estate aEstate)	Sets the associated estate for the door.
Estate getEstate()	Returns the associated estate.
String toString()	Returns a string representation of the door.
boolean isEntrance()	Checks if the door is an entrance.
String getDisplayChar()	Returns the display character for the door.

	Responsibilities	Collaborators

Class: Card

Card	Represent a card used in the game. Store attributes such as the card type and name. Provide methods to access the card's type and name. Implement an equals method to compare cards for equality. Implement a toString method to provide a string representation of the card.	

Method	Description
CardType getCardType()	Returns the type of the card (Character, Weapon, or Estate).
String getCardName()	Returns the name of the card.
boolean equals(Card other)	Compare the current card with another card for equality based on card type and name.
String toString()	Returns a string representation of the card, including its type and name.

	Responsibilities	Collaborators
Class: Estate		
Estate	Represent an estate on the game board with its properties and cells.	Inside, Person, Cell
	Store the estate's name, origin, width, height, and associated cells.	Cell
	Manage the list of estate cells and players within the estate.	Cell, Person
	Implement methods to manipulate the estate's cells and players.	
	Implement methods to set the origin and dimensions of the estate.	
	Provide the name of the estate to a string.	
	Implement a method to build the interior cells of the estate.	
	Implement a method to check if a player is inside the estate.	

Methods:

Method	Description
boolean setEstateCells(List<Inside> aEstateCells)	Sets the list of estate cells.
boolean setPlayersInEstate(Set<Person> aPlayersInEstate)	Sets the set of players in the estate.
Estate setOrigin(int row, int col)	Sets the origin (row and column) of the estate.
Estate setWidthHeight(int width, int height)	Sets the width and height of the estate.
List<Inside> getEstateCells()	Returns the list of estate cells.
Set<Person> getPlayersInEstate()	Returns the set of players in the estate.

String toString()	Returns a string representation of the estate.
Estate buildInside()	Builds the interior cells of the estate based on origin, width, and height.
boolean playerIsInside(int x, int y)	Checks if a player is inside the estate based on coordinates.

Class: Game

	Responsibilities	Collaborators
Game	Manage the game flow, turns, players, board, and cards.	Board, Card, Murderer, Player, Estate, ConsoleCommands
	Store the number of players, current player turn, and game state.	Player
	Store lists of all cards, murderer, and player mapping.	Card, Murderer, Player
	Provide methods to get game properties and associations.	
	Implement game setup, player setup, and game running logic.	
	Handle player movement and actions during the game.	Person, Player
	Handle player guesses and solving the mystery.	
	Handle dice rolling and displaying board and turn information.	Board
	Handle passing over the device and controlling the game loop.	

Methods:

Method	Description
int getCurrentPlayerTurn()	Returns the current player turn.
Board getBoard()	Returns the game board.
List<Card> getAllNonMurderCards()	Returns a list of all non-murder cards.
Murderer getMurderer()	Returns the murderer.
Map<Integer, Player> getPlayerMap()	Returns the player mapping.
void Setup()	Handles setting up the game, including creating cards and setting up the

	murderer.
void setupPlayers()	Handles player setup, creating players, allocating cards, and setting positions.
void Run()	Handles running the game loop, turns, and player actions.
boolean currentPlayerInEstate()	Checks if the current player is in an estate.
void handleAttempt(Estate estate)	Handles player guessing or solving when in an estate.
void guess(Estate estate)	Handles player guessing a solution.
void nextGuessRotation(int rotationNumber)	Handles next Guess for next player
void solve()	Handles player attempting to solve the mystery.
List<String> getEstates()	Returns list of Estate Cards
List<String> getWeapons()	Returns a list of weapon card names.
List<String> getCharacters()	Returns a list of character card names.
int[] parseInput(String input, int maxMove)	Parses input into a movement array based on direction and max move distance.
void displayMoveHelp()	Displays help for valid move input.
int[] rollDice()	Rolls dice and returns the result as an array.
void passOverDevice()	Handles passing the device to the next player.
int getRandomNumber(int min, int max)	Generates a random number within a given range.
List<Card> CreateWeaponCards()	Creates and returns a list of weapon cards.
List<Card> CreateCharacterCards()	Creates and returns a list of character cards.

List<Card> CreateEstateCards()	Creates and returns a list of estate cards.
Map<Integer, Player> CreatePlayers(List<Card> cards, int playerNum)	Creates players and allocates cards to them.
void DisplayBoard()	Displays the game board.
void DisplayTurnInfo()	Displays information about the current turn.
void NextPlayerTurn()	Moves the game to the next player's turn.
Card getCardByName(String cardName)	Returns a card class based on its name.
Player getCurrentPlayer()	Returns the current player object.

Class: ConsoleCommands	**Responsibilities**	**Collaborators**
ConsoleCommands	Provides console formatting and color codes. Provides methods for colored console text and screen clearing.	Game

	Responsibilities	Collaborators
Class: Inside		
Inside	Represents a cell inside an estate on the game board. Stores the row and column coordinates of the cell. Provides methods to get and set the row and column coordinates. Implements methods for deletion, displaying, and walkability.	Cell

Methods:

Method	Description
void delete()	Deletes the inside cell.
String toString()	Returns a string representation of the inside cell.
boolean isWalkable()	Returns whether the inside cell is walkable.
String getDisplayChar()	Returns the display character for the inside cell.
int getRow()	Returns the row coordinate of the inside cell.
int getCol()	Returns the column coordinate of the inside cell.

	Responsibilities	Collaborators
Class: Murderer		
Murderer	Represents the murderer in the game with a set of three specific cards: weapon, estate, and person.	Card
	Stores the weapon, estate, and person cards associated with the murderer.	Card
	Provides methods to check if a given set of cards matches the murderer's set.	Card
	Implements a method to get the collection of murder cards.	Card
	Implements a formatted string output for display.	

Methods:

Method	Description
Murderer(Card weapon, Card estate, Card person)	Constructs a Murderer object with the specified weapon, estate, and person cards.
boolean checkMurderer(Card weapon, Card estate, Card person)	Checks if the given weapon, estate, and person cards match the murderer's cards.
Collection<Card> getCards()	Returns a collection containing the weapon, estate, and person cards.
String ToString()	Returns a formatted string representation of the murder scenario.

	Responsibilities	Collaborators
Class: Person		
Person	Represents a person character on the game board. Stores the name of the person and a reference to the associated player. Implements methods to retrieve the current and previous positions of the person. Implements methods to retrieve the current player and player's name. Implements a formatted string output for display.	Player

Methods:

Method	Description
Person(Player player)	Constructs a Person object associated with the given player.
String toString()	Returns a string representation of the person.
boolean isWalkable()	Returns false since a person is not walkable.
String getDisplayChar()	Returns the first character of the person's name for display purposes.
Player getPlayer()	Returns the player associated with the person.
int getX()	Returns the current x-coordinate (row) of the person's position.
int getY()	Returns the current y-coordinate (column) of the person's position.
int getPrevX()	Returns the previous x-coordinate (row) of the person's position.
int getPrevY()	Returns the previous y-coordinate (column) of the person's position.

Class: Player

	Responsibilities	Collaborators
Player	Represents a player in the game. Stores the player's name, guessed status, and list of cards. Implements methods to manipulate and retrieve player attributes. Stores the current position and previous position of the player. Implements methods to set player's position and add a card. Implements methods to get player's name, guessed status, and cards. Implements a formatted string output for displaying player information.	Card

Methods:

Method	Description
Player(String aName)	Constructs a Player object with the given name and initializes the list of cards.
boolean setHasGuessed(boolean aHasGuessed)	Sets whether the player has made a guess.
boolean setCards(List<Card> aCards)	Sets the list of cards held by the player.
boolean setPositionWithOffset(int x, int y)	Sets the player's position with an offset and updates the previous position.
boolean setPosition(int x, int y)	Sets the player's position and updates the previous position.
void addCard(Card card)	Adds a card to the player's list of cards.
String getName()	Returns the name of the player.
boolean getHasGuessed()	Returns whether the player has made a guess.
List<Card> getCards()	Returns the list of cards held by the player.

String toString()	Returns a formatted string representation of the player's name and cards.
int[] getPosition()	Returns the current position of the player.