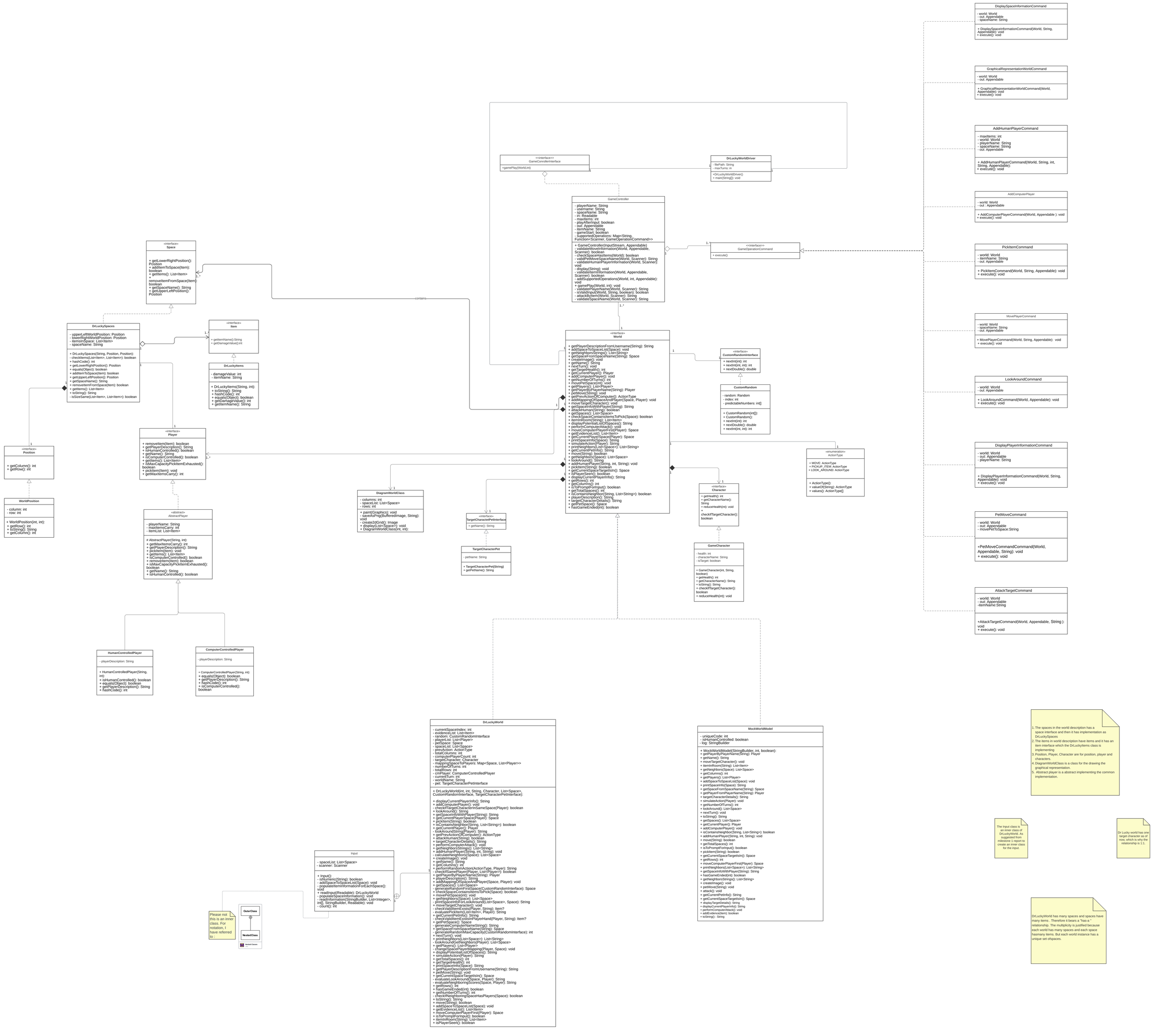
DisplaySpaceInformationCommand
- world: World
- out: Appendable
- spaceName: String
+ DisplaySpaceInformationCommand(World, String, Appendable): void
+ execute(): void
GraphicalRepresentationWorldCommand
- world: World
- out: Appendable
+ GraphicalRepresentationWorldCommand(World, Appendable): void
+ execute(): void
AddHumanPlayerCommand
- maxItems: int
- world: World
- playerName: String
- spaceName: String
- out: Appendable
+ AddHumanPlayerCommand(World, String, int, String, Appendable):
+ execute(): void
AddComputerPlayer
- world: World
- out : Appendable
+ AddComputerPlayerCommand(World, Appendable): void
+ execute(): void
PickItemCommand
- world: World
- itemName: String
- out: Appendable
+ PickItemCommand(World, String, Appendable): void
+ execute(): void
MovePlayerCommand
- world: World
- spaceName: String
- out: Appendable
+ MovePlayerCommand(World, String, Appendable): void
+ execute(): void
LookAroundCommand
- world: World
- out: Appendable
+ LookAroundCommand(World, Appendable): void
+ execute(): void
DisplayPlayerInformationCommand
- world: World
- out: Appendable
- playerName: String
+ DisplayPlayerInformationCommand(World, String, Appendable): void
+ execute(): void
PetMoveCommand
- world: World
- out: Appendable
- movePetToSpace:String
+PetMoveCommandCommand(World, Appendable, String): void
+ execute(): void
AttackTargetCommand
- world: World
- out: Appendable
-itemName:String
+AttackTargetCommand(World, Appendable, String): void
+ execute(): void
<<interface>>
GameControlerInterface
+gamePlay(World,int)
DrLuckyWorldDriver
- filePath: String
- maxTurns: in
+DrLuckyWorldDriver()
+ main(String[]): void
GameController
- playerName: String
- username: String
- spaceName: String
- in: Readable
- maxItems: int
- playAfterInput: boolean
- out: Appendable
- itemName: String
- gameStart: boolean
- supportedOperations: Map<String, Funcion<Scanner, GameOperationCommand>>
+ GameController(InputStream, Appendable)
- validateMoveInformation(World, Appendable, Scanner): boolean
- checkSpaceHasItems(World): boolean
- validPetMoveSpaceName(World, Scanner): String
- validateHumanPlayerInformation(World, Scanner): void
- display(String): void
- validateItemInformation(World, Appendable, Scanner): boolean
- addSupportedOperations(World, int, Appendable): void
+ gamePlay(World, int): void
- validatePlayerName(World, Scanner): String
- isValidInput(World, String, boolean): boolean
- attackByItem(World, Scanner): String
- validateSpaceName(World, Scanner): String
<<interface>>
GameOperationCommand
+ execute()
«interface»
Space
+ getLowerRightPosition(): Position
+ addItemToSpace(Item): boolean
+ getItems(): List<Item>
+ removeItemFromSpace(Item): boolean
+ getSpaceName(): String
+ getUpperLeftPosition(): Position
contains
DrLuckySpaces
- upperLeftWorldPosition: Position
- lowerRightWorldPosition: Position
- itemsInSpace: List<Item>
- spaceName: String
+ DrLuckySpaces(String, Position, Position):
- checkItems(List<Item>, List<Item>): boolean
+ hashCode(): int
+ getLoverRightPosition(): Position
+ equals(Object): boolean
+ addItemToSpace(Item): boolean
+ getUpperLeftPosition(): Position
+ getSpaceName(): String
+ removeItemFromSpace(Item): boolean
+ getItems(): List<Item>
+ toString(): String
- isSizeSame(List<Item>, List<Item>): boolean
«interface»
Item
+ getItemName():String
+ getDamageValue():int
DrLuckyItems
- damageValue: int
- itemName: String
+ DrLuckyItems(String, int):
+ toString(): String
+ hashCode(): int
+ equals(Object): boolean
+ getDamageValue(): int
+ getItemName(): String
«interface»
World
+ getPlayerDescriptionFromUsername(String): String
+ addSpaceToSpaceListSpace): void
+ getNeighborsStrings(): List<String>
+ getSpaceFromSpaceName(String): Space
+ createImage(): void
+ getName(): String
+ nextTurn(): void
+ getTargetHealth(): int
+ getCurrentPlayer(): Player
+ addComputerPlayer(): void
+ getNumberOfTurns(): int
+ movePetSpace(int): void
+ getPlayers(): List<Player>
+ getPlayerByPlayerName(String): Player
+ petMove(String): void
+ getPrevActionOfComputer(): ActionType
+ addMappingOfSpaceAndPlayer(Space, Player): void
+ moveTargetCharacter(): void
+ getSpaceInfoWithPlayer(String): String
+ attackHuman(String): boolean
+ getSpaces(): List<Space>
+ checkSpaceContainsItemsToPick(Space): boolean
+ itemInRoom(String): List<Item>
+ displayPotentialListOfSpaces(): String
+ performComputerAttack(): void
+ moveComputerPlayerFirst(Player): Space
+ getEvidenceList(): List<Item>
+ getCurrentPlayerSpace(Player): Space
+ printSpaceInfo(Space): String
+ simulateActionPlayer(): String
+ printNeighbors(List<Space>): List<String>
+ getCurrentPetInfo(): String
+ move(String): boolean
+ getNeighbors(Space): List<Space>
+ lookAround(): String
+ addHumanPlayer(String, int, String): void
+ pickItem(String): boolean
+ getCurrentSpaceTargetIsIn(): Space
+ isPlayerSeen(): boolean
+ displayCurrentPlayerInfo(): String
+ getRows(): int
+ getColumns(): int
+ isToPromptForInput(): boolean
+ getTotalSpaces(): int
+ isContainsNeighbor(String, List<String>): boolean
+ playerDescription(): String
+ targetCharacterDetails(): String
+ getPetSpace(): Space
+ hasGameEnded(int): boolean
«interface»
CustomRandomInterface
+ nextInt(int): int
+ nextInt(int, int): int
+ nextDouble(): double
CustomRandom
- random: Random
- index: int
- predictableNumbers: int[]
+ CustomRandom(int[]):
+ CustomRandom():
+ nextInt(int): int
+ nextDouble(): double
+ nextInt(int, int): int
«enumeration»
ActionType
+ MOVE: ActionType
+ PICKUP_ITEM: ActionType
+ LOOK_AROUND: ActionType
+ ActionType():
+ valueOf(String): ActionType
+ values(): ActionType[]
«interface»
Character
+ getHealth(): int
+ getCharacterName(): String
+ reduceHealth(int): void
+ checkIfTargetCharacter(): boolean
GameCharacter
- health: int
- characterName: String
- isTarget: boolean
+ GameCharacter(int, String, boolean)
+ getHealth(): int
+ getCharacterName(): String
+ toString(): String
+ checkIfTargetCharacter(): boolean
+ reduceHealth(int): void
«interface»
Position
+ getColumn(): int
+ getRow(): int
WorldPosition
- column: int
- row: int
+ WorldPosition(int, int):
+ getRow(): int
+ toString(): String
+ getColumn(): int
«interface»
Player
+ removeItem(Item): boolean
+ getPlayerDescription(): String
+ isHumanControlled(): boolean
+ getName(): String
+ isComputerControlled(): boolean
+ getItems(): List<Item>
+ isMaxCapacityPickItemExhausted(): boolean
+ pickItem(Item): void
+ getMaxItemsCarry(): int
DiagramWorldClass
- columns: int
- spaceList: List<Space>
- rows: int
+ paint(Graphics): void
- saveAsPng(BufferedImage, String): void
- create2dGrid(): Image
+ display(List<Space>): void
+ DiagramWorldClass(int, int):
«interface»
TargetCharactetPetInterface
+ getName(): String
TargetCharacterPet
- petName: String
+ TargetCharacterPet(String)
+ getPetName(): String
«abstract»
AbstractPlayer
- playerName: String
- maxItemsCarry: int
- itemList: List<Item>
AbstractPlayer(String, int):
+ getMaxItemsCarry(): int
+ getPlayerDescription(): String
+ pickItem(Item): void
+ getItems(): List<Item>
+ isComputerControlled(): boolean
+ removeItem(Item): boolean
+ isMaxCapacityPickItemExhausted(): boolean
+ getName(): String
+ isHumanControlled(): boolean
HumanControlledPlayer
- playerDescription: String
+ HumanControlledPlayer(String, int):
+ isHumanControlled(): boolean
+ equals(Object): boolean
+ getPlayerDescription(): String
+ hashCode(): int
ComputerControlledPlayer
- playerDescription: String
+ ComputerControlledPlayer(String, int):
+ equals(Object): boolean
+ getPlayerDescription(): String
+ hashCode(): int
+ isComputerControlled(): boolean
DrLuckyWorld
- currentSpaceIndex: int
+ evidenceList: List<Item>
- random: CustomRandomInterface
- playerList: List<Player>
- petSpace: Space
- spaceList: List<Space>
- prevAction: ActionType
- totalColumns: int
- computerPlayerCount: int
- targetCharacter: Character
- mappingSpaceToPlayers: Map<Space, List<Player>>
- numberOfTurns: int
- totalRows: int
- cmPlayer: ComputerControlledPlayer
- currentTurn: int
- worldName: String
- pet: TargetCharacterPetInterface
+ DrLuckyWorld(int, int, String, Character, List<Space>, CustomRandomInterface, TargetCharacterPetInterface):
+ displayCurrentPlayerInfo(): String
+ addComputerPlayer(): void
- checkIfTargetCharacterInSameSpace(Player): boolean
+ lookAround(): String
+ getSpaceInfoWithPlayer(String): String
+ getCurrentPlayerSpace(Player): Space
+ pickItem(String): boolean
+ isContainsNeighbor(String, List<String>): boolean
+ getCurrentPlayer(): Player
- lookAroundString(Player): String
+ getPrevActionOfComputer(): ActionType
+ attackHuman(String): boolean
+ targetCharacterDetails(): String
+ performComputerAttack(): void
+ getNeighborsStrings(): List<String>
+ addHumanPlayer(String, int, String): void
- calculateNeighborsISpace): List<Space>
+ createImage(): void
+ getName(): String
+ getColumns(): int
+ performRandomAction(ActionType, Player): String
- checkIfSamePlayer(Player, List<Player>): boolean
+ getPlayerByPlayerName(String): Player
+ playerDescription(): String
+ addMappingOfSpaceAndPlayer(Space, Player): void
+ getSpaces(): List<Space>
- generateRandomFirstSpace(CustomRandomInterface): Space
+ checkSpaceContainsItemsToPick(Space): boolean
+ movePetSpace(int): void
+ getNeighbors(Space): List<Space>
- printSpaceInfoForLookAround(List<Space>, Space): String
+ moveTargetCharacter(): void
- checkValidItemExists(Player, String): Item?
- evaluatePickItem(List<Item>, Player): String
+ getCurrentPetInfo(): String
- checkValidItemExistsInPlayerHand(Player, String): Item?
+ getPetSpace(): Space
- generateComputerNameString(): String
+ getSpaceFromSpaceName(String): Space
- generateRandomMaxCapacity(CustomRandomInterface): int
+ nextTurn(): void
+ printNeighbors(List<Space>): List<String>
- lookAroundGetNeighbors(Player): List<Space>
+ getPlayers(): List<Player>
- changeSpacePlayerMapping(Player, Space): void
+ displayPotentialListOfSpaces(): String
+ simulateActionPlayer(): String
+ getTotalSpaces(): int
+ getTargetHealth(): int
+ printSpaceInfo(Space): String
+ getPlayerDescriptionFromUsername(String): String
+ petMove(String): void
+ getCurrentSpaceTargetIsIn(): Space
- evaluateLookAround(Space, Player): String
- evaluateNeighboringScores(Space, Player): String
+ getRows(): int
+ hasGameEnded(int): boolean
+ getNumberOfTurns(): int
- checkIfNeighboringSpaceHasPlayers(Space): boolean
+ toString(): String
+ move(String): boolean
+ addSpaceToSpaceList(Space): void
+ getEvidenceList(): List<Item>
+ moveComputerPlayerFirst(Player): Space
+ isToPromptForInput(): boolean
+ itemInRoom(String): List<Item>
+ isPlayerSeen(): boolean
Input
- spaceList: List<Space>
- scanner: Scanner
+ Input():
- isNumeric(String): boolean
- addSpaceToSpaceList(Space): void
- populateItemInformationForEachSpace(): void
+ readInput(Readable): DrLuckyWorld
- populateSpaceInformation(): void
- readInformation(StringBuilder, List<Integer>, int[], StringBuilder, Readable): void
- count(): int
Please not this is an inner class. For notation, I have referred to :
OuterClass
NestedClass
Nested Classes
MockWorldModel
- uniqueCode: int
- isHumanControlled: boolean
- log: StringBuilder
+ MockWorldModel(StringBuilder, int, boolean):
+ getPlayerByPlayerName(String): Player
+ getName(): String
+ moveTargetCharacter(): void
+ itemInRoom(String): List<Item>
+ getNeighbors(Space): List<Space>
+ getColumns(): int
+ getPlayers(): List<Player>
+ addSpaceToSpaceList(Space): void
+ printSpaceInfo(Space): String
+ getSpaceFromSpaceName(String): Space
+ getPlayerFromPlayerName(String): Player
+ targetCharacterDetails(): String
+ simulateAction(Player): void
+ getNumberOfTurns(): int
+ lookAround(): List<Space>
+ nextTurn(): void
+ toString(): String
+ getSpaces(): List<Space>
+ getCurrentPlayer(): Player
+ addComputerPlayer(): void
+ isContainsNeighbor(String, List<String>): boolean
+ addHumanPlayer(String, int, String): void
+ move(String): boolean
+ getTotalSpaces(): int
+ isToPromptForInput(): boolean
+ pickItem(String): boolean
+ getCurrentSpaceTargetIsIn(): Space
+ getRows(): int
+ moveComputerPlayerFirst(Player): Space
+ printNeighbors(List<Space>): List<String>
+ getSpaceInfoWithPlayer(String): String
+ hasGameEnded(int): boolean
+ getNeighborsStrings(): List<String>
+ createImage(): void
+ petMove(String): void
+ attack(): void
+ getCurrentPetInfo(): String
+ getCurrentSpaceTargetIsIn(): Space
+ displayTargetDetails(): String
+ displayCurrentPlayerInfo(): String
+ performComputerAttack(): void
+ addEvidence(Item): boolean
+ toString() : String
1. The spaces in the world description has a space interface and then it has implementation as DrLuckySpaces
2. The items in world description have items and it has an item interface which the DrLuckyItems class is implementing
3. Position, Player, Character are for position, player and characters.
4. DiagramWorldClass is a class for the drawing the graphical representation.
5. Abstract player is a abstract implementing the common implementation.
The Input class is an inner class of DrLuckyWorld. As suggested from milestone 1 report to create an inner class for the input.
Dr Lucky world has one target character as of now, which is why the relationship is 1:1.
DrLuckyWorld has many spaces and spaces have many items . Therefore it bears a "has-a " relationship. The multiplicity is justified because each world has many spaces and each space hasmany items. But each world instance has a unique set ofspaces.