Changes that made the model to separate:
Our model is separate from controller and view.
It can handle the corresponding situation and
player actions. Thus, we only add
ReadOnlyWorld interface and implement
getters for View to read information needed in
the model.

<<interface>>
BuildWorld

+ getRoomList() : List<Room>
+ getTargetCharacter() : Piece
+ getPlayerList() : List<Player>
+ getMaxTurn() : int
+ getCurrentPlayer(turn : int) : Player
+ getLocation(player : Player) : Space
+ getItemInRoom(id : int) : List<Item>
+ getNeigList(int id) : List<Space>
+ lookAround(playerId : int) : String
+ displayRoomInfo(playerId : int) : String
+ displayPlayerInfo(playerId : int):String
+ generateMap() : void
+ moveTarget() : String
+ move(playerId : int, roomId : int) : String
+ pickItem(playerId : int, itemId : int) : String
+ addPlayer(name : String, id : int, isHuman : boolean, bagCapacity: int, roomId : int
+ randomNum(max : int) : int
+ attackTarget(playerId : int, itemId : int) : String
+ pokeTarget(playerId : int) : String
+ movePet(roomId : int) : String
+ getPet() : Pet
+ targetHealth() : int
+ displayItemInBag(playerId : int) :String
+ canBeSeen(player : Player) : boolean
+ movePetFollowDfs(startRoomId : int, numMoves : int) :void
+ computerCmd(playerId : int) : String
+ isGameOver : Boolean
+ findRoomId(x : int, y : int) : int
+ displayItemInRoom(playerId : int) : String
+ getTurn() : int
+ nextTurn() : void
+ displayGameStates(playerId : String) : String
+ setMaxTurn(max : int)
+ getFilePath() : String
+ findRoomByPlayerId(playerId : int) : Space

<<interface>>
WorldController

<<interface>>
ReadOnlyWorld

+ getPlayerList():List<Player>
+ getRoomList(): List<Space>
+ findRoomId(x : int, y : int) : int
+ hasItem(playerId : int) : Boolean
+ getMap() : BufferedImage
+ computerAction(playerId : int) : String
+ randomNum(max : int) : int
+ getAllRoomsName() : String

<<interface>>
WorldView

BuildWorldModel

- image : BiufferedImage
- mapCol : int
- mapRow : int
- numRoom : int
- numWeapons: int
- tempRoom : Room
- roomList: List<Room>
- targetName : String
- targetHealth : int
- targetCharacter : Piece
- playerList: List<Player>
- pet: Pet
- petName : String
- winner : String

+ BuidWorldModel(args : String[], turns : int)
- checkOverlappedRoom() : void
- isOverlapped(room1 : Space, room2 : Space) : boolean
- findNeighbors() : void
+ getRoomList() : List<Room>
+ getTargetCharacter() : Piece
+ getPlayerList() : List<Player>
+ getMaxTurn() : int
+ getCurrentPlayer(turn : int) : Player
+ getLocation(player : Player) : Space
+ getItemInRoom(id : int) : List<Item>
+ getNeigList(int id) : List<Space>
+ lookAround(playerId : int) : String
+ displayRoomInfo(playerId : int) : String
+ displayPlayerInfo(playerId : int):String
+ generateMap() : void
+ moveTarget() : String
+ move(playerId : int, roomId : int) : String
+ pickItem(playerId : int, itemId : int) : String
+ addPlayer(name : String, id : int, isHuman : boolean, bagCapacity: int, roomId : int)
+ randomNum(max : int) : int
+ attackTarget(playerId : int, itemId : int) : String
+ pokeTarget(playerId : int) : String
+ movePet(roomId : int) : String
+ getPet() : Pet
+ targetHealth() : int
+ displayItemInBag(playerId : int) :String
+ canBeSeen(player : Player) : boolean
+ movePetFollowDfs(startRoomId : int, numMoves : int) : void
- dfsPath(startRoomId : int) : List<Integer>
+ computerCmd(playerId : int) : String
+ isGameOver : Boolean
+ findRoomId(x : int, y : int) : int
+ displayItemInRoom(playerId : int) : String
+ getTurn() : int
+ nextTurn() : void
+ displayGameStates(playerId : String) : String
+ setMaxTurn(max : int)
+ getFilePath() : String
+ findRoomByPlayerId(playerId : int) : Space

<<interface>>
Pet

+ move(roomId : int) : void
+ getName() : String
+ getCurrentRoomId() : int

TargetPet

- name : String
- currentRoomId : int

+ TargetPet(name : String)
+ move(roomId : int) : void
+ getName() : String
+ getCurrentRoomId() : int
+ equals() : boolean
+ hashCode() : int

<<interface>>
Space

+ getName() : String
+ getItems() : List<item>
+ getUpRow() : int
+ getUpCol() : int
+ getDownRow() : int
+ getDownCol() : int
+ getId() : int
+ getNeighbors() : List<Space>
+ addItem(item: Item) : void
+ removeItem(item: Item) : void
+ isNeigjhbor(room: Space) : void
+ getItemList() : List<Item>
+ getPlayers() : List<Player>
+ addPlayer(Player player) : void
+ removePlayer(Player player) : void

Room

- name : String
- upCol : int
- upRow : int
- downCol : int
- downRow: int
- id: int
- items : List<Item>
- neighbors : List<Space>
- players: List<Player>

+ Room(iname : String, Id : id, upRow : int, id : int, upCol : int, downRow : int, downCol : int)
+ getName() : String
+ getItems() : List<item>
+ getUpRow() : int
+ getUpCol() : int
+ getDownRow() : int
+ getDownCol() : int
+ getId() : int
+ getNeighbors() : List<Space>
+ addItem(item: Item) : void
+ removeItem(item: Item) : void
+ isNeigjhbor(room : Space) : void
+ getItemList():List<Item>
+ getPlayers() : List<Player>
+ addPlayer(Player player) : void
+ removePlayer(Player player) : void
+ equals() : boolean
+ hashCode() : int
+ toSting() : String

<<interface>>
Item

+ getName() : String
+ getPower() : int

<<interface>>
Piece

+ move() : void
+ getHealth() : int
+ getName() : String
+ getCurrentRoomId() : int
+ reduceHealth(damage : int) : void

<<interface>>
Player

+ move(roomId:int) : void
+ getName() : String
+ getId() : int
+ getCurrentRoomId() : int
+ pickItem(item:Item):void
+ getItemList():List<Item>
+ getIsHuman():boolean
+ getBagCapacity():int
+ removeItem(item : Item):void

PlayerCharacter

- currentRoomId : int
- name : String
- id : int
- itemsInBag : List<Item>
- isHuman: boolean

+ PlayerCharacter(name:String, id:int, isHuman : boolean, bagCapacity : int , currentRoomId :int)
+ move(roomId:int) : void
+ getName() : String
+ getId() : int
+ getCurrentRoomId() : int
+ pickItem(item:Item):void
+ getItemList():List<Item>
+ getIsHuman():boolean
+ getBagCapacity():int
+ equals():boolean
+ hashCode():int
+ removeItem(item : Item):void
- sortItemList(List<Item>) : void

Weapon

- name : String
- power : int

+ Weapon(name : String, power : int)
+ getPower() : int
+ getName() : String
+ equals() : boolean
+ hashCode() : int

TargetCharacter

- currentRoomId : int = 0
- name : String
- health : int
- id : numRoom

+ TargetCharacter(name: Sting, health: int, numRoom: int)
+ move() : void
+ getHealth() : int
+ getName() : String
+ getCurrentRoomId() : int
+ equals() : boolean
+ hashCode() : int
+ reduceHealth(damage : int) : void

<<interface>>
WorldView

+ addPlayerBtn(id : int,posX : int, posY : int, posInRoom : int) : void
+ updatePlayerBtn(id : int,posX : int, posY : int, posInRoom : int) : void
+ setTextToWindow(str : String) : void
+ updateTargetBtn(x : int, y : int) : void
+ setCurrentPlayer(playerId : int)
+ onlyShowPlayersInCurrentRoom(roomId : int): void
+ setAction(action : String) : void
+ updateOldRoom(playerId : int): void
+ closeFrame(): void
+ setTextToWindow(str : String):void
+ appendTextToWindow(str : String):void
+ setFeatures (f : Features) : void

WelcomePanel

- welcome : JLabel
- credits : JLabel
- serialVersionUID : Long

+ WelcomePanel()

SwingWorldView

- serialVersionUID : long
- welcomePanel : WelcomePanel
- addPlayerPanel : AddPlayerPanel
- worldPanel : WorldPanel
- worldMenu : JMenuBar
- m : ReadOnlyWorld
- option : JMenu
- exit : JMenuItem
- startNewWorld : JMenuItem
- startCurrentWorld : JMenuItem
- textWindow : JTextArea
- worldMap : JLabel
- target : TargetButton
- num_player : int
- curPlayerId : int
- x : Integer
- y : Integer
- action : String

+ SwingGameView(m: BuildWorld)
- initialJmenu() : void
- createMenuBar() : JMenuBar
- browseFiles(f : Features) : void
- askMaxNumOfTurn() : int
- displayWelcomePanel() : void
- displayAddPlayerPanel() : void
- displayWorldPanel() : void
- convertToRoomId(roomName : String) : int
+ addPlayerBtn(id : int,posX : int, posY : int, posInRoom : int) : void
+ updatePlayerBtn(id : int,posX : int, posY : int, posInRoom : int) : void
+ setTextToWindow(str : String) : void
+ updateTargetBtn(x : int, y : int) : void
+ setCurrentPlayer(playerId : int)
+ onlyShowPlayersInCurrentRoom(roomId : int): void
+ setAction(action : String) : void
+ updateOldRoom(playerId : int): void
+ closeFrame(): void
+ setTextToWindow(str : String):void
+ appendTextToWindow(str : String):void
+ setFeatures (f : Features) : void

AddPlayerPanel

- serialVersionUID: long
- title : JLabel
- nameHint : JLabel
- nameInput : JTextField
- birthRoomHint: JLabel
- birthInput : JTextField
- controlled : JCheckBox
- roomsName : JTextArea
- addPlayer : JButton
- finish : JButton
- buttonMsg : String

+ AddPlayerPanel()
+ showTextFields(): void
- initialLayout() : void
+ getButtonMsg() : String
+ getPlayerName() : String
+ getBagCapacity() : int
+ getRoomName() : String
+ getIsHuman() : Boolean
+ getAddButton() : JButton
+ getFinishButton() : JButton
+ setTextArea(src : String) :void
+ clearInputTextFields() : void

TargetButton

- posX: int
- posY: int
- m : readOnlyWorldModel

+ TargetButton (posX: int, posY: int, m : ReadOnlyWorld)
+ getRoomId(m : ReadOnlyWorld) : int
+ getRoomDc(m : ReadOnlyWorld) : int
+ getRoomDr(m : ReadOnlyWorld) : int
+ paintComponent(g : Graphics)

PlayerButton

- serialVersionUID: long
- id: int
- posX: int
- posY: int
- posInRoom: int

+ PlayerButton (id: int, posX: int, posY: int, posInRoom: int)
+ getId() : int
+ getPosX() : int
+ getPosY() : int
+ getPosInRoom() : int
+ setPosInRoom() : void
+ getRoomId(m : ReadOnlyWorld) : int
+ makeInvisible() : void
+ makeVisible() : void
+ move(posX : int, posY : int, posInRoom : int)
+ paintComponent(g : Graphics) : void
+ refresh() : void

WorldPanel

- serialVersionUID : long
- textWindow: JTextArea
- map : BufferedImage
- worldMap : JLabel
- playerBtnList : List<PlayerButton>

+ WorldPanel(m: ReadOnlyWorld)
+ getTextWindow(): JTextArea
+ getWorldMap(): JLabel
+ getPlayerBtnList(): List<PlayerButton>
+ addPlayerBtn(id : int,posX : int, posY : int, posInRoom : int) : void
+ onlyShowPlayersInCurrentRoom(roomId : int, model : ReadOnlyWorld) : void
- updatePlayerView() : void
+ refresh() : void

<<interface>>
Features

<<interface>>
ReadOnlyWorld

setFeatures method adds click and key listeners to conponents. All add listener methods are helper functions for setFeatures method.

SwingWorldView

<<interface>>
WorldCommand

+ execute(m : BuildWorld)

Move

- roomId : int
- playerId : int

+ Move(playerId : int, roomId : int)
+ execute(m : BuildWorldModel)
+ showView(view: View)

PickItem

- roomId : int
- playerId : int
- view : WorldView

+ PickItem(playerId : int, roomId : int, view : WorldView)
+ execute(m : BuildWorldModel)

DisplayRoomInfo

- roomId : int
- view : WorldView

+ DisplayRoomInfo(roomId : int, view : WorldView)
+ execute(id : player, m : BuildWorldModel)

GenerateMap

+ GenerateMap()
+ execute(m : BuildWorldModel)
+ showView(view: View)

MoveTarget

view : WorldView

+ MoveTarget(view : WorldView)
+ execute(m : BuildWorldModel)

DisplayPlayerInfo

- playerId : int
- view : WorldView

+ DisplayPlayerInfo(playerId : int, view : WorldView)
+ execute(id : player, m : BuildWorldModel)

LookAround

- playerId : int
- view : WorldView

+ LookAround(playerId : int, view : WorldView)
+ execute(id : player, m : BuildWorldModel)

AddPlayer

- name : String
- id : int
- isHuman : boolean
- bagCapacity : int
- roomId : int

+ AddPlayer(name : String. id : int, isHuman : boolean, bagCapacity : int, roomId : int)
+ execute(m : BuildWorldModel)

Attack

- playerId : int
- itemId : int
- withItem : boolean
- view : WorldView

+ Attack(playerId : int, itemId : int, view : WorldView)
+ Attack(playerId : int, view : WorldView)
+ execute(m : BuildWorldModel)

MovePet

- playerId : int
- roomId : int
- view : WorldView

+ MovePet(roomId : int, view : WorldView)
+ execute(m : BuildWorldModel)

<<interface>>
WorldController

+ play(m : BuildWorld, view: View);

SwingController

- model: BuildWorld
- view: WorldView
- cmd : WorldCommand

+ SwingController(model, view)
+ setView(view : WorldView) : void
+ exitProgram() : void
+ addPlayer(name : String. id : int, isHuman : boolean, bagCapacity : int, roomId : int) : void
+ displayPlayerInfo(playerId : int) : void
+ move(playerId : int, roomId : int) : void
+ pick(playerId : int, roomId : int) : void
+ moveTarget() : void
+ lookAround(playerId : int) : void
+ attack(playerId : int, itemId : int) : void
+ movePet(roomId : int) : void
+ startNewWorld(): void
+ startCurrentWorld(): void
+ poke(playerId : int) : void
+ displayItemInRoom(playerId : int) : void
+ displayItemInBag(playerId : int) : void
+ nextTurn() : void
+ showGameState(playerId : int) : void
+ turnStart(turn : int) : void
+ optionsHint() : String

<<interface>>
Features

+ exitProgram() : void
+ addPlayer(name : String. id : int, isHuman : boolean, bagCapacity : int, roomId : int) : void
+ displayPlayerInfo(playerId : int) : void
+ move(playerId : int, roomId : int) : void
+ pick(playerId : int, roomId : int) : void
+ moveTarget() : void
+ lookAround(playerId : int) : void
+ attack(playerId : int, itemId : int) : void
+ movePet(roomId : int) : void
+ startNewWorld(): void
+ startCurrentWorld(): void
+ poke(playerId : int) : void
+ displayItemInRoom(playerId : int) : void
+ displayItemInBag(playerId : int) : void
+ nextTurn() : void
+ showGameState(playerId : int) : void
+ turnStart(turn : int) : void
+ optionsHint() : String

<<interface>>
WorldView

1

<<interface>>
BuildWorld

1