GameTypes

+ RoomTypes : enum
+ PUZZLE_ROOM : RoomTypes
+ DIALOGUE_ROOM : RoomTypes
+ ITEM_ROOM : RoomTypes

DialogueRoom

- fella : NPC

+ DialogueRoom(title : string, description : string, connections : vector<string>, fella : NPC) : constructor
+ ~DialogueRoom() : destructor
+ playerTakeAction() : bool
+ display() : void

ItemPuzzleRoom

- itemPuzzle : ItemPuzzle

+ ItemPuzzleRoom(title : string, description : string, connections : vector<string>, itemPuzzle : ItemPuzzle) : constructor
+ ~ItemPuzzleRoom() : destructor
+ playerTakeAction() : bool
+ display() : void

ItemRoom

+ ItemRoom(title : string, description : string, connections : vector<string>, item : Item) : constructor
+ ~ItemRoom() : destructor
+ playerTakeAction() : bool
+ display() : void

ThinkingPuzzleRoom

+ ThinkingPuzzleRoom(title : string, description : string, connections : vector<string>, dialoguePuzzle : DialoguePuzzle) : constructor
+ ~ThinkingPuzzleRoom() : destructor
+ playerTakeAction() : bool
+ display() : void

Room

- player : Player
- title : string
- description : string
- connections : vector<string>
- state : RoomState
- roomtype : GameTypes::RoomTypes

+ Room(title : string, description : string, connections : vector<string>) : constructor
+ ~Room() : destructor
+ getState() : RoomState
+ playerEnterRoom(player : player) : void
+ getTitle() : string
+ getConnections() : vector<string>
+ entered() : void
+ getRoomType() : GameTypes::RoomTypes
+ isDone() : bool
+ playerTakeAction() : bool
+ display() : void
- setState(state : RoomState) : void

<<extends>>
<<extends>>
<<extends>>
<<extends>>

RoomNode

- room : Room
- up : RoomNode
- down : RoomNode
- north : RoomNode
- east : RoomNode
- south : RoomNode
- west : RoomNode
- next : RoomNode
- previousNode : RoomNode

+ RoomNode(r : Room) : constructor
+ ~RoomNode() : destructor
+ getRoom() : Room
+ getUpNode() : RoomNode
+ getDownNode() : RoomNode
+ getNorthNode() : RoomNode
+ getEastNode() : RoomNode
+ getSouthNode() : RoomNode
+ getWestNode() : RoomNode
+ getNextNode() : RoomNode
+ setNextNode(nextNode : RoomNode) : void
+ setNextNode(up : RoomNode) : void
+ setNextNode(down : RoomNode) : void
+ setNextNode(north : RoomNode) : void
+ setNextNode(east : RoomNode) : void
+ setNextNode(south : RoomNode) : void
+ setNextNode(west : RoomNode) : void
+ setPreviousNode(previousNode : RoomNode) : void
+ getPreviousNode() : RoomNode

RoomList

- UP : int
- DOWN : int
- NORTH : int
- EAST : int
- SOUTH : int
- WEST : int
- head : RoomNode
- tail : RoomNode

+ RoomList() : constructor
+ ~RoomList() : destructor
+ insert(room : Room) : void
+ circley() : void
+ solidify() : void
+ RoomsDone() : bool
+ getHead() : RoomNode

<<includes>>

ItemPuzzle

- expectedItem : Item

+ ItemPuzzle(description : string, item : Item) : constructor
+ ItemPuzzle(itemPuzzle : ItemPuzzle) : copy constructor
+ ~ItemPuzzle() : destructor
+ getExpectedItemName() : string
+ checkItem(item : Item) : void
+ toString() : string

<<extends>>

Puzzle

- description : string
- solved : bool

+ Puzzle(description : string) : constructor
+ ~Puzzle() : destructor
+ wasSolved() : bool
+ toString : string

DialoguePuzzle

- expectedInput : string

+ DialoguePuzzle(description : string, expectedInput : string) : constructor
+ DialoguePuzzle(dialoguePuzzle : DialoguePuzzle) : copy constructor
+ ~DialoguePuzzle() : destructor
+ getInput(answer : string) : void
+ toString() : string

<<extends>>

NPC

+ NPC(name : string, dialogue : string) : constructor
+ NPC(fella : NPC) : copy constructor
+ ~NPC() : destructor
+ display() : void

<<extends>>

<<includes>>

Character

- name : string
- dialogue : string

+ Character(name : string) : constructor
+ ~Character() : destructor
+ getName() : string

Player

- itemList : list<Item*>

+ Player (name : string, item : Item) : constructor
+ ~Player() : destructor
+ pickUp(item : Item) : void
+ drop() : void
+ InspectItem() : void
+ listInventory() : void
+ useItem() : Item
- findItem(item : Item) : bool
- findItem(itemName : string) : bool

<<extends>>

Item

- name : string
- description : string
- useLocation : string

+ Item(name : string, useLocation : string, description : string) : constructor
+ Item(item : Item) : copy constructor
+ ~Item() : destructor
+ getName () : string
+ getDescription() : string
+ getUselocation() : string
+ equals(item : Item) : bool
+ toString() : string

FullyExploredRoom

+ RoomState() : constructor
+ ~RoomState() : destructor
+ isExplored() : bool = true
+ roomDone() : bool = true

RoomState

+ RoomState() : constructor
+ ~RoomState() : destructor
+ isExplored() : bool
+ roomDone() : bool

<<extends>>

ExploredRoom

+ RoomState() : constructor
+ ~RoomState() : destructor
+ isExplored() : bool = true
+ roomDone() : bool = false

<<extends>>

UnexploredRoom

+ RoomState() : constructor
+ ~RoomState() : destructor
+ isExplored() : bool = false
+ roomDone() : bool = false

<<extends>>

Game

- UP : int
- DOWN : int
- NORTH : int
- EAST : int
- SOUTH : int
- WEST : int
- player : Player
- currentNode : RoomNode
- roomlist : RoomList
- yellowKeyCard : Item
- blueKeyCard : Item
- redKeyCard : Item
- knife : Item
- pictureFrame : Item
- will : Item
- map : Item
- yellowItemPuzzle : ItemPuzzle
- blueItemPuzzle : ItemPuzzle
- redItemPuzzle : ItemPuzzle
- studyPuzzle : DialoguePuzzle
- libraryPuzzle : DialoguePuzzle
- atticPuzzle : DialoguePuzzle
- craig : NPC
- steph : NPC
- rose : NPC
- brent : NPC
- craig2 : NPC

+ Game(playerName : string) : constructor
+ ~Game() destructor
+ playGame()
- setCurrentNode(currentNode : RoomNode) : void
- getCurrentNode() : RoomNode
- accuse() : void
- inventory() : void
- displaySplashScreen(condition : int) : void
- moveRoom() : void
- interact() : void
- getPathways() : vector<unsigned>
- listRoomOptions(pathways : vector<unsigned>) : void