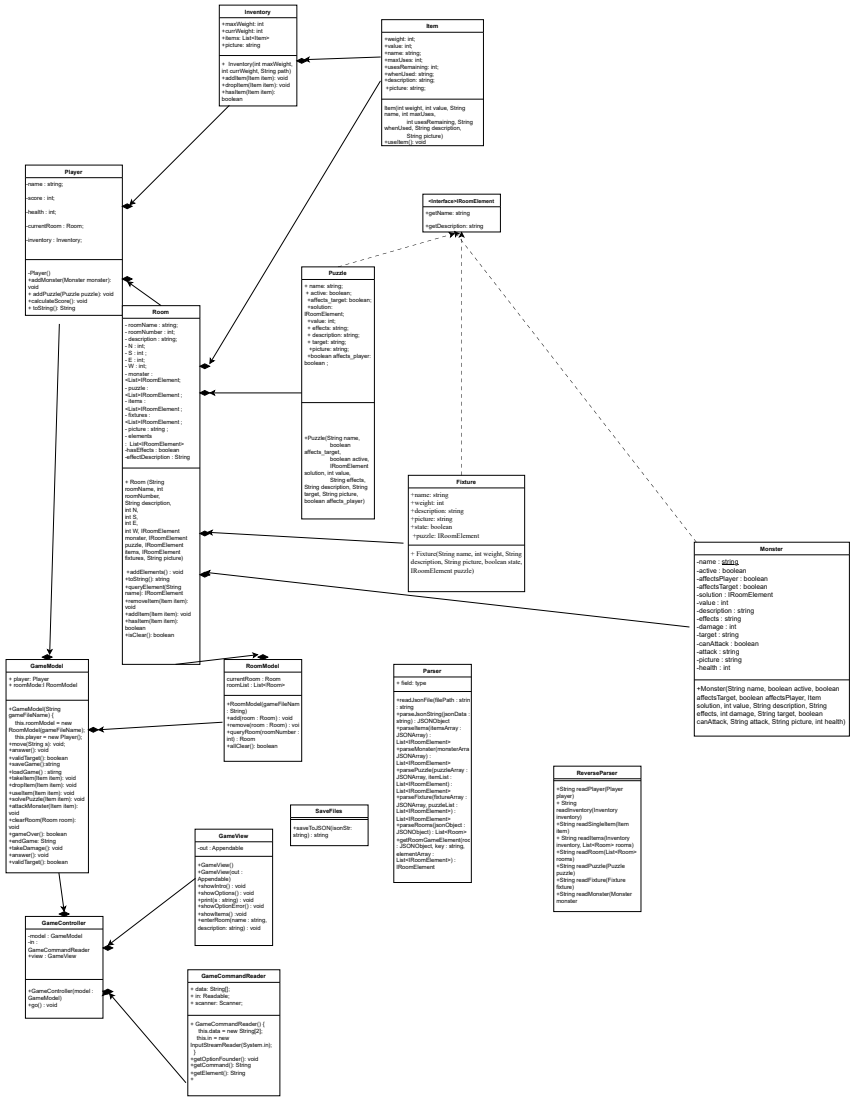
Inventory
+maxWeight: int
+currWeight: int
+items: List<Item>
+picture: string
+ Inventory(int maxWeight, int currWeight, String path)
+addItem(Item item): void
+dropItem(Item item): void
+hasItem(Item item): boolean
Item
+weight: int;
+value: int;
+name: string;
+maxUses: int;
+usesRemaining: int;
+whenUsed: string;
+description: string;
+picture: string;
Item(int weight, int value, String name, int maxUses, int usesRemaining, String whenUsed, String description, String picture)
+useItem() : void
Player
-name : string;
-score : int;
-health : int;
-currentRoom : Room;
-inventory : Inventory;
-Player()
+addMonster(Monster monster): void
+ addPuzzle(Puzzle puzzle): void
+calculateScore(): void
+ toString(): String
+Interface<IRoomElement
+getName: string
+getDescription: string
Puzzle
+ name: string;
+ active: boolean;
+affects_target: boolean;
+solution: IRoomElement;
+value: int;
+ affects: string;
+ description: string;
+ target: string;
+picture: string;
+boolean affects_player: boolean ;
+Puzzle(String name, boolean affects_target, boolean active, IRoomElement solution, int value, String affects, String description, String target, String picture, boolean affects_player)
Room
- roomName : string;
- roomNumber : int;
- description : string;
- N : int;
- S : int ;
- E : int ;
- W : int;
- charactor : ;
- north:IRoomElement;
- south: ;
- East:IRoomElement ;
- Items :
- List<IRoomElement ;
- fixtures :
- List<IRoomElement ;
- picture : string ;
- elements : List<IRoomElement>
+setEffects : boolean
-effectDescription : String
+ Room (String roomName, int roomNumber, String description, int N, int S, int E, int W, IRoomElement monster, IRoomElement puzzle, IRoomElement items, IRoomElement fixtures, String picture)
+ addElements() : void
+toString() : string
+queryElement(String name): IRoomElement
+removeItem(Item item): void
+addItem(Item item): void
+hasItem(Item item): boolean
+isClear(): boolean
Fixture
+name: string
+weight: int
+description: string
+picture: string
+state: boolean
-puzzle: IRoomElement
+ Fixture(String name, int weight, String description, String picture, boolean state, IRoomElement puzzle)
Monster
-name : string
-active : boolean
-affectsPlayer : boolean
-affectsTarget : boolean
-solution : IRoomElement
-value : int
-description : string
-effects : string
-damage : int
-target : string
-canAttack : boolean
-attack : string
-picture : string
-health : int
+Monster(String name, boolean active, boolean affectsTarget, boolean affectsPlayer, Item solution, int value, String description, String effects, int damage, String target, boolean canAttack, String attack, String picture, int health)
GameModel
+ player: Player
+ roomModel: RoomModel
+GameModel(String gameFileName) {
this.roomModel = new RoomModel(gameFileName);
this.player = new Player();
+move(String s): void;
+answer(): void
+validTarget(): boolean
+saveGame():string
+loadGame() : string
+takeItem(Item item): void
+dropItem(Item item): void
+useItem(Item item): void
+solvePuzzle(Item item): void
+attackMonster(Item item): void
+clearRoom(Room room): void
+gameOver(): boolean
+endGame: String
+takeDamage(): void
+answer(): void
+validTarget(): boolean
RoomModel
currentRoom : Room
roomList : List<Room>
+RoomModel(gameFileNam : String)
+add(room : Room) : void
+remove(room : Room) : void
+queryRoom(roomNumber : int) : Room
+allClear(): boolean
Parser
+ field: type
+readJsonFile(filePath : string
+parseJsonString(jsonData : string) : JSONObject
+parseItems(itemsArray : JSONArray) : List<IRoomElement>
+parseMonster(monsterArray : JSONArray) : List<IRoomElement>
+parsePuzzle(puzzleArray : JSONArray, itemList : List<IRoomElement) : List<IRoomElement>
+parseFixture(fixtureArray : JSONArray, puzzleList : List<IRoomElement>) : List<IRoomElement>
+parseRooms(jsonObject : JSONObject) : List<Room>
+getRoomGameElement(roo : JSONObject, key : string, elementArray : List<IRoomElement>) : IRoomElement
ReverseParser
+String readPlayer(Player player)
+ String readInventory(Inventory inventory)
+String readSingleItem(Item item)
+ String readItems(Inventory inventory, List<Room> rooms)
+String readRoom(List<Room> rooms)
+String readPuzzle(Puzzle puzzle)
+String readFixture(Fixture fixture)
+String readMonster(Monster monster
SaveFiles
+saveToJSON(jsonStr: string) : string
GameView
-out : Appendable
+GameView()
+GameView(out : Appendable)
+showInfo() : void
+showOptions() : void
+print(s : string) : void
+showOptionError() : void
+showItems() :void
+enterRoom(name : string, description: string) : void
GameController
-model : GameModel
-in : GameCommandReader
+view : GameView
+GameController(model : GameModel)
+go() : void
GameCommandReader
+ data: String[]
+ in: Readable;
+ scanner: Scanner;
+ GameCommandReader() {
this.data = new String[2];
this.in = new InputStreamReader(System.in);
}
+getOptionFounder(): void
+getCommand(): String
+getElement(): String