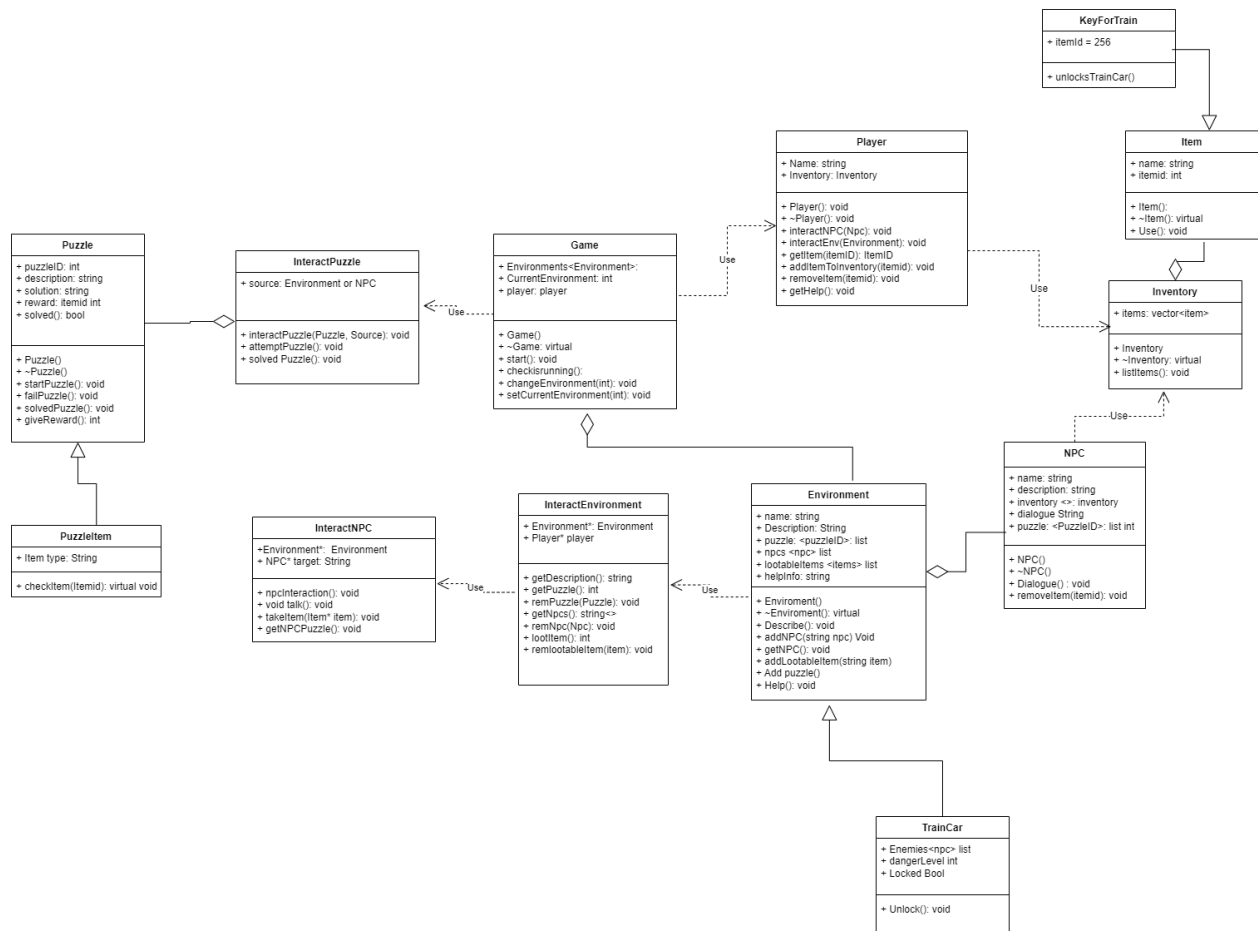# [The Iron Pursuit]

[Group 20]

[Todd Across The Mountain, Nikita Evtushenko, Colby Hanna, and Jem Pineda]

[2024-11-08]

Software Design

Design – Class diagrams

KeyForTrain
+ itemId = 256

+ unlocksTrainCar()

Item
+ name: string
+ itemid: int

+ Item():
+ ~Item(): virtual
+ Use(): void

Player
+ Name: string
+ Inventory: Inventory

+ Player(): void
+ ~Player(): void
+ interactNPC(Npc): void
+ interactEnv(Environment): void
+ getItem(itemID): ItemID
+ addItemToInventory(itemid): void
+ removeItem(itemid): void
+ getHelp(): void

Game
+ Environments<Environment>:
+ CurrentEnvironment: int
+ player: player

+ Game()
+ ~Game: virtual
+ start(): void
+ checkisrunning():
+ changeEnvironment(int): void
+ setCurrentEnvironment(int): void

Puzzle
+ puzzleID: int
+ description: string
+ solution: string
+ reward: itemid int
+ solved(): bool

+ Puzzle()
+ ~Puzzle()
+ startPuzzle(): void
+ failPuzzle(): void
+ solvedPuzzle(): void
+ giveReward(): int

InteractPuzzle
+ source: Environment or NPC

+ interactPuzzle(Puzzle, Source): void
+ attemptPuzzle(): void
+ solved Puzzle(): void

Inventory
+ items: vector<item>

+ Inventory
+ ~Inventory: virtual
+ listItems(): void

NPC
+ name: string
+ description: string
+ inventory <>: inventory
+ dialogue String
+ puzzle: <PuzzleID>: list int

+ NPC()
+ ~NPC()
+ Dialogue() : void
+ removeItem(itemid): void

Environment
+ name: string
+ Description: String
+ puzzle: <puzzleID>: list
+ npcs <npc> list
+ lootableItems <items> list
+ helpInfo: string

+ Enviroment()
+ ~Enviroment(): virtual
+ Describe(): void
+ addNPC(string npc) Void
+ getNPC(): void
+ addLootableItem(string item)
+ Add puzzle()
+ Help(): void

InteractEnvironment
+ Environment*: Environment
+ Player* player

+ getDescription(): string
+ getPuzzle(): int
+ remPuzzle(Puzzle): void
+ getNpcs(): string<>
+ remNpc(Npc): void
+ lootItem(): int
+ remlootableItem(item): void

PuzzleItem
+ Item type: String

+ checkitem(Itemid): virtual void

InteractNPC
+Environment*: Environment
+ NPC* target: String

+ npcInteraction(): void
+ void talk(): void
+ takeItem(Item* item): void
+ getNPCPuzzle(): void

TrainCar
+ Enemies<npc> list
+ dangerLevel int
+ Locked Bool

+ Unlock(): void

(for a more detailed image please check our gitlab)

Class Descriptions

Class Name	Method Name	Description
Game	Game()	Default Constructor
Game	~Game()	Default Destructor
Game	Start()	main loop for the game
Game	checkisRunning()	Check if the main loop of the game is still running
Game	changeEnvironment(int)	When the player decides to move to a different location call changeEnvironment to Change the player's environment
Game	setCurrentEnvironment(int)	when the player walks into a different environment it will set it according to player's location
InteractPuzzle	interactPuzzle(Puzzle, Source)	Identifies the source and the puzzle id of the puzzle that's being interacted with.
InteractPuzzle	attemptPuzzle	Displays the dialogue to let the player know it is attempting the puzzle.

InteractPuzzle	solvedPuzzle	Checks if the puzzle is solved and then sets it as solved once it is.
Puzzle	Puzzle()	Default Constructor
Puzzle	~Puzzle():virtual	Default Destructor
Puzzle	startPuzzle():void	Starts puzzle for the player
Puzzle	failPuzzle():void	Check if the attempt has failed.
Puzzle	solvedPuzzle():void	Check if the puzzle is solved.
Puzzle	giveReward():int	Gives the player an item once it is solved.
PuzzleItem	checkItem(itemid):virtual void	Provides item id for the appropriate puzzle.
Player	Player()	Default Constructor
Player	~Player()	Default Destructor
Player	interactNPC(npc)	Logic for interacting with an NPC
Player	interactEnv(Environment)	Logic for Interacting with Environment
Player	getItem(itemid)	Gets item id from player Inventory
Player	addItem(itemid)	Adds item to players inventory
Player	removeItem(itemid)	Removes Item from Players Inventory
Player	getHelp()	Calls to get help
InteractEnvironment	getDescription()	Prints a description of the current environment
InteractEnvironment	getPuzzle()	Gets Puzzles ID
InteractEnvironment	remPuzzle(Puzzleid)	Removes a puzzle from environment
InteractEnvironment	getNpcs(): String	Gets the Npcs in the environment
InteractEnvironment	remNpc(Npc)	Removes Npc from environment
InteractEnvironment	lootItem()	Grabs Item from environment
InteractEnvironment	remlootableItem()	Removes lootable item from environment
Environment	Environment()	Default Constructor
Environment	~Environment()	Default Destructor
Environment	Describe()	Tells you about the Environment the player is currently in
Environment	addNPC(string NPC)	Adds Environment specific NPCs
Environment	getNPC()	Gives back a list of all NPC in the Environment
Environment	addLootableItem(string item)	Environment might have puzzles that will give a player a lootable item that will help them to progress through the story
Environment	addPuzzle()	Add an Environment puzzle for the player to complete
Environment	Help()	the player can call help to get a hint on how to proceed through the Environment.
NPC	NPC()	Default Constructor
NPC	~Virtual NPC()	Default Destructor
NPC	Dialogue()	Handles logic for NPC Dialogue
NPC	removeItem(itemid)	Removes items from NPC Inventory
InteractNPC	npcInteraction()	Start an interaction sequence that will allow the player to start talking to the NPC
InteractNPC	talk()	Allows the player to talk to the NPC and spawn dialogue
InteractNPC	takeItem(item* item)	NPC might offer items to the player allowing the player to take an item from the NPC

InteractNPC	getnpcPuzzle	Some NPC's might have puzzles associated with them this will start the NPC puzzle sequence
Inventory	Item()	Default Constructor
Inventory	~Inventory()	Default Destructor
Inventory	listItems(): void	List Items in Inventory
item	item()	Default Constructor
item	~Virtual Item()	Default Destructor
item	use()	uses current item
KeyForTrain	unlockTrainCar()	Items that can only unlock the TrainCar
TrainCar	Unlock()	This is called only when the player has the correct item to unlock the TrainCar