CPSC 2720 FALL 2025

Text-Based Adventure Game

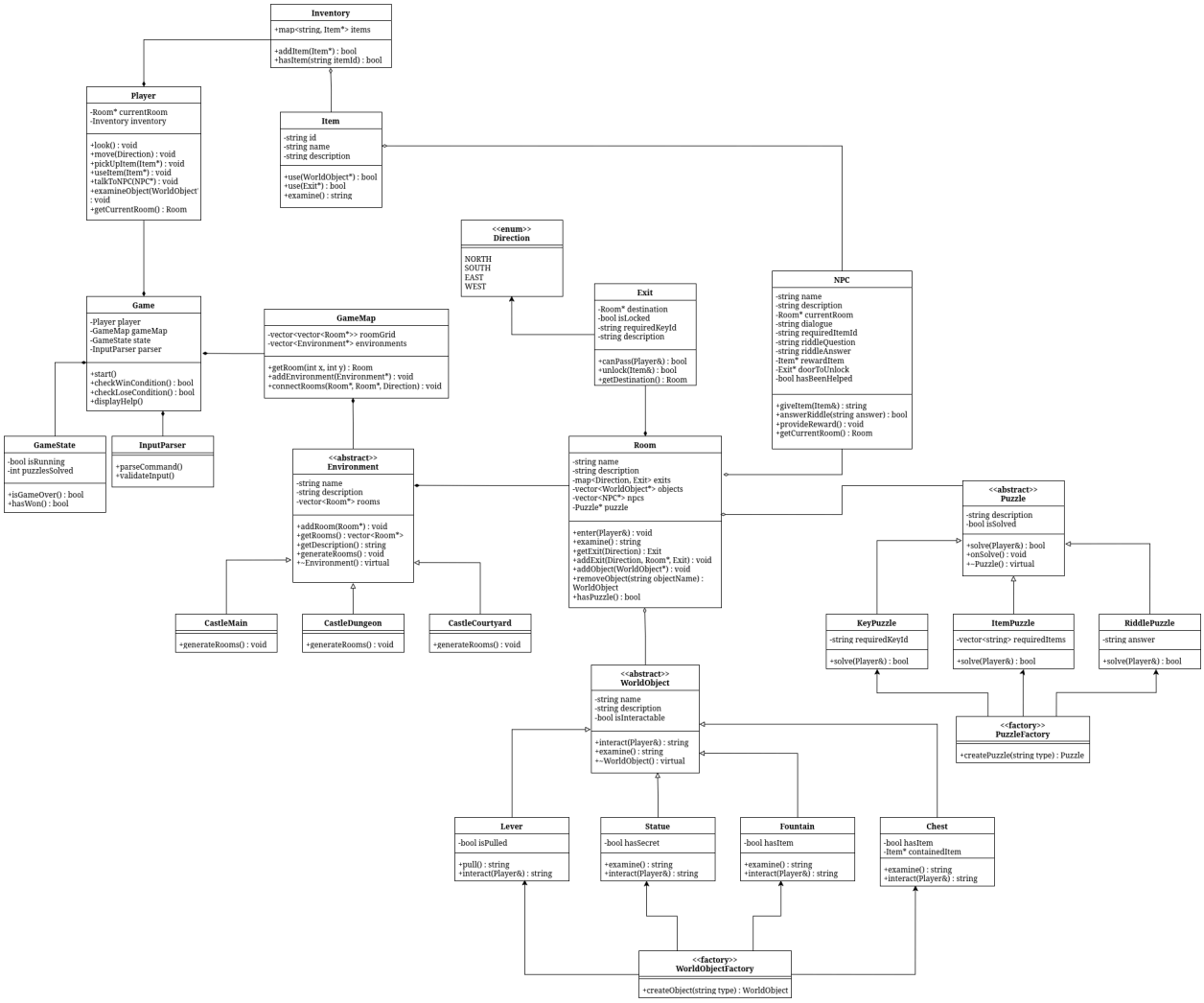
LargeProject6

Liam Jackson, Niko Stranzinger, Nick Tan, Jean Pascua

October 31, 2025

Software Design

Design – Class diagrams



Class Descriptions

Class Name	Method Name	Description
Game	start()	Starts the game and sets everything up.
	checkWinCondition() : bool	Checks if the player has won.
	checkLoseCondition() : bool	Checks if the player has lost.
	displayHelp()	Shows a list of available commands.
InputParser	parseCommand()	Reads what the player types and splits it into words.
	validateInput()	Make sure the command is valid before running it.
Player	look() : void	Shows details about the current room.
	move(Direction) : void	Moves the player in the given direction.
	pickUpItem(Item*) : coid	Picks up an item and adds it to inventory.

	useItem(Item*) : void	Uses an item from the inventory.
	talkToNPC(NPC*) : void	Talks to an NPC in the same room.
	examineObject(WorldObject*) : void	Look closely at the object.
	getCurrentRoom() : Room	Returns the player's current room.
Inventory	addItem(Item*) : bool	Adds an item to the inventory.
	hasItem(string itemId) : bool	Checks if a specific item is in the inventory.
Item	use(WorldObject*) : bool	Uses the item on an object.
	use(Exit*) : bool	Uses the item on the exit (like unlocking a door).
	examine() : string	Describes what the item looks like.
NPC	giveItem(Item&) : string	Gives an item to the player.
	answerRiddle(string answer) : bool	Checks if the player answered the riddle correctly.
	provideReward() : void	Gives a reward after success.
	getCurrentRoom() : Room	Returns the room where the NPC is.
Room	enter(Player&) : void	Lets the player enter the room.
	examine() : string	Describes the room and what's in it.
	getExit(Direction) : Exit	Returns the exit for that direction.
	addExit(Direction, Room*, Exit) : void	Connects one room to another.
	addObject(WorldObject*) : void	Adds an object to the room.
	removeobject(string objectName) : WorldObject	Removes an object from the room.
	hasPuzzle() : bool	Checks if the room has a puzzle.
Environment (Abstract)	addRoom(Room*) : void	Adds a room to the environment.
	getRooms() : vector<Room*>	Returns all rooms in the environment.
	getDescription() : string	Gives a short description of the area.
	generateRooms() : void	Builds and links all rooms.
	~Environment() : virtual	Cleans up memory when destroyed.
CastleMain (Environment)	generateRooms() : void	Create the main castle rooms.
CastleDungeon (Environment)	generateRooms() : void	Creates dungeon rooms.
CastleCourtyard (Environment)	generateRooms() : void	Creates courtyard rooms.
WorldObject (Abstract)	interact(Player&) : string	Lets the player interact with the object.
	examine() : string	Describes the object.

	~WorldObject() : virtual	Cleans up memory when destroyed.
Lever (WorldObject)	examine() : string	Describes the lever.
	interact(Player&) : string	Pulls or toggles the lever.
Statue (WorldObject)	examine() : string	Describes the statue.
	interact(Player&) : string	Lets the player inspect or move the statue.
Fountain (WorldObject)	examine() : string	Describes the fountain.
	interact(Player&) : string	Lets the player interact (drink, drop item, etc.)
Chest (WorldObject)	examine() : string	Describes the chest.
	interact(Player&) : string	Opens or unlock the chest.
WorldObjectFactory (Factory)	createObject(string type) : WorldObject	Makes a new world object based on its type.
Puzzle (Abstract)	solve(Player&) : bool	Tries to solve the puzzle.
	onSolve() : void	Runs when the puzzle is solved.
	~Puzzle() : virtual	Cleans up memory when destroyed.
KeyPuzzle (Puzzle)	solve(Player&) : bool	Solves the puzzle by using a key.
ItemPuzzle (Puzzle)	solve(Player&) : bool	Solves the puzzle by using the right item.
RiddlePuzzle (Puzzle)	solve(Player&) : bool	Solves the puzzle by answering the riddle.
PuzzleFactory (Factory)	createPuzzle(string type) : Puzzle	Makes a new puzzle of the given type.
Exit	canPass(Player&) : bool	Checks if the player can go through the exit.
	unlock(Item&) : bool	Unlocks the exit with an item.
	getDestination() : Room	Returns the room on the other side of the exit.
Direction (Enum)	NORTH, SOUTH, EAST, WEST	Lists possible directions to move.