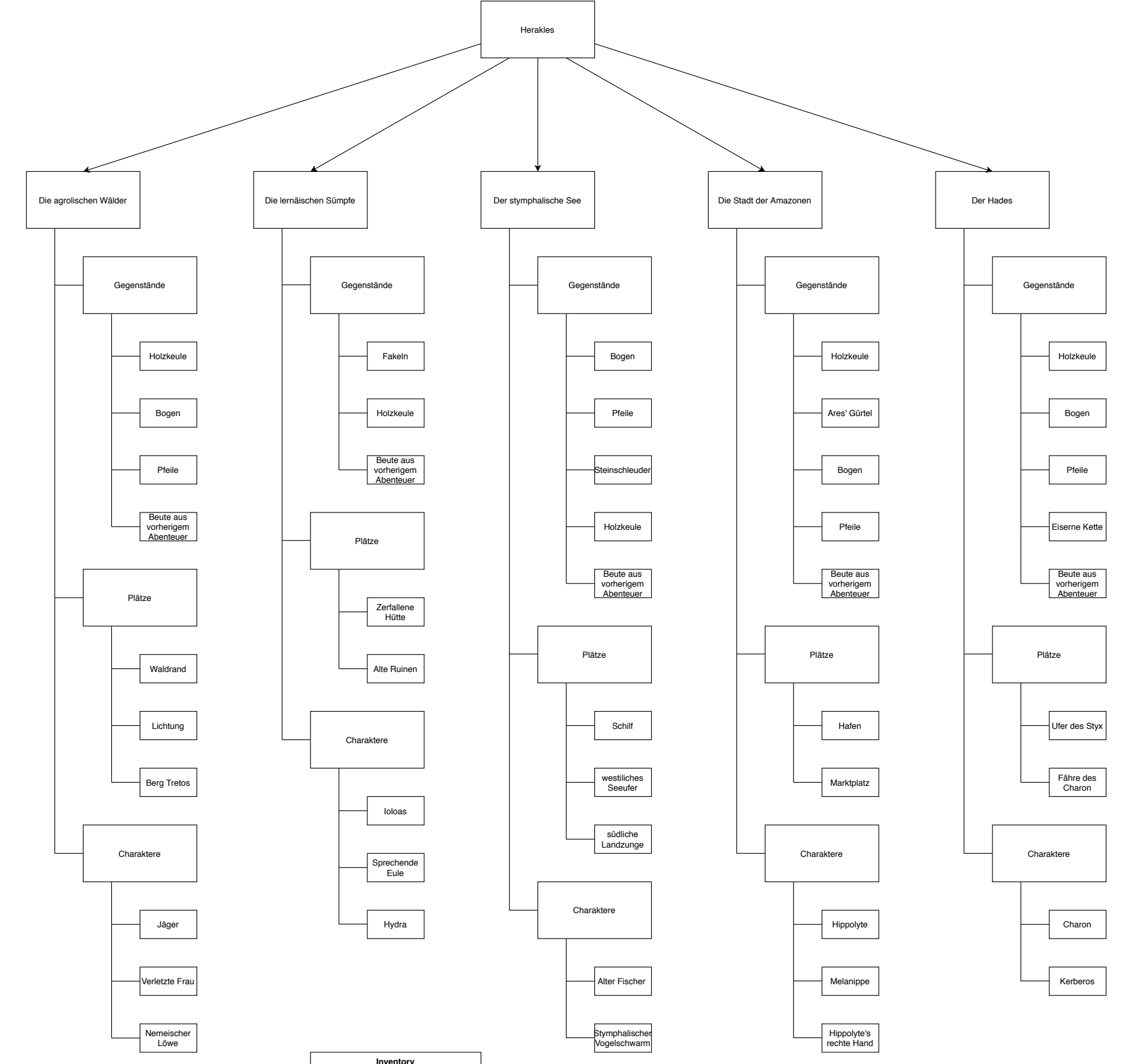
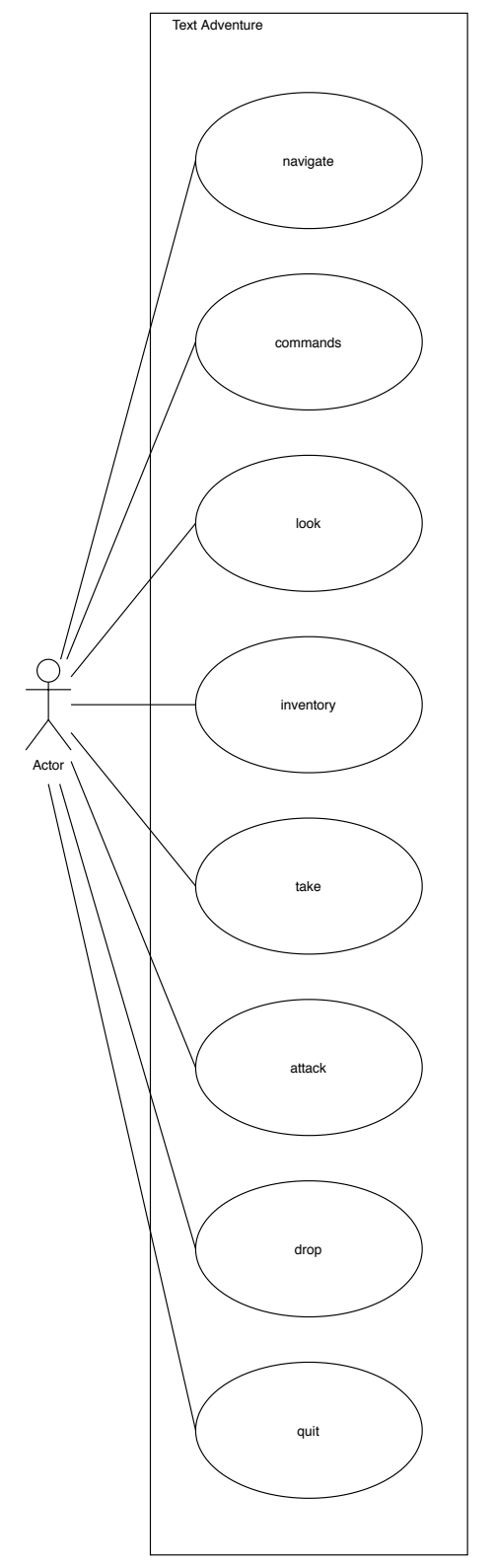
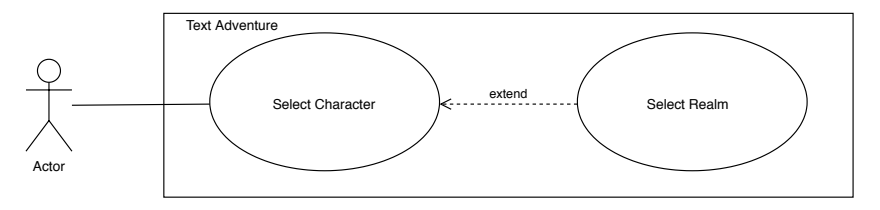
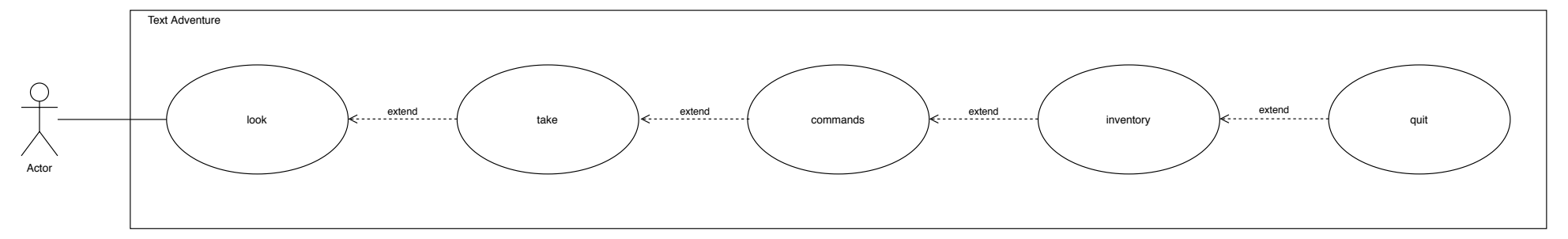
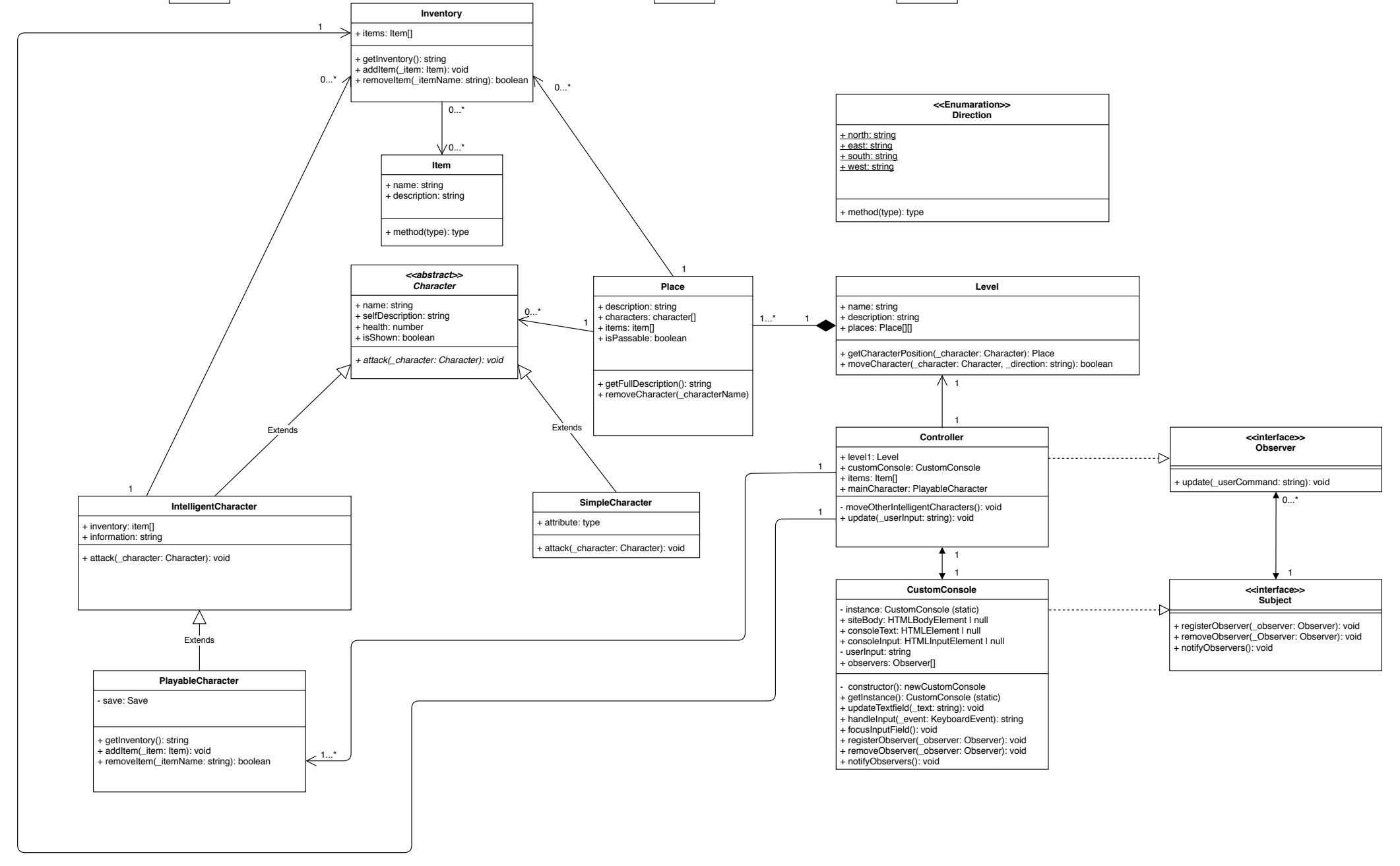
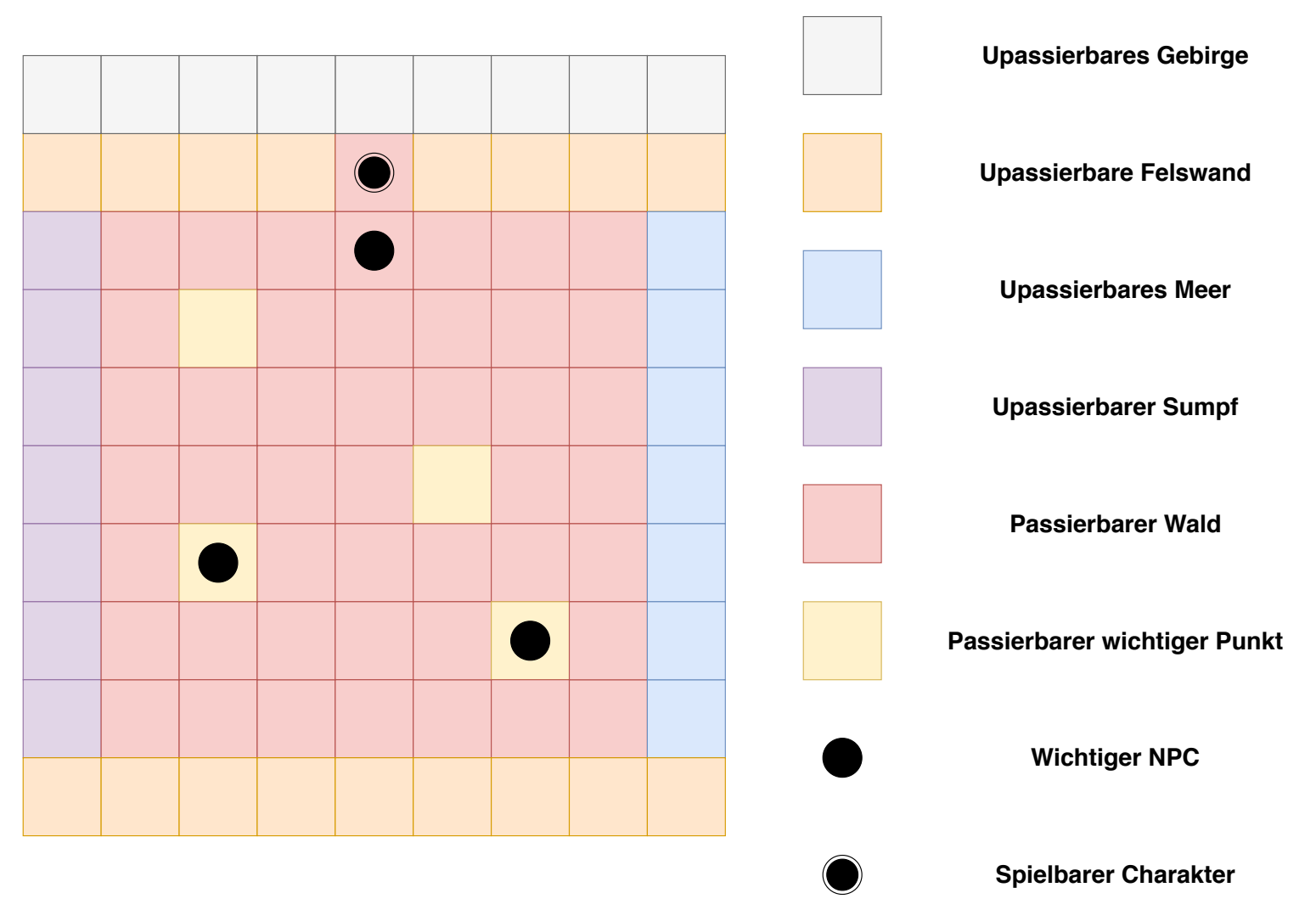






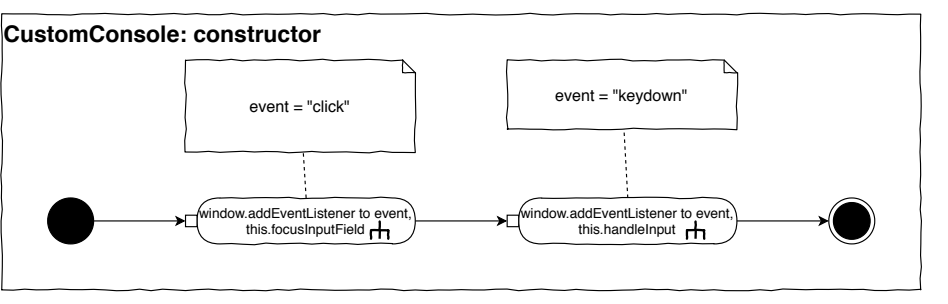



Inventory
+ items: Item[]
+ getInventory(): string
+ addItem(_item: Item): void
+ removeItem(_itemName: string): boolean
Item
+ name: string
+ description: string
+ method(type): type
<<Enumaration>>
Direction
+ north: string
+ east: string
+ south: string
+ west: string
+ method(type): type
<<abstract>>
Character
+ name: string
+ selfDescription: string
+ health: number
+ isShown: boolean
+ attack(_character: Character): void
Place
+ description: string
+ characters: character[]
+ items: item[]
+ isPassable: boolean
+ getFullDescription(): string
+ removeCharacter(_characterName)
Level
+ name: string
+ description: string
+ places: Place[][]
+ getCharacterPosition(_character: Character): Place
+ moveCharacter(_character: Character, _direction: string): boolean
Controller
+ level1: Level
+ customConsole: CustomConsole
+ items: Item[]
+ mainCharacter: PlayableCharacter
- moveOtherIntelligentCharacters(): void
+ update(_userInput: string): void
<<interface>>
Observer
+ update(_userCommand: string): void
CustomConsole
- instance: CustomConsole (static)
+ siteBody: HTMLBodyElement | null
+ consoleText: HTMLElement | null
+ consoleInput: HTMLInputElement | null
- userInput: string
+ observers: Observer[]
- constructor(): newCustomConsole
+ getInstance(): CustomConsole (static)
+ updateTextfield(_text: string): void
+ handleInput(_event: KeyboardEvent): string
+ focusInputField(): void
+ registerObserver(_observer: Observer): void
+ removeObserver(_observer: Observer): void
+ notifyObservers(): void
<<interface>>
Subject
+ registerObserver(_observer: Observer): void
+ removeObserver(_Observer: Observer): void
+ notifyObservers(): void
IntelligentCharacter
+ inventory: item[]
+ information: string
+ attack(_character: Character): void
SimpleCharacter
+ attribute: type
+ attack(_character: Character): void
PlayableCharacter
- save: Save
+ getInventory(): string
+ addItem(_item: Item): void
+ removeItem(_itemName: string): boolean
Extends
Extends
Extends
1
0...*
0...*
0...*
0...*
1
1...*
1
1
1
1
1
1
0...*
1
1...*

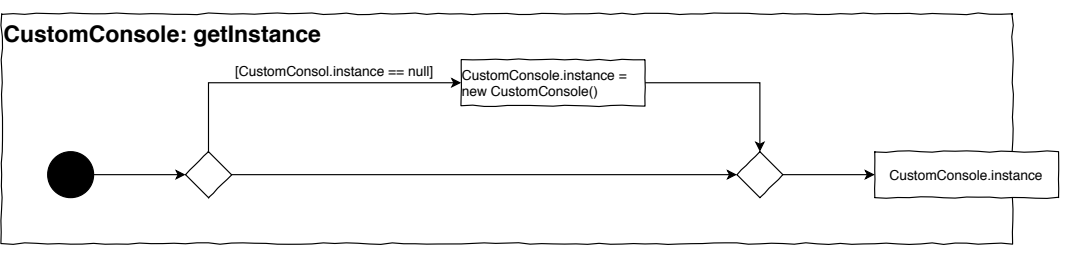


CustomConsole

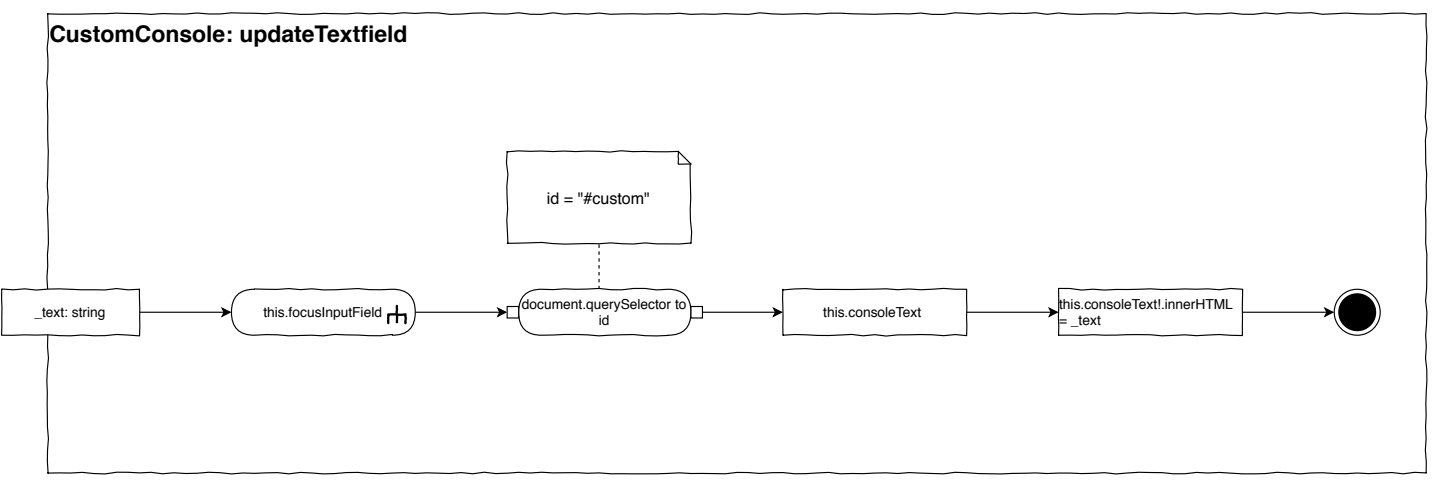
CustomConsole: constructor


event = "click"
event = "keydown"
window.addEventListener to event, this.focusInputField
window.addEventListener to event, this.handleInput

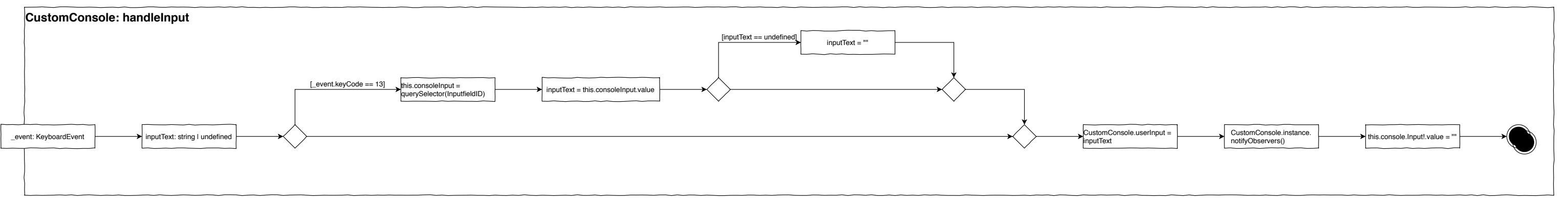
CustomConsole: getInstance


[CustomConsol.instance == null]
CustomConsole.instance = new CustomConsole()
CustomConsole.instance

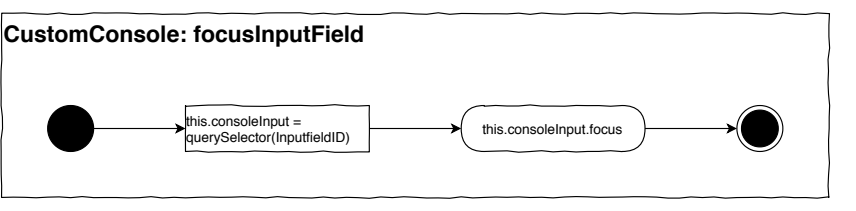
CustomConsole: updateTextfield


id = "#custom"
_text: string
this.focusInputField
document.querySelector to id
this.consoleText
this.consoleText!.innerHTML = _text

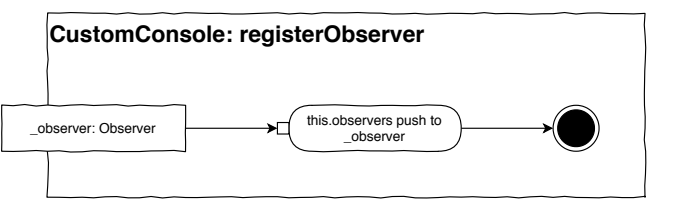
CustomConsole: handleInput


_event: KeyboardEvent
inputText: string | undefined
[_event.keyCode == 13]
this.consoleInput = querySelector(InputfieldID)
inputText = this.consoleInput.value
[inputText == undefined]
inputText = ""
CustomConsole.userInput = inputText
CustomConsole.instance. notifyObservers()
this.console.Input!.value = ""

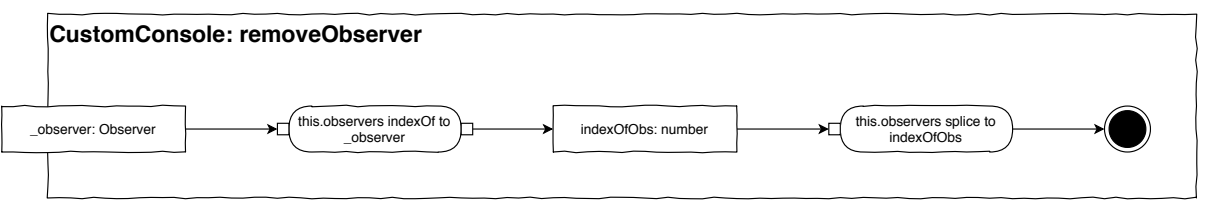
CustomConsole: focusInputField



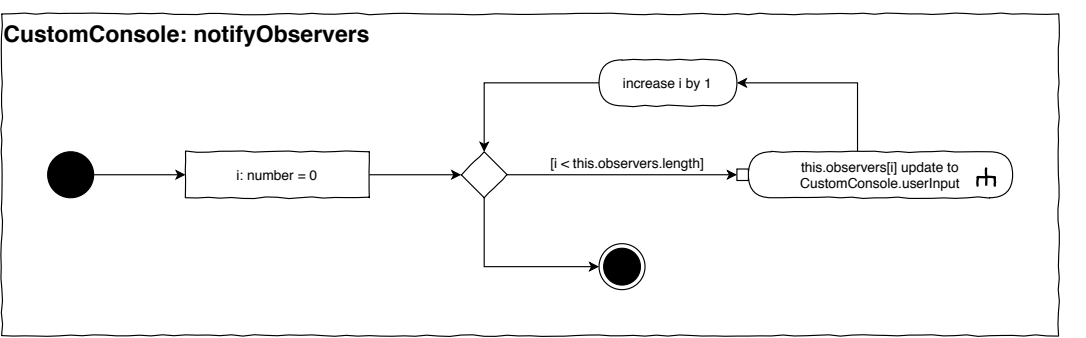
CustomConsole: registerObserver

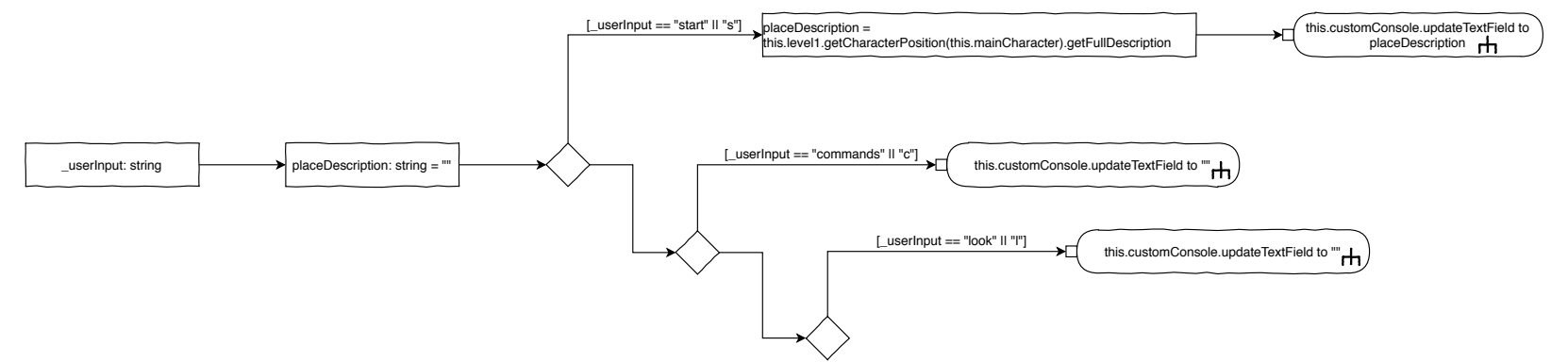


CustomConsole: removeObserver


_observer: Observer
this.observers indexOf to _observer
indexOfObs: number
this.observers splice to indexOfObs

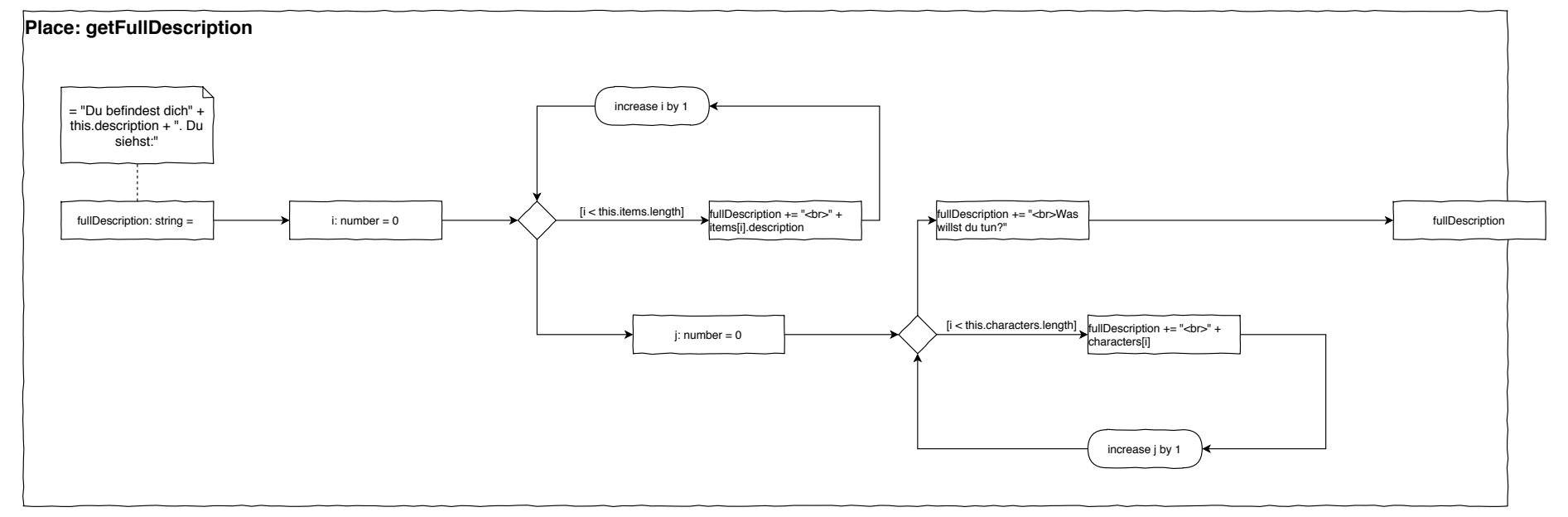
CustomConsole: notifyObservers


i: number = 0
increase i by 1
[i < this.observers.length]
this.observers[i] update to CustomConsole.userInput

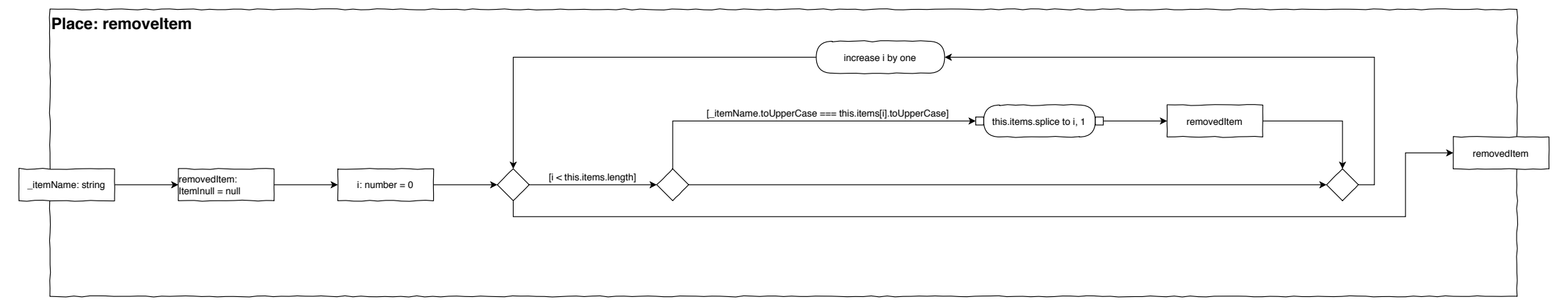

_userInput: string
placeDescription: string = ""
[_userInput == "start" || "s"]
placeDescription = this.level1.getCharacterPosition(this.mainCharacter).getFullDescription
this.customConsole.updateTextField to placeDescription
[_userInput == "commands" || "c"]
this.customConsole.updateTextField to ""
[_userInput == "look" || "l"]
this.customConsole.updateTextField to ""

Place

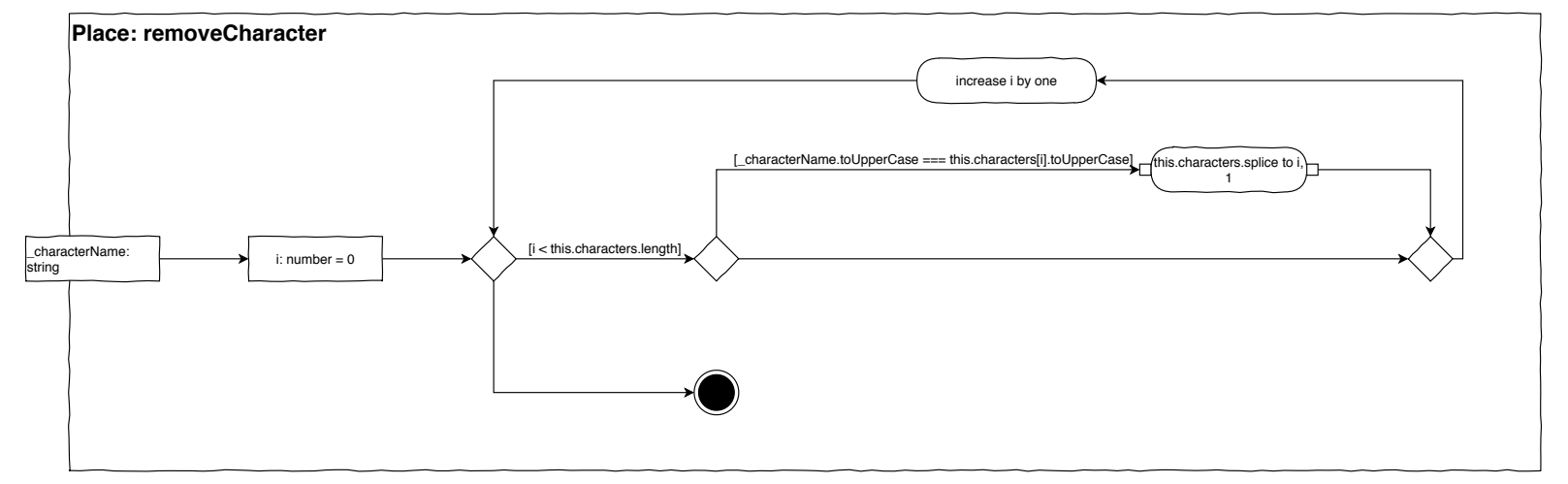
Place: getFullDescription


= "Du befindest dich" + this.description + ". Du siehst:"
fullDescription: string =
i: number = 0
increase i by 1
[i < this.items.length]
fullDescription += "<br>" + items[i].description
j: number = 0
[i < this.characters.length]
fullDescription += "<br>" + characters[i]
increase j by 1
fullDescription += "<br>Was willst du tun?"
fullDescription

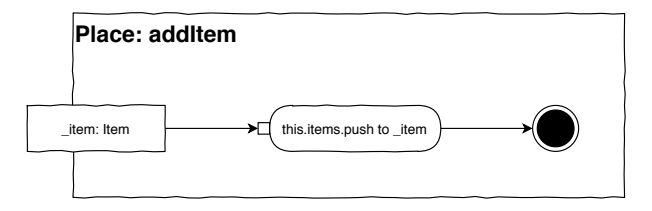
Place: removeItem


_itemName: string
removedItem: Item|null = null
i: number = 0
[i < this.items.length]
[_itemName.toUpperCase === this.items[i].toUpperCase]
this.items.splice to i, 1
removedItem
increase i by one
removedItem

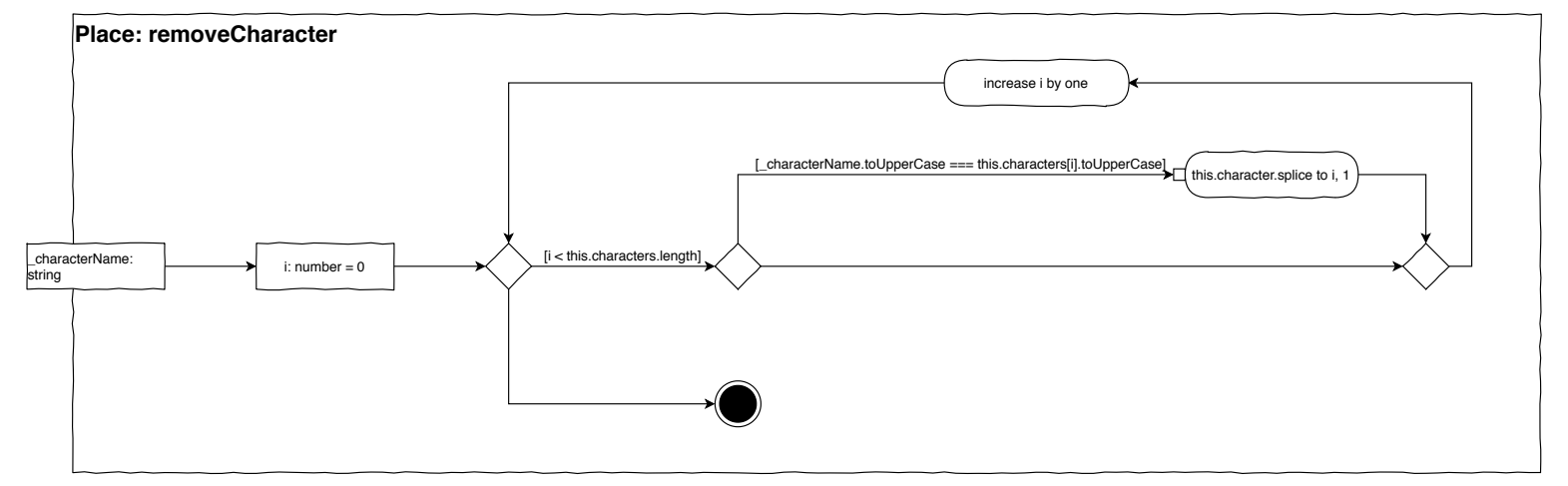
Place: removeCharacter


_characterName: string
i: number = 0
[i < this.characters.length]
[_characterName.toUpperCase === this.characters[i].toUpperCase]
this.characters.splice to i, 1
increase i by one

Place: addItem

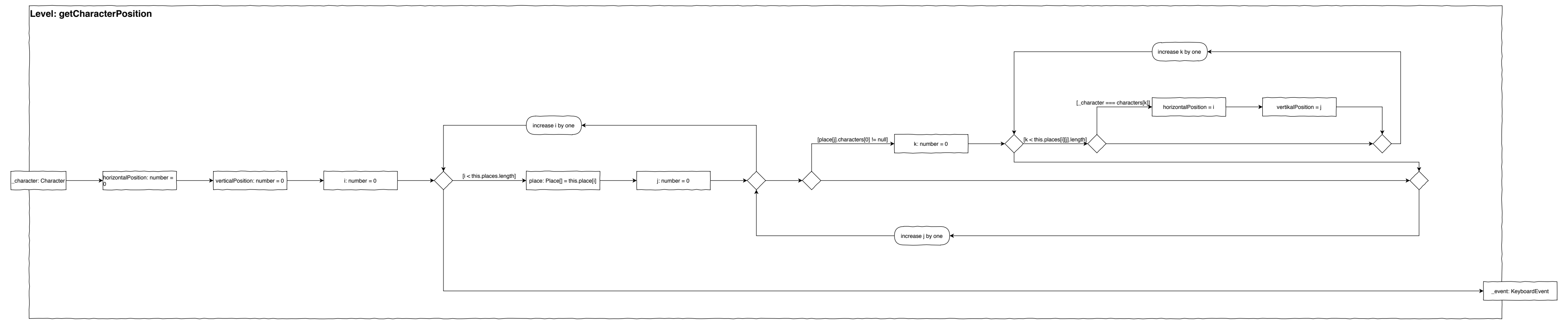


Place: removeCharacter

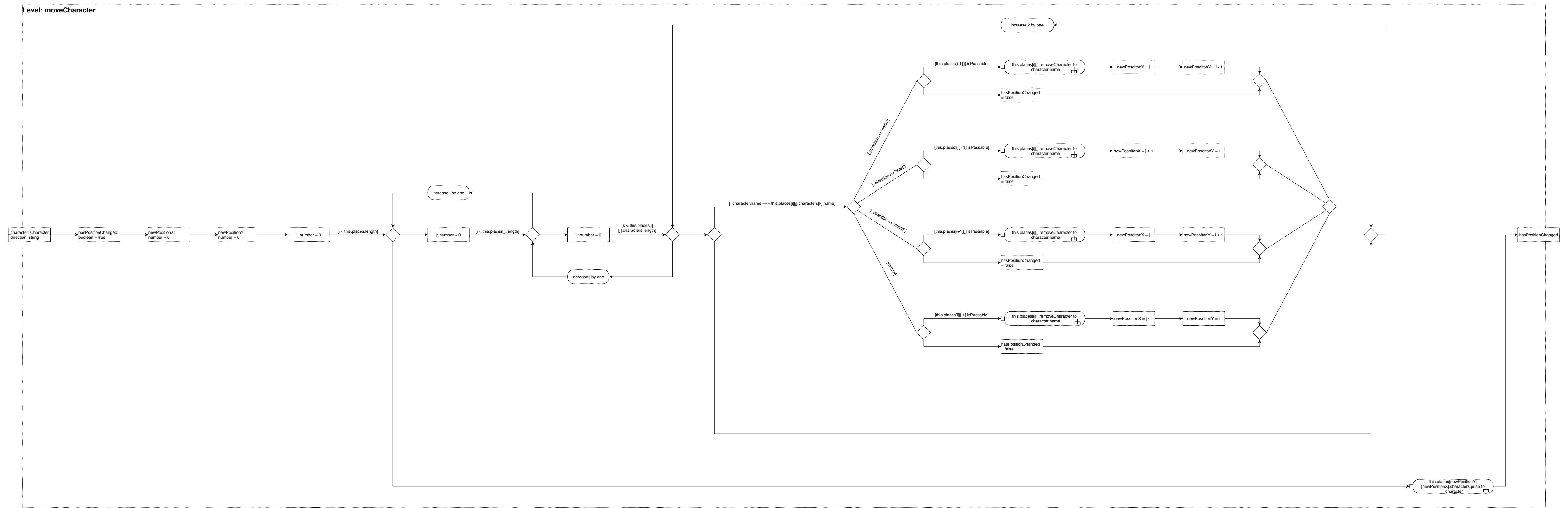

_characterName: string
i: number = 0
[i < this.characters.length]
[_characterName.toUpperCase === this.characters[i].toUpperCase]
this.character.splice to i, 1
increase i by one

Level

Level: getCharacterPosition

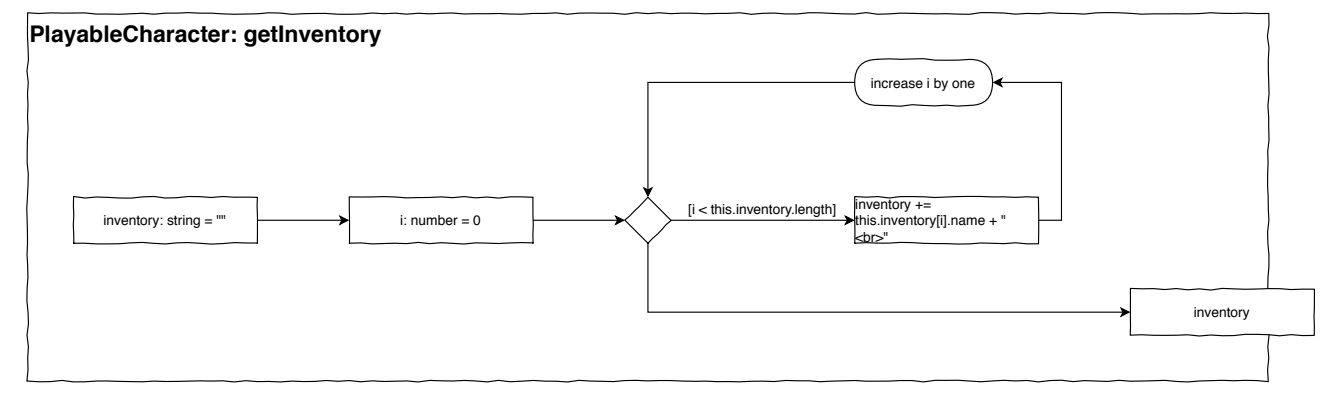


Level: moveCharacter

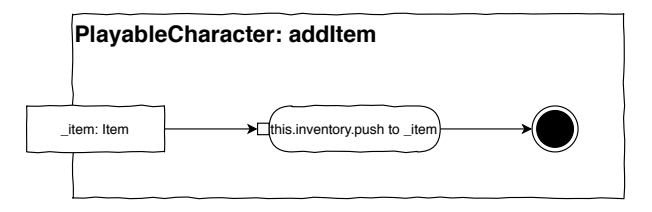

_character: Character, _direction: string
hasPositionChanged: boolean = true
newPositionX: number = 0
newPositionY: number = 0
i: number = 0
[i < this.places.length]
j: number = 0
[j < this.places[i].length]
k: number = 0
[k < this.places[i][j].characters.length]
increase i by one
increase j by one
increase k by one
[_character.name === this.places[i][j].characters[k].name]
[_direction == "north"]
[_direction == "east"]
[_direction == "south"]
[default]
[this.places[i-1][j].isPassable]
[this.places[i][j+1].isPassable]
[this.places[i+1][j].isPassable]
[this.places[i][j-1].isPassable]
this.places[i][j].removeCharacter to _character.name
hasPositionChanged = false
newPosoitonX = j
newPosoitonY = i - 1
newPosoitonX = j + 1
newPosoitonY = i
newPosoitonY = i + 1
newPosoitonX = j - 1
this.places[newPositionY][newPositionX].characters.push to character
hasPositionChanged

PlayableCharacter

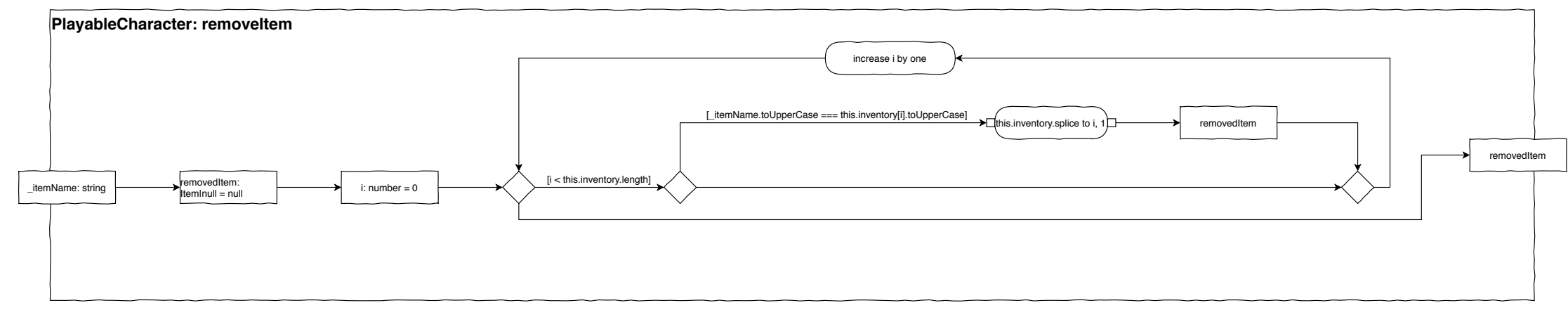
PlayableCharacter: getInventory


inventory: string = ""
i: number = 0
[i < this.inventory.length]
inventory += this.inventory[i].name + "<br>"
increase i by one
inventory

PlayableCharacter: addItem



PlayableCharacter: removeItem


_itemName: string
removedItem: Item|null = null
i: number = 0
[i < this.inventory.length]
[_itemName.toUpperCase === this.inventory[i].toUpperCase]
this.inventory.splice to i, 1
removedItem
increase i by one
removedItem