

Program
Main: void

Player
- health: int - inventory: List<Item>
+ getInventory(): List<Item> + showInventory(): void + getHealth(): int + reduceHealth(): void + addItem(Item): void + removeItem(Item): void + playerDropItem(String): Item

NPC
- name: String - fightable: bool - description: String - answer: String - health: int - currentRoom: Room - inventory: List<Item>
+ NPC(...) + changeCurrentRoom(Room): void + getCurrentRoom(): Room + getName(): String + getFightable(): bool + getDescription(): String + getAnswer(): String + getHealth(): int + reduceHealth(): void + getInventory(): List<Item> + addItem(Item): void + removeItem(Item): void +

Item
<ul style="list-style-type: none"> + Name: String - Useable: bool description: String - description: String
<ul style="list-style-type: none"> + Item() + getName(): String + getUseable(): bool + getDescription(): String

Room
<ul style="list-style-type: none"> - roomName: String - roomDescription: String - doors: List<Door> - inventory: List<Item> - npcs: List<NPC>
<ul style="list-style-type: none"> + Room() + addDoor(Door): void + getDoors(): List<Door> + getNPCs(): List<NPC> + addNPC(NPC): void + removeNPC(NPC): void + getInventory(): List<Item> + addItem(Item): void + removeItem(Item): void + getTitle(): String + getDescription(): String

Door
<ul style="list-style-type: none"> + Directions: enum + moveShortcuts: String - direction: Directions - directionName: String - locked: bool
<ul style="list-style-type: none"> + Door() + getLocked(): bool + setLocked(): void + setDirection(Directions): void + getDirection(): Directions + setLeadsTo(Room): void + getLeadsTo(): Room + getDirectionName(): String

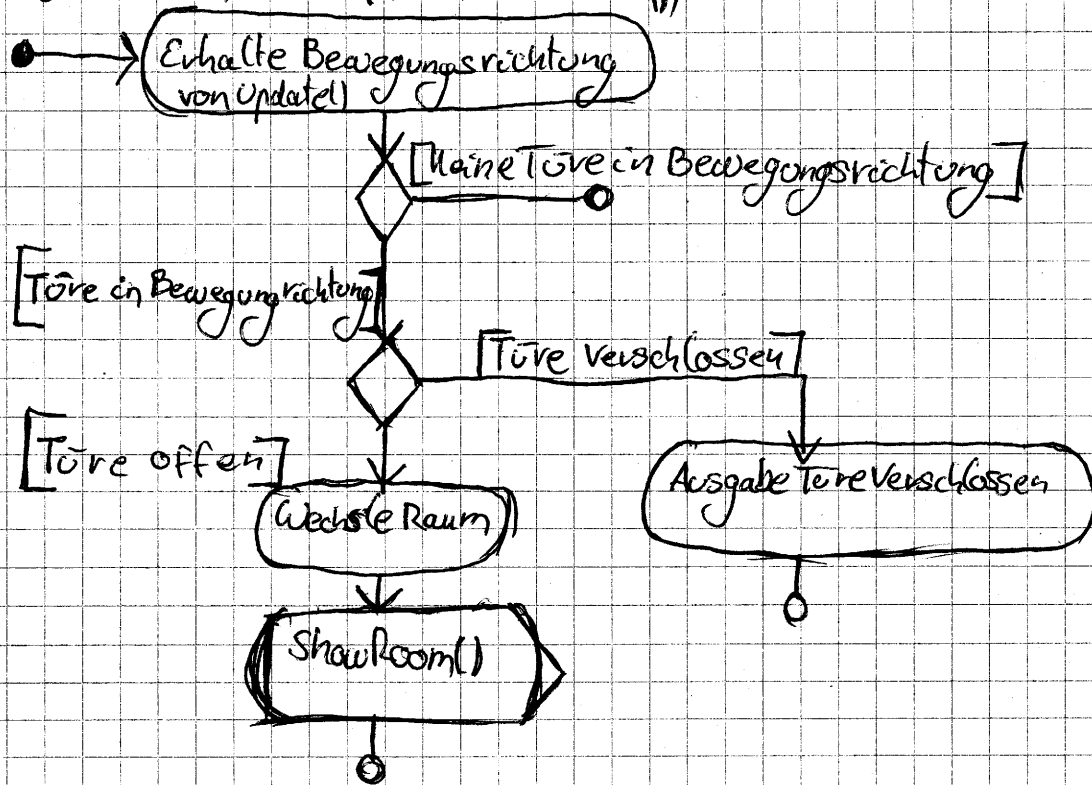
Main Game

- currentRoom : Room
- lastRoom : Room
- Secret Passage : Room
- currentNpc : Npc
- currentEnemy
- ~~player : Player~~
- player : Player
- allRooms : List<Room>
- allNpcs : List<Npc>
- running : bool
- gameOver : bool
- turnCounter : int
- caseSolved : bool

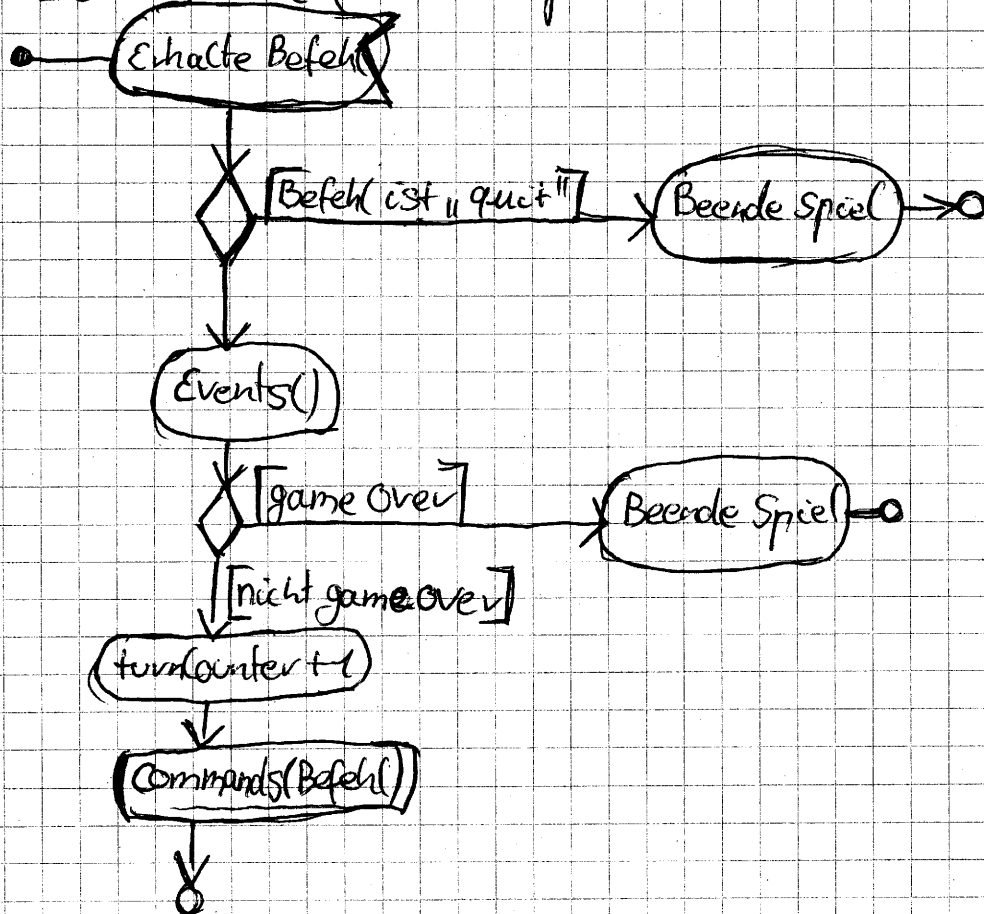
+ MainGame()

- ~~player : Player~~
- + Commands(string) : void
- + move(string) : void
- + moveNpc(string, string) : void
- ~~player : Player~~
- + Events() : void
- + Update() : void
- + Drop(string) : void
- + Attack(string) : void
- + LookAt(string) : void
- + TalkTo(string) : void
- + ShowRoom() : void
- + Take(string) : void
- + NpcGiveItem(Item, Npc) : void
- + PlayerGiveItem(Item, Npc) : void

Klasse MainGame (Methode Move())



Klasse MainGame (Methode Update())



MainGame (Methode Attack())

• → Erhalte Befehl

~~Ans~~ [Angriffsziel existiert nicht] →

[Angriffsziel existiert]

[Ziel nicht Angreifbar] →

[Ziel ist Angreifbar]

[Kämpfer leben > 0]

Angriffsweite holen

[Angriffsweite Spieler > Aus NPC]

Npc.ReduceHealth()

[Kampf nicht fortsetzen] → [Setze Spieler in vorherigen Raum]

[Kampf fortsetzen]

[Aus S < Aus NPC]

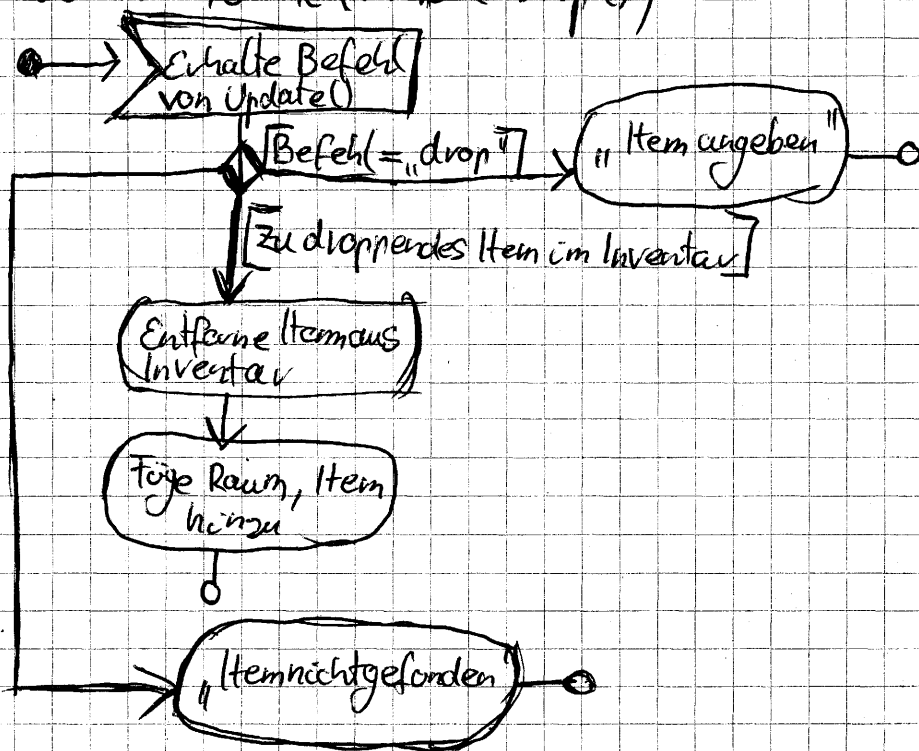
Spieler.ReduceHealth()

[Aus S > Aus NPC] → [Spieler Health() <= 0] → Beende Spiel

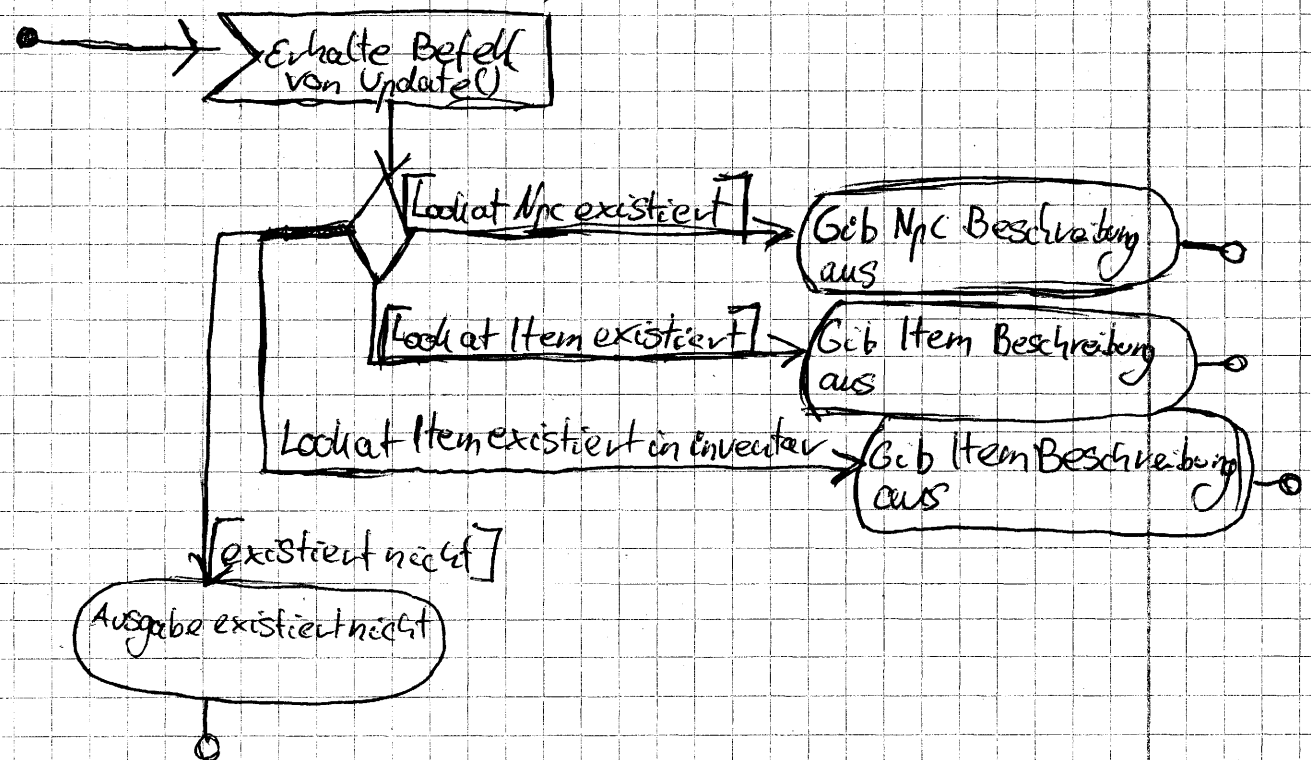
[NPC Health() <= 0] → Lösche NPC aus Raum
Füge NPC Leiche hinzu

[Kämpfer leben <= 0]

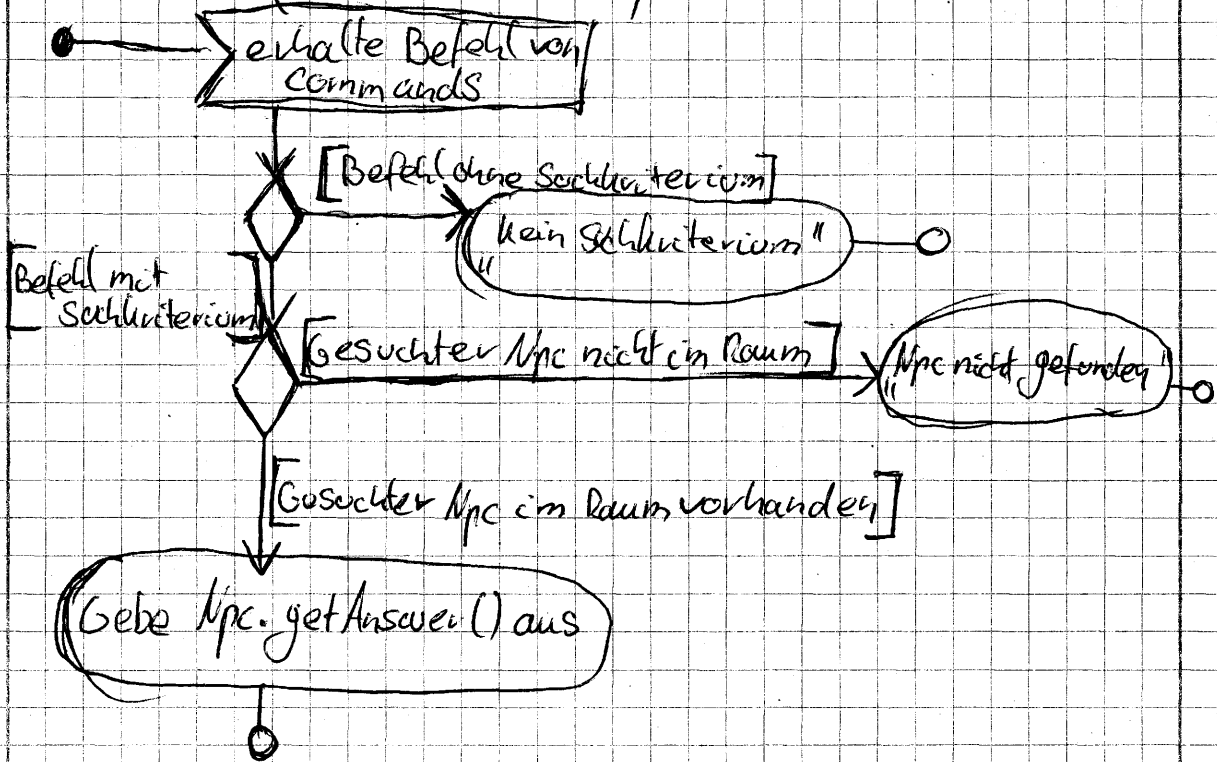
Klasse MainGame (Methode Drop())



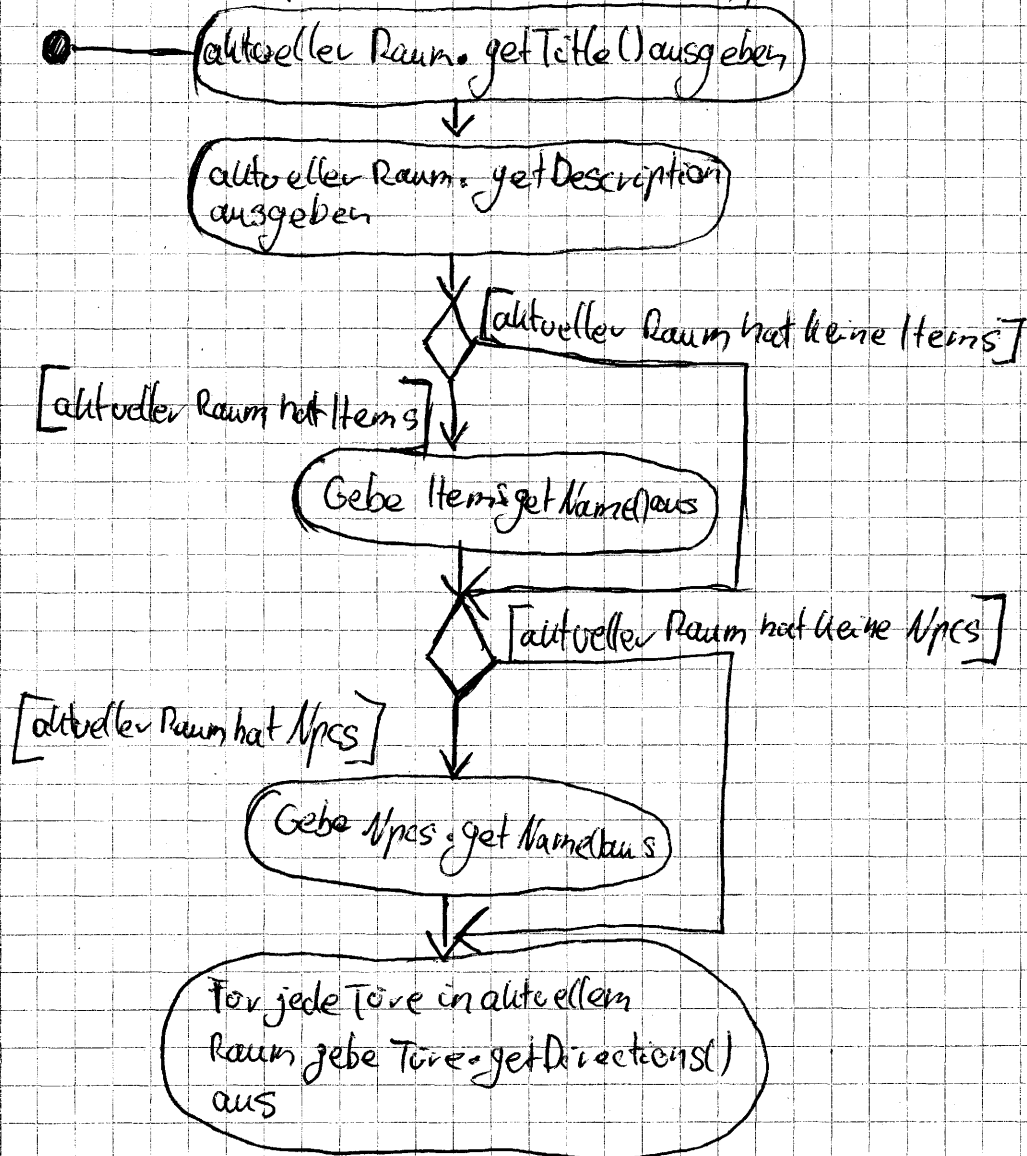
Klasse MainGame (Look at ())



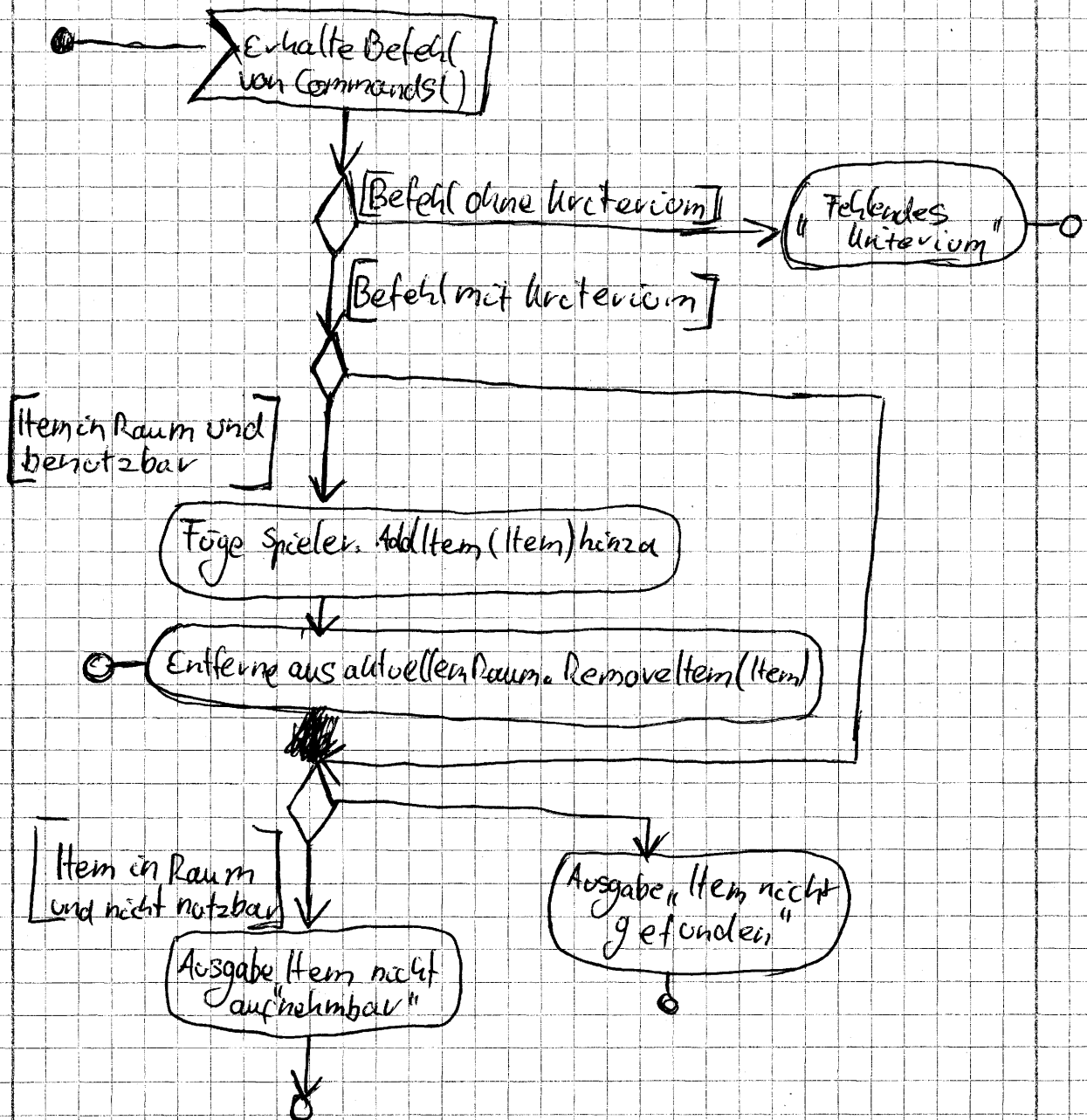
Main Game (Methode TalkTo())



Main Game (Methode Show Room())



Main Game (Methode Take())



Player (Methode ShowInventory())

