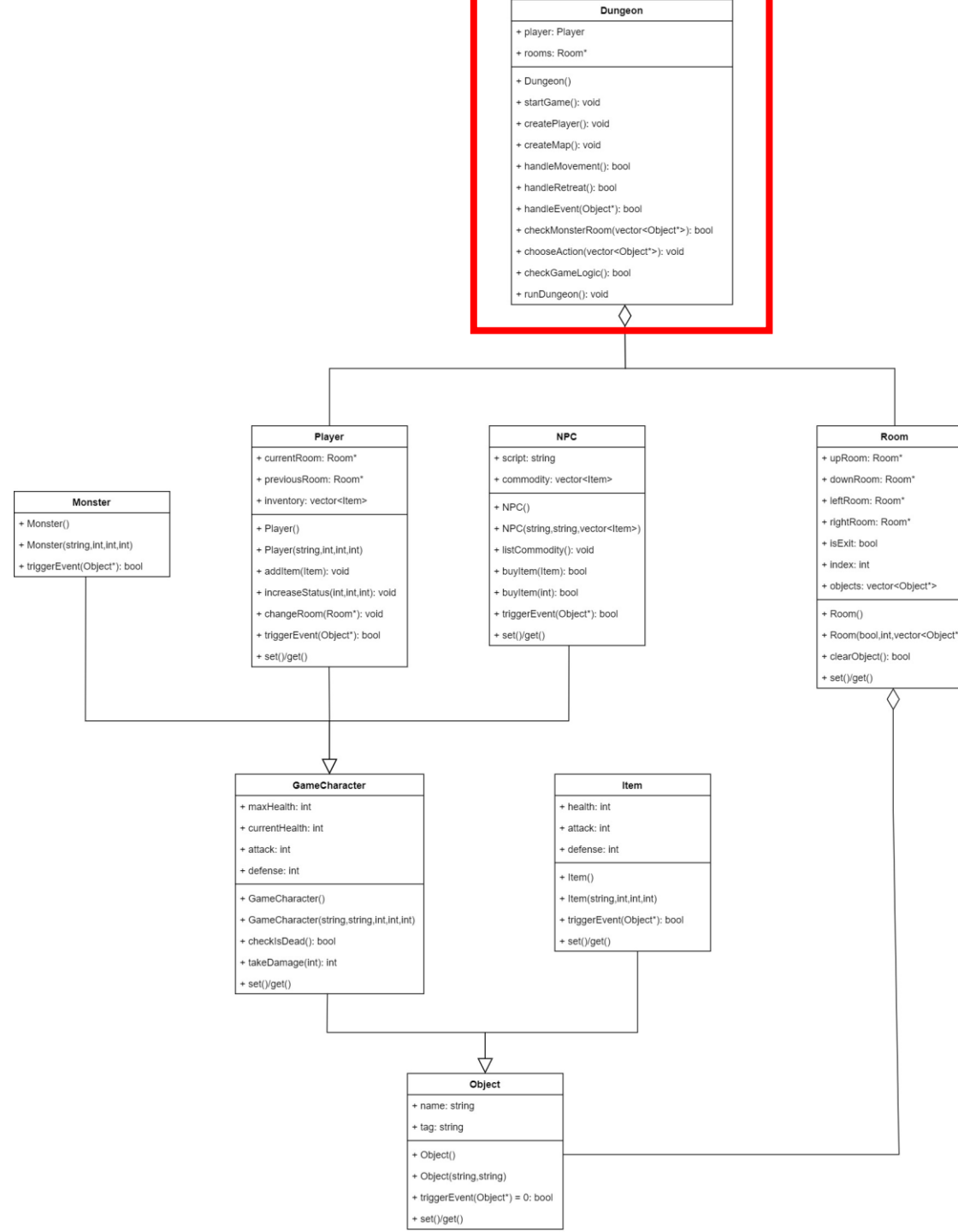# Dungeon

Dungeon

+ player: Player
+ rooms: Room*

+ Dungeon()
+ startGame(): void
+ createPlayer(): void
+ createMap(): void
+ handleMovement(): bool
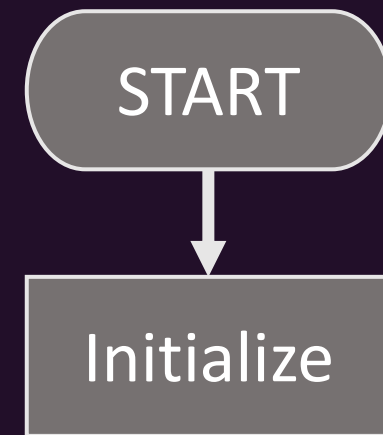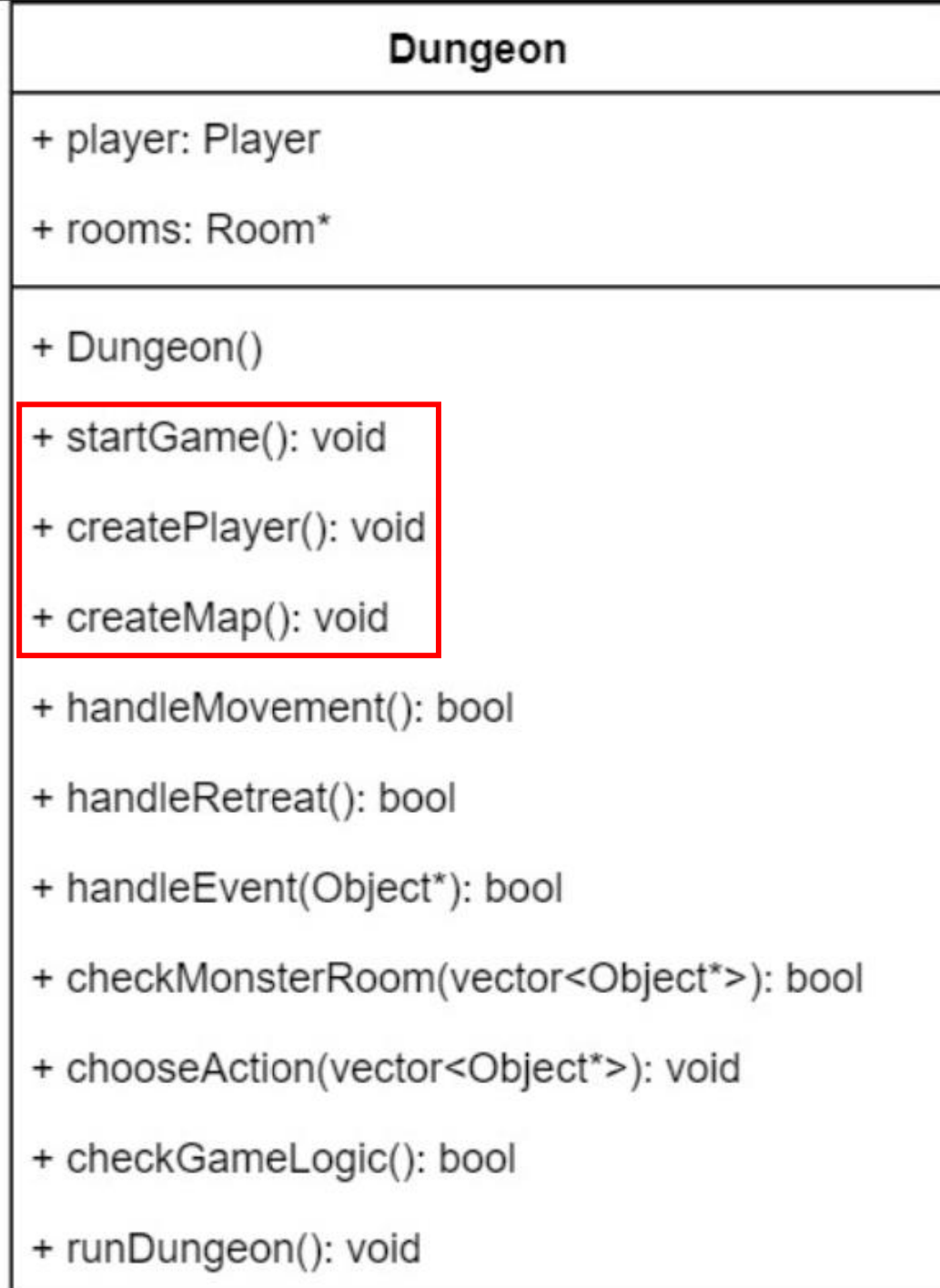+ handleRetreat(): bool
+ handleEvent(Object*): bool
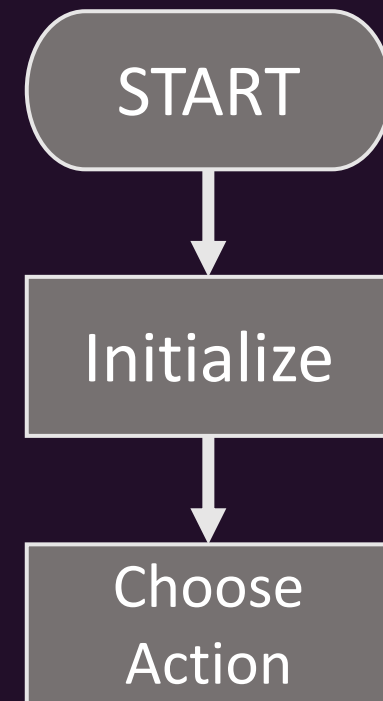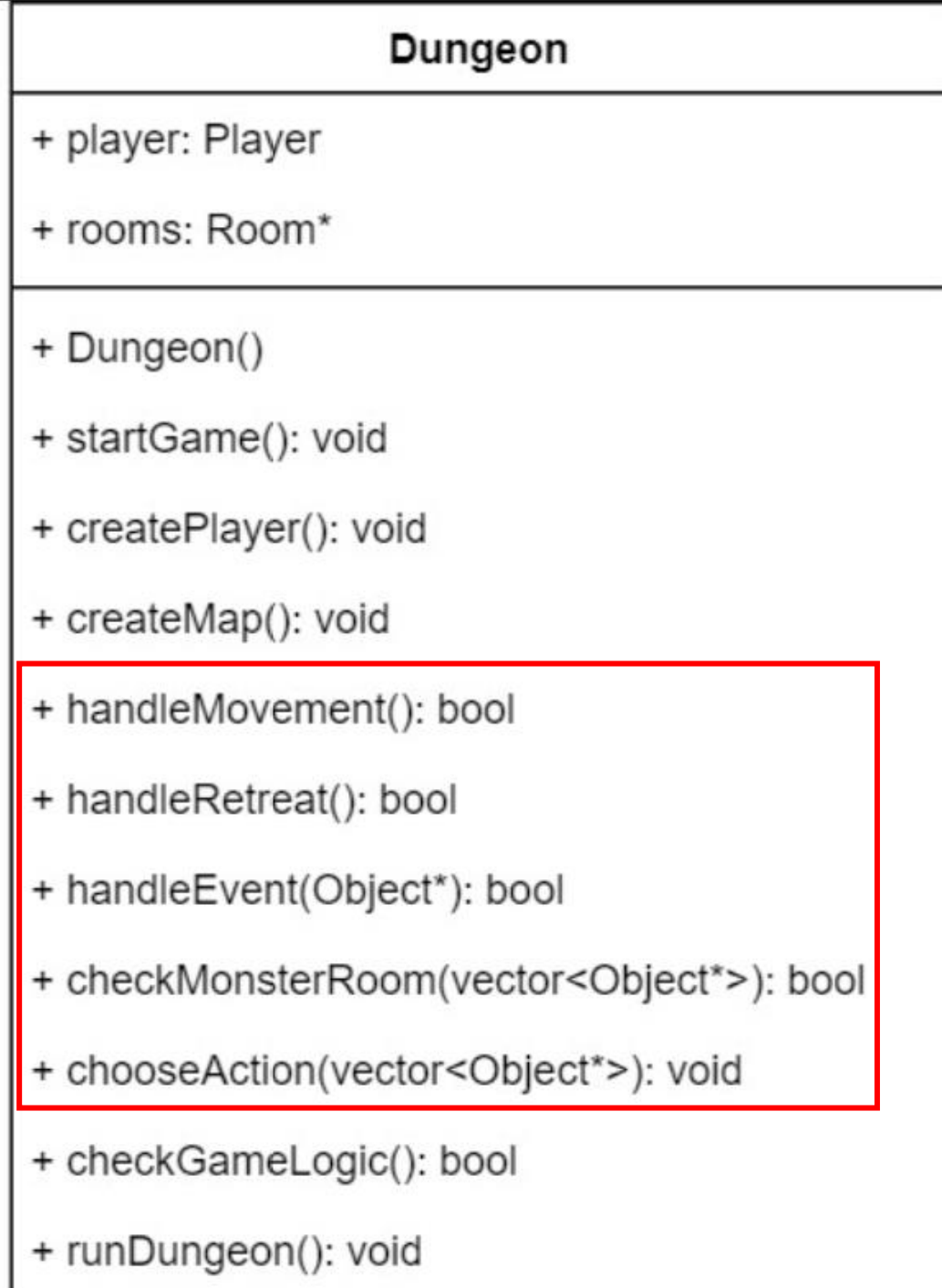+ checkMonsterRoom(vector<Object*>): bool
+ chooseAction(vector<Object*>): void
+ checkGameLogic(): bool
+ runDungeon(): void

Player

+ currentRoom: Room*
+ previousRoom: Room*
+ inventory: vector<Item>

+ Player()
+ Player(string,int,int,int)
+ addItem(Item): void
+ increaseStatus(int,int,int): void
+ changeRoom(Room*): void
+ triggerEvent(Object*): bool
+ set()/get()

NPC

+ script: string
+ commodity: vector<Item>

+ NPC()
+ NPC(string,string,vector<Item>)
+ listCommodity(): void
+ buyItem(Item): bool
+ buyItem(int): bool
+ triggerEvent(Object*): bool
+ set()/get()

Room

+ upRoom: Room*
+ downRoom: Room*
+ leftRoom: Room*
+ rightRoom: Room*
+ isExit: bool
+ index: int
+ objects: vector<Object*>

+ Room()
+ Room(bool,int,vector<Object*>)
+ clearObject(): bool
+ set()/get()

Monster

+ Monster()
+ Monster(string,int,int,int)
+ triggerEvent(Object*): bool

GameCharacter

+ maxHealth: int
+ currentHealth: int
+ attack: int
+ defense: int

+ GameCharacter()
+ GameCharacter(string,string,int,int,int)
+ checkIsDead(): bool
+ takeDamage(int): int
+ set()/get()

Item

+ health: int
+ attack: int
+ defense: int

+ Item()
+ Item(string,int,int,int)
+ triggerEvent(Object*): bool
+ set()/get()

Object

+ name: string
+ tag: string

+ Object()
+ Object(string,string)
+ triggerEvent(Object*) = 0: bool
+ set()/get()

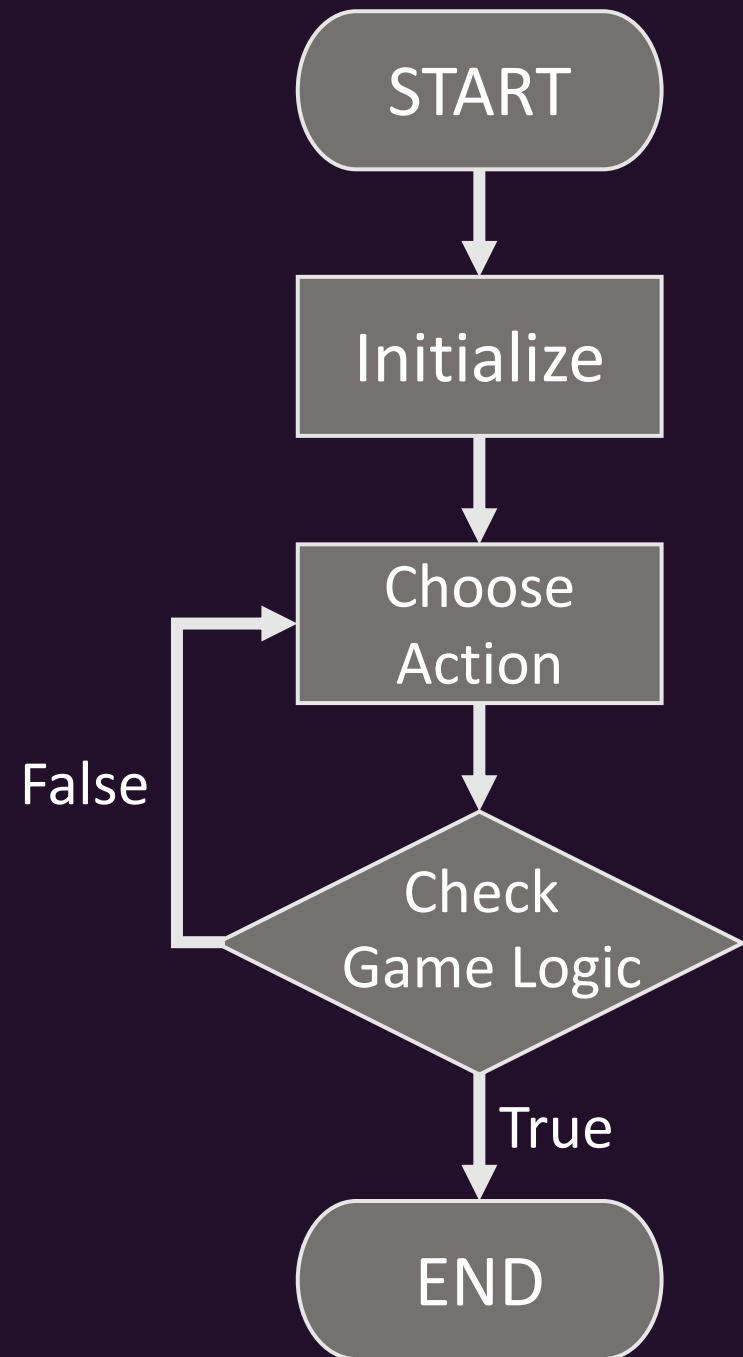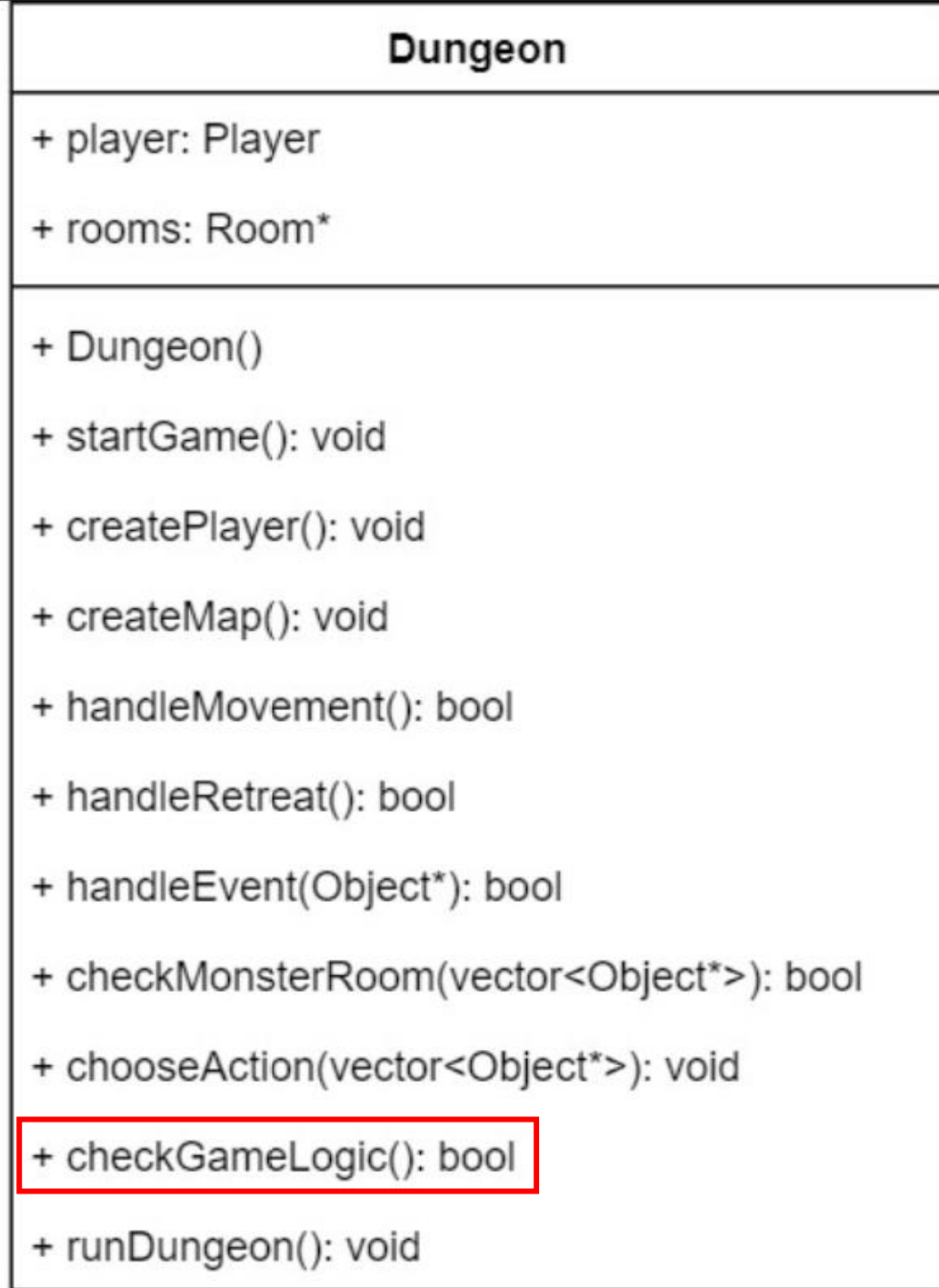| **Dungeon** |
|---|
| + player: Player |
| + rooms: Room* |
|---|
| + Dungeon() |
| + startGame(): void |
| + createPlayer(): void |
| + createMap(): void |
| + handleMovement(): bool |
| + handleRetreat(): bool |
| + handleEvent(Object*): bool |
| + checkMonsterRoom(vector<Object*>): bool |
| + chooseAction(vector<Object*>): void |
| + checkGameLogic(): bool |
| + runDungeon(): void |

START

Dungeon

+ player: Player

+ rooms: Room*

+ Dungeon()

+ startGame(): void

+ createPlayer(): void

+ createMap(): void

+ handleMovement(): bool

+ handleRetreat(): bool

+ handleEvent(Object*): bool

+ checkMonsterRoom(vector<Object*>): bool

+ chooseAction(vector<Object*>): void

+ checkGameLogic(): bool

+ runDungeon(): void

START

Initialize

Dungeon

+ player: Player

+ rooms: Room*

+ Dungeon()

+ startGame(): void

+ createPlayer(): void

+ createMap(): void

+ handleMovement(): bool

+ handleRetreat(): bool

+ handleEvent(Object*): bool

+ checkMonsterRoom(vector<Object*>): bool

+ chooseAction(vector<Object*>): void

+ checkGameLogic(): bool

+ runDungeon(): void

START

Initialize

Choose Action

Dungeon
+ player: Player
+ rooms: Room*
+ Dungeon()
+ startGame(): void
+ createPlayer(): void
+ createMap(): void
+ handleMovement(): bool
+ handleRetreat(): bool
+ handleEvent(Object*): bool
+ checkMonsterRoom(vector<Object*>): bool
+ chooseAction(vector<Object*>): void
+ checkGameLogic(): bool
+ runDungeon(): void

START

Initialize

Choose Action

Check Game Logic

False

True

END

Game Initialization

- **Generate Map**

  - Create rooms and objects in the room

  - Link each room to complete the map

- **Generate Player**

  - Decide player's name

  - Create player with basic status
    (name, health, attack, defense)

START

Initialize

Choose
Action

False

Check
Game Logic

True

END
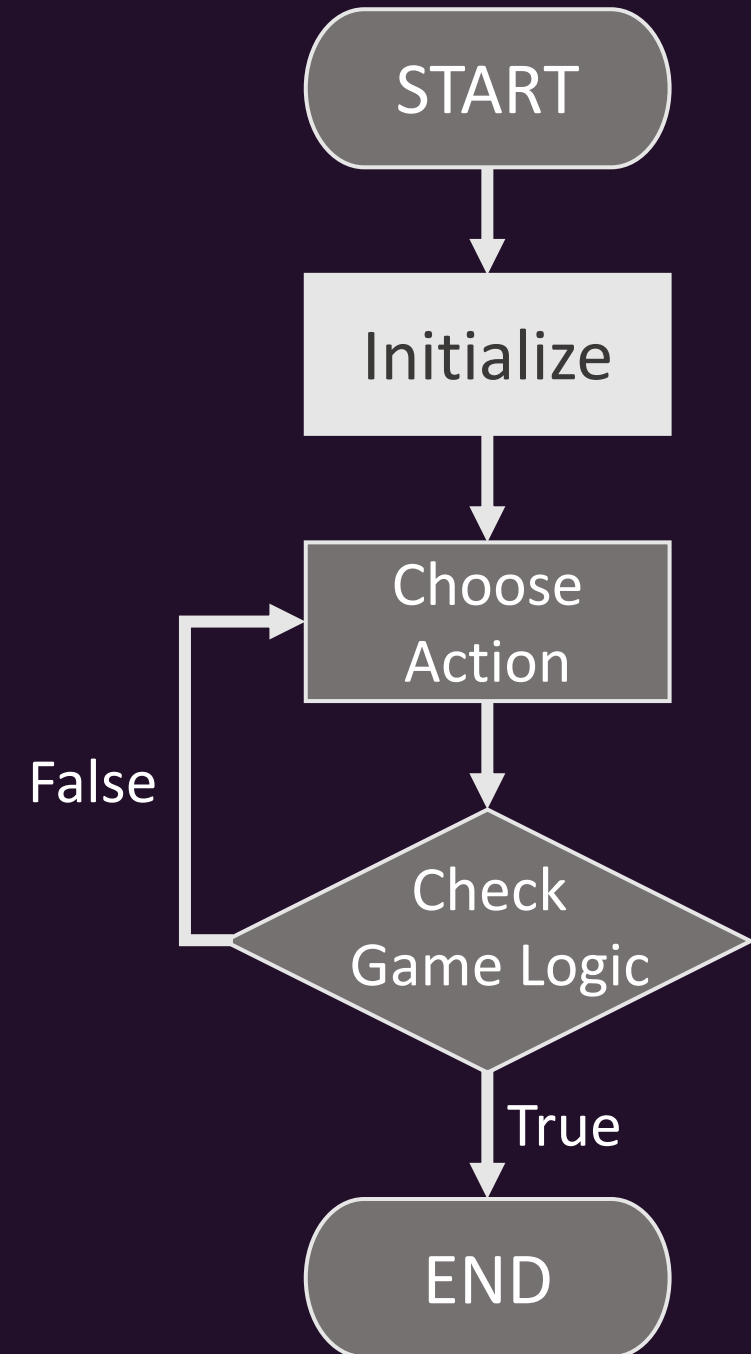
Choose Action

- **List all actions player can do**
  - Move to another room
  - Check player's status
  - Pick up item / Open chest
  - Fight with monster
  - Talk to npc
- **Get input from player and call corresponding function**

```
START
  ↓
Initialize
  ↓
Choose Action  ←── False
  ↓
Check Game Logic
  ↓ True
END
```
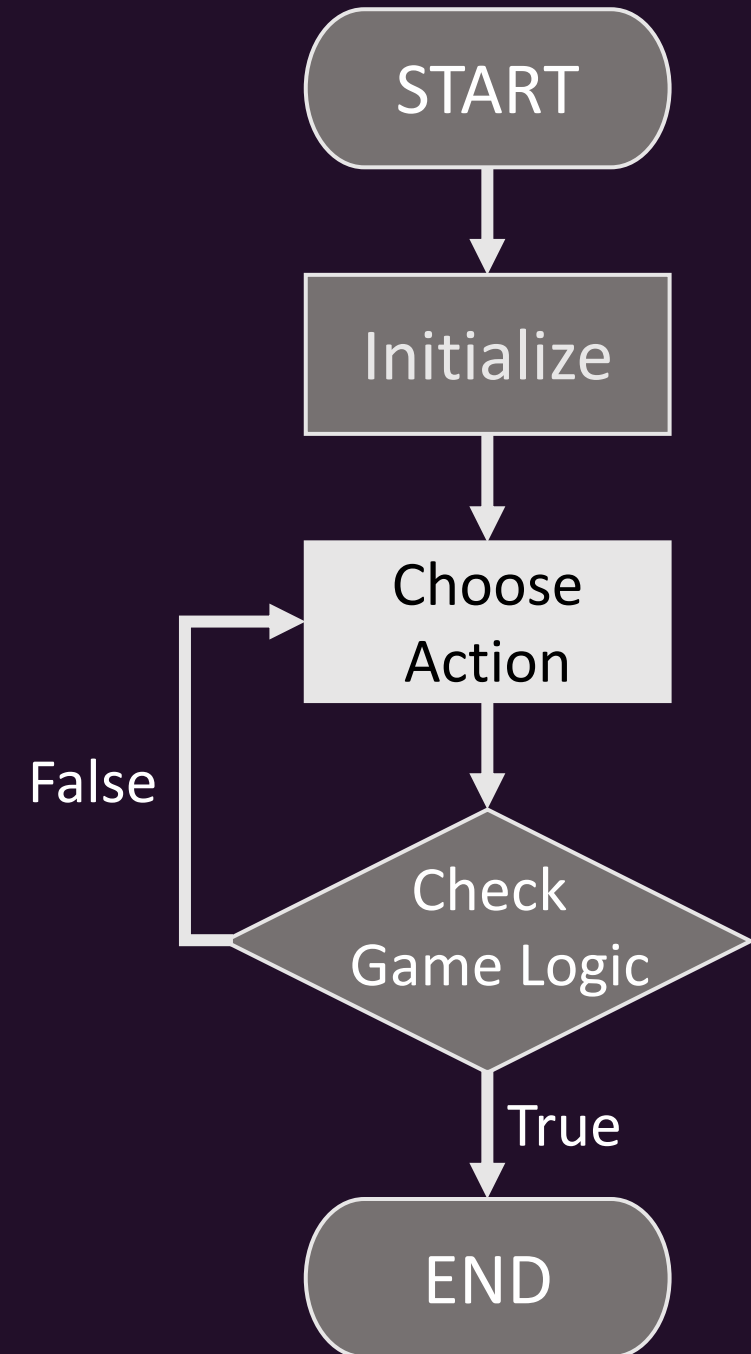
Choose Action

- **Move to another room** (handleMovement)
  - Show directions player can move
  - Move player to the chosen room
  - If there is any monster in the room, player can only back to previous room
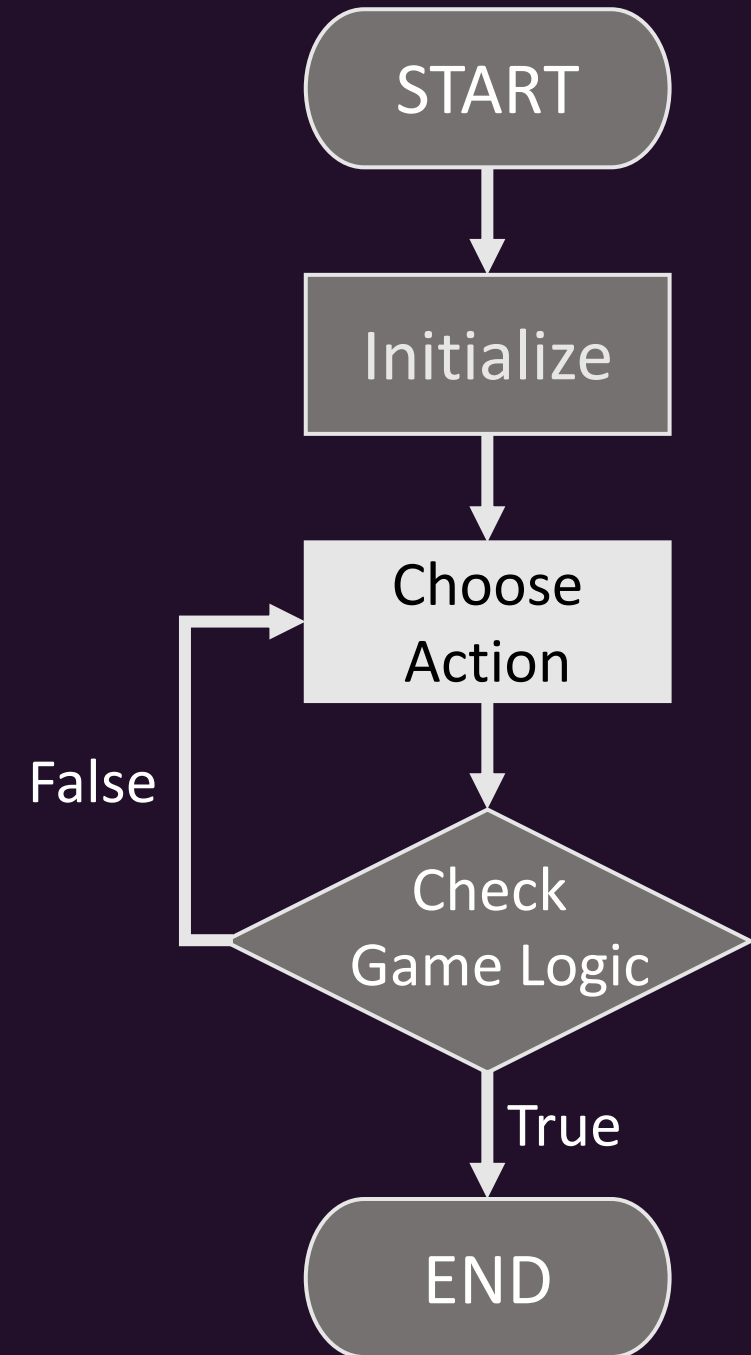
START

Initialize

Choose Action

False

Check Game Logic

True

END
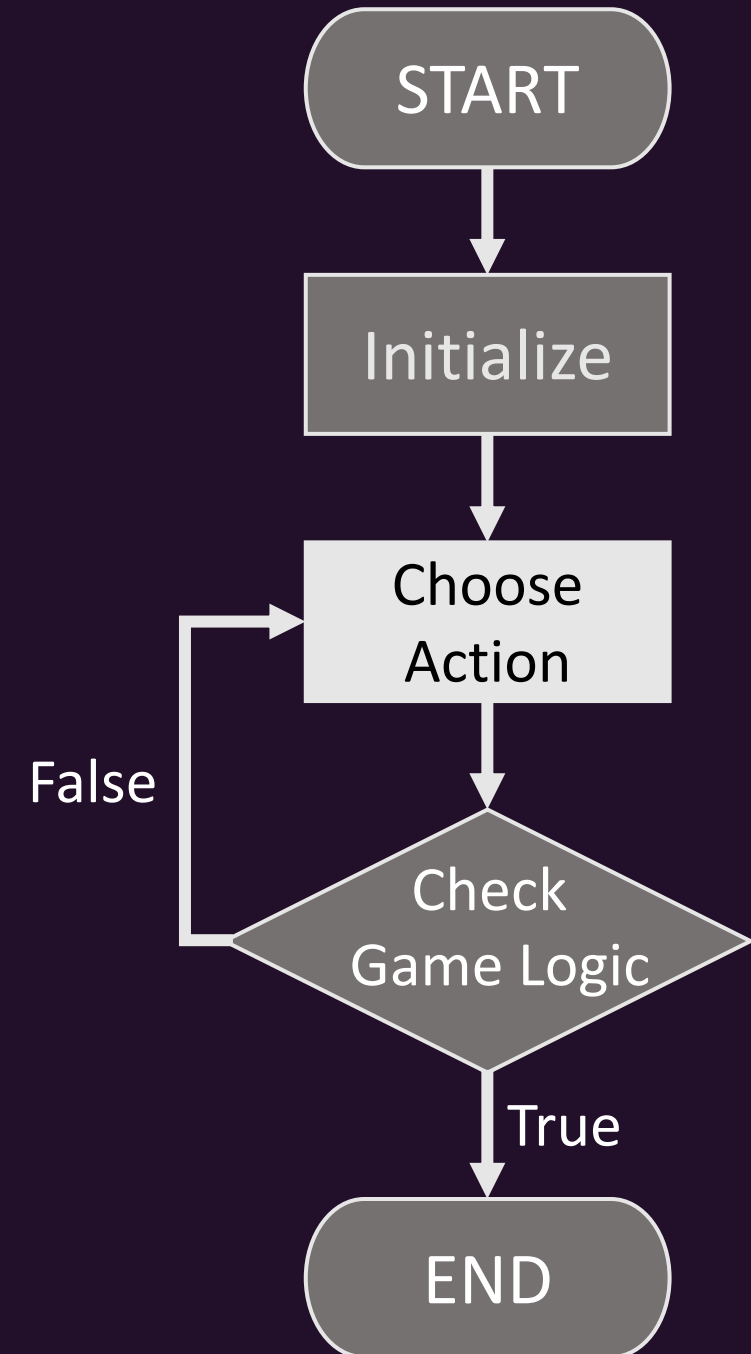
Choose Action

- **Check player's status** **(triggerEvent – Player)**
 - Show Name, Health, Attack, Defense...
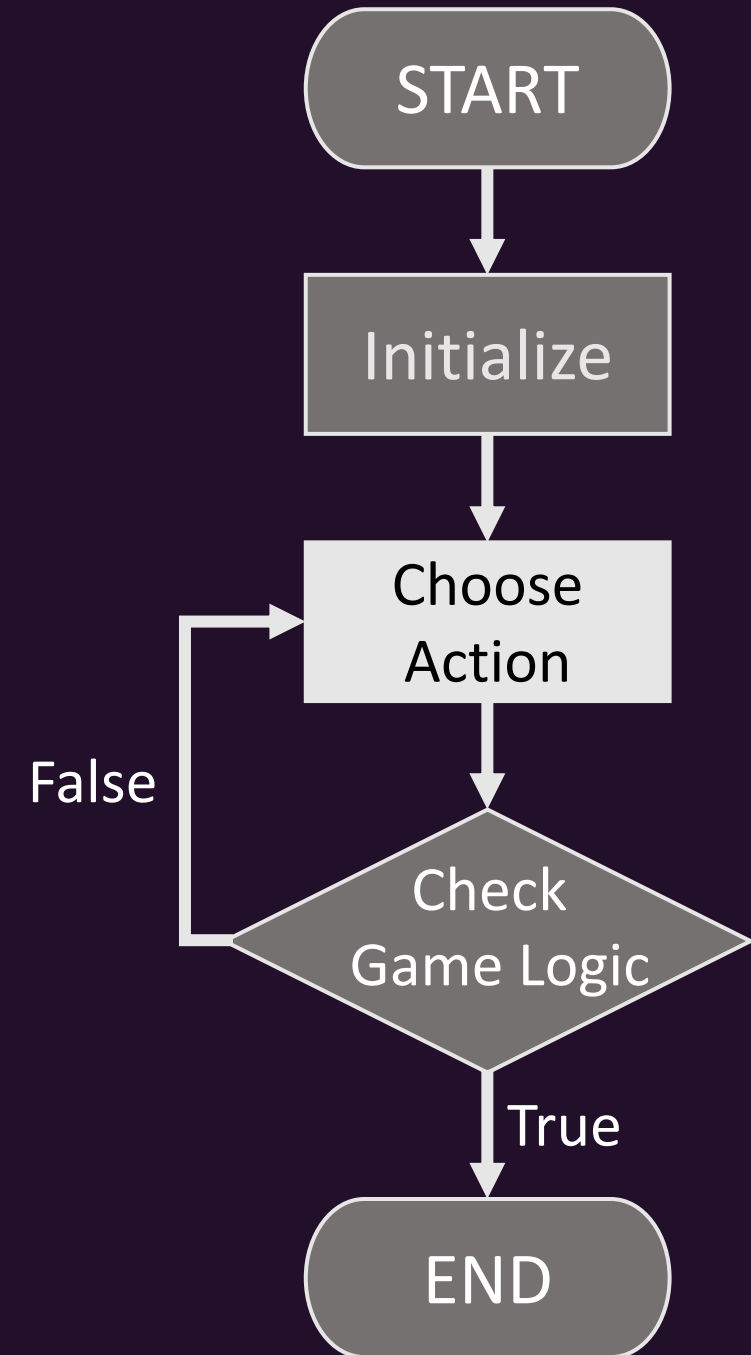
Choose Action

- **Pick up item / Open chest** (triggerEvent – Item)
 - Add item to player's inventory
  - Update player's status

```
START
  ↓
Initialize
  ↓
Choose
Action  ←── False
  ↓         ↑
Check       │
Game Logic ─┘
  ↓ True
END
```

Choose Action

- **Fight with monster** **(triggerEvent – Monster)**
  - **Retreat**
    - Back to previous room
  - **Attack**
    - Player first hit the monster
    - Check if monster is dead or not, if true break
    - Monster then hit player
    - Check if player is dead or not, if true break
    - Redo until one of them is dead
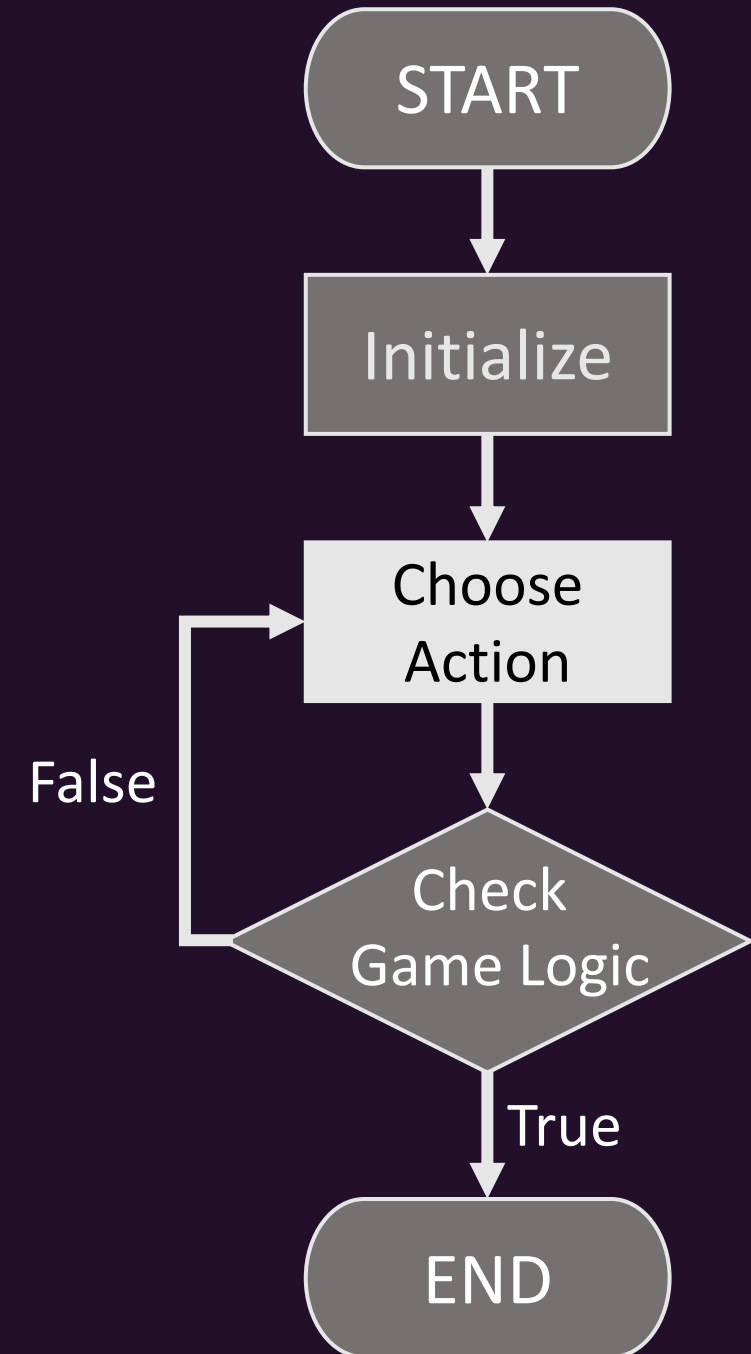
Choose Action

- **Talk to NPC** **(triggerEvent – NPC)**
  - Show the script of NPC
  - Show the item of NPC
  - Player select and pick up the item

START

Initialize

Choose
Action

False

Check
Game Logic

True

END
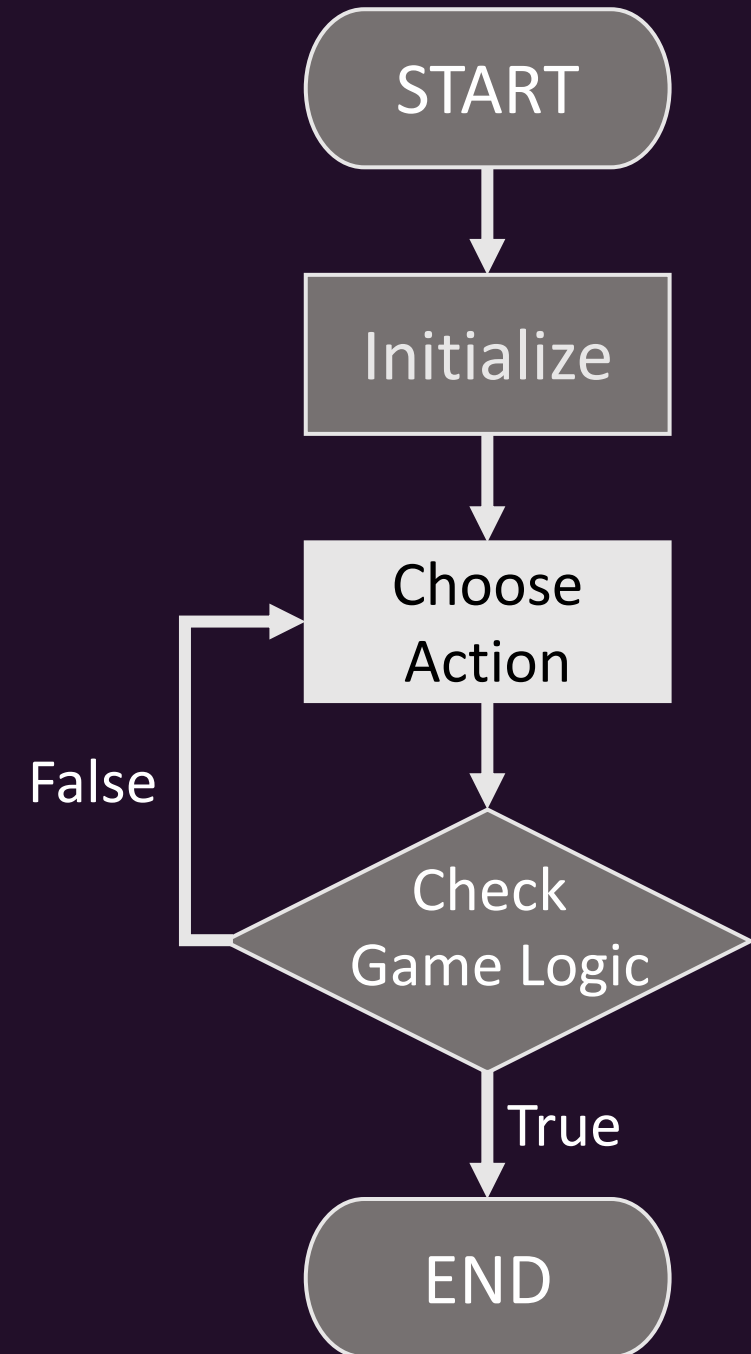
Check Game Logic

- **Check player is dead or not**
  - If true, show Game Over and break the loop
  - If false, check another condition

- **Check if player beat the Boss or not**
  - If true, show Victory and break the loop
  - If false, back to "Choose Action"

START

Initialize

Choose
Action

False

Check
Game Logic

True

END