Node

- id : int
- links : List<Node>

+ Node (id: int)
+ addLink(nodeToLink : Node): void
+ addLinks(nodesToLink: List<Node>): void
+ getId(): int
+ getLinks() : List<Node>

<Abstract> Entity

- health : int
- attack : int

+ Entity (health: int, attack: int)
+ getHealth(): int
+ getAttack(): int
+ combat(first: Entity, second: Entity): void
- takeDamage(damage: int): void
+ isAlive() : boolean
+ modifyHealth(range: int): void
+ modifyAttack(range: int): void

parent

child

Player

- currentNode : Node

+ Player()

child

Monster

+ Monster()

child

Cammo

+ Cammo()

PathFinder

- start : Node
- end : Node
- nodesToIgnore: List<Node>
- previousCities: Map<Node, Node>

+ PathFinder(start: Node, end: Node, nodesToIgnore: List<Node>)
- calculateOptimalRoute() : void
+ isRoutePossible(): boolean
+ getOptimalRoute() : List<Node>
+ getPathLength(): int

WorldSelectionMenu

- numberOfTriesLeft: int
- worlds : Map<int , List<Node>>
- beatenWorlds : Map<int, boolean>

+ WorldSelectionMenu(numberOfTries: int, worlds: Map<int, List<Node>>
+ openMenu(): void

XMLHandler

- xmlif: XMLInputFactory

+ getWorlds(filepath: string): List<List<Node>>
- getLinksBetweenNodes(filepath: string): Map<Node, List<Node>>
- readFromFile(filepath: string, tagNames: List<String>):
Map<String,List<String>>