state

Direction
<<enumeration>>
+NONE = 0
+EAST = 1
+NORTH = 2
+WEST = 3
+SOUTH = 4

TypeId
<<enumeration>>
+WALL = 1
+SPACE = 2
+MC = 3
+MONSTER = 4
+DOOR = 5
+CHEST = 6

Element
+Element()
+~Element()
+getTypeId(): TypeId const
+isStatic(): bool const
+clone(): Element*

ElementTab
-width: size_t
-height: size_t
-element_list: std::vector<std::vector<std::unique_ptr<Element>>>
+ElementTab(width:size_t=0,height:size_t=1)
+getWidth(): size_t const
+getHeight(): size_t const
+resize(width:size_t,height:size_t): void
+get(i:int,j:int=0): Element* const const
+set(i:int,j:int,e:Element*): void
+operator()(i:int,j:int=0): const Element&

MobileElement
#speed: int
#position: Coords
#isFighting: bool = false
+MobileElement()
+isStatic(): bool const
+clone(): Element*
+attack(target:state::MobileElement&): void
+attacked(attacker:state::MobileElement&): void

StaticElement
+isStatic(): bool const
+clone(): Element*

State
-floor: ElementTab
-wall: ElementTab
-chars: std::vector<std::shared_ptr<state::MobileElement>>
-monsters: std::vector<std::shared_ptr<state::MobileElement>>
#epoch: int = 0
#epochRate: int = 12
+initFloor(map:std::string,width:int,height:int): void
+initWall(map:std::string,width:int,height:int): void
+initChars(map:std::string,width:int,height:int): void
+getFloor(): const ElementTab& const
+getWall(): const ElementTab& const
+getChars(): const std::vector<std::shared_ptr<state::MobileElement>>& const
+getMonsters(): const std::vector<std::shared_ptr<state::MobileElement>>& const

Coords
#x: int
#y: int
+Coords()
+Coords(x:int,y:int)
+operator==(a:const Coords&): bool

Monster
#HealthPoints: int
#Defense: int
#Attack: int
+Monster(id:MonsterTypeId)
+getTypeId(): TypeId const
+clone(): Element*
+attack(target:state::MainCharacter&): void
+attacked(attacker:state::MainCharacter&): void

MainCharacter
#HealthPoints: int = 6
#Defense: int = 0
#Attack: int = 1
+MainCharacter()
+getTypeId(): TypeId const
+clone(): Element*
+attack(target:state::Monster&): void
+attacked(attacker:state::Monster&): void

Wall
+Wall(id:WallTypeId)
+getTypeId(): TypeId const
+clone(): Element*

Space
#colored: bool = false
+Space(id:SpaceTypeId)
+getTypeId(): TypeId const
+getSpaceTypeId(): SpaceTypeId const
+setSpaceTypeId(id:SpaceTypeId): void
+clone(): Element*

Door
#statusDoor: bool = false
+Door()
+getTypeId(): TypeId const
+isOpened(): bool
+clone(): Element*

Chest
#statusChest: bool = false
+Chest(id:ChestContentId)
+getTypeId(): TypeId const
+isOpened(): bool
+clone(): Element*

MonsterTypeId
<<enumeration>>
+GHOST = 1
+BOSS = 2

SpaceTypeId
<<enumeration>>
+EMPTY = 1
+START = 2
+END = 3
+SPAWN = 4

Observable
-observer: mutable std::vector<Observer*>
+~Observable()
+registerObserver(o:Observer*): void const
+unregisterObserver(o:Observer*): void const
+notifyObserver(e:const Event&): void const

Observer
<<interface>>
+stateChanged(e:const Event&): void

Event
+~Event()
+getEventType(): EventTypeId const

EventTypeId
<<enumeration>>
+EVENT_STATE = 1
+EVENT_TAB = 2

ChestContentId
<<enumeration>>
+HealthPoints = 1
+PowerBoost = 2
+DefenseBoost = 3

WallTypeId
<<enumeration>>
+TOPLEFT = 1
+TOPRIGHT = 2
+BOTTOMRIGHT = 3
+BOTTOMLEFT = 4
+HORI = 5
+VERT_LEFT = 6
+VERT_RIGHT = 7
+TOP = 8
+CORNERLEFT = 9
+CORNERRIGHT = 10
+CORNERFRONTLEFT = 11
+CORNERFRONTRIGHT = 12
+INNERCORNER_L_TL = 13
+INNERCORNER_L_TR = 14
+INNERCORNER_ML = 15
+INNERCORNER_MR = 16
+INNERCORNER_T_TL = 17
+INNERCORNER_T_TR = 18

TabEvent
+x: int
+y: int
+TabEvent(x:int,y:int)
+getEventType(): EventTypeId const

StateEvent
+stateEvent: StateEventId
+StateEvent(se:StateEventId)
+getEventType(): EventTypeId const

StateEventId
<<enumeration>>
+ALL_CHANGED = 0
+EPOCH_CHANGED = 1
+GRID_CHANGED = 2
+MC_CHANGED = 3
+MONSTER_CHANGED = 4
+DOOR_CHANGED = 5
+CHEST_CHANGED = 6

Association labels: typeId, direction, monsterTypeId, spaceTypeId, wallTypeId, chestContentId