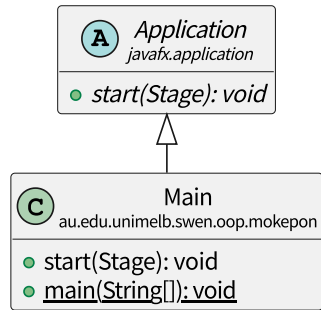
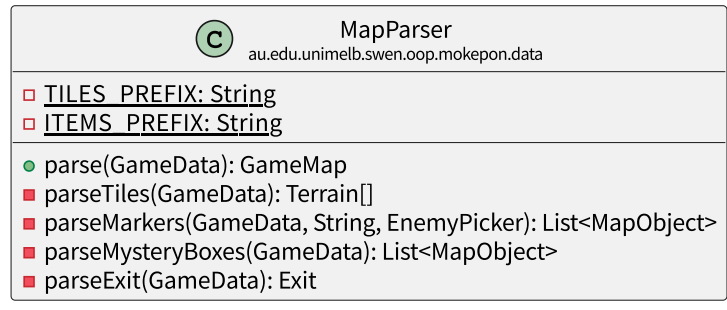


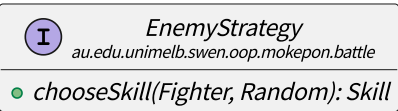
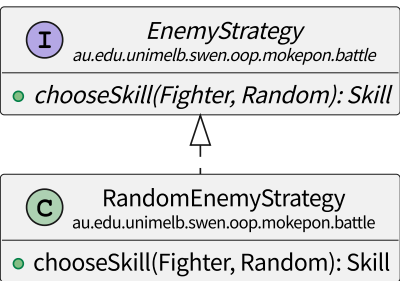
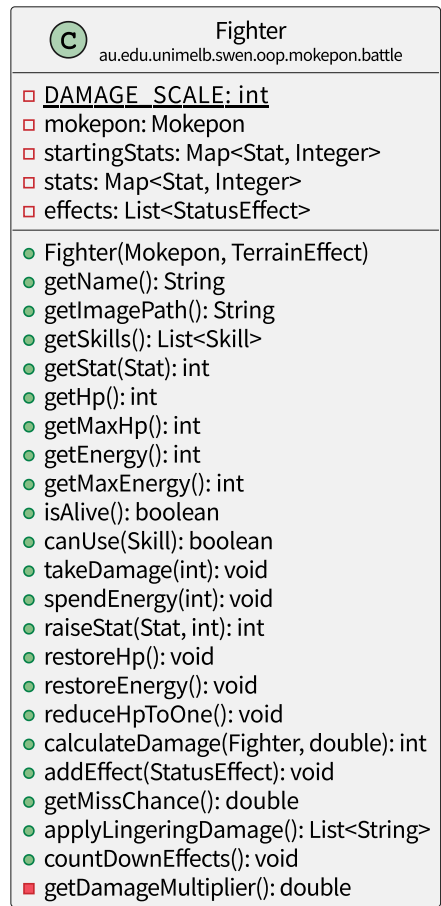
au.edu.unimelb.swen.oop.mokepon



au.edu.unimelb.swen.oop.mokepon.data



au.edu.unimelb.swen.oop.mokepon.battle



au.edu.unimelb.swen.oop.mokepon.data

C

MokeponParser
au.edu.unimelb.swen.oop.mokepon.data

PREFIX: String

● parse(GameData): List<Mokepon>

■ parseMokepon(GameData, String): Mokepon

■ parseSkills(String): List<Skill>

UMLDoclet 2.3.2

C

GameData
au.edu.unimelb.swen.oop.mokepon.data

properties: Properties

● GameData(Properties)

● has(String): boolean

● getText(String): String

● getNumber(String): int

● getPosition(String): Position

● countEntries(String, String): int

UMLDoclet 2.3.2

C

GamelImages
au.edu.unimelb.swen.oop.mokepon.data

typeImages: Map<MokeponType, String>

■ tileImages: Map<Terrain, String>

■ buttonImages: Map<Direction, String>

■ itemImages: Map<Item, String>

■ playerImage: String

● GamelImages(Map<MokeponType, String>, Map<Terrain, String>, Map<Direction, String>, Map<Item, String>, String)

● hasTypeImage(MokeponType): boolean

● getTypeImage(MokeponType): String

● getTileImage(Terrain): String

● getButtonImage(Direction): String

● getItemImage(Item): String

● getPlayerImage(): String

UMLDoclet 2.3.2

C

GameDataReader
au.edu.unimelb.swen.oop.mokepon.data

ARGUMENT_PREFIX: String

● read(List<String>): GameData

UMLDoclet 2.3.2

C

ImageParser
au.edu.unimelb.swen.oop.mokepon.data

● parse(GameData): GamelImages

■ readImages(GameData, String, Enum<T>[]): Map<T extends Enum<T>, String>

■ readItemImages(GameData): Map<Item, String>

UMLDoclet 2.3.2

au.edu.unimelb.swen.oop.mokepon.effect

A

StatusEffect
au.edu.unimelb.swen.oop.mokepon.effect

C

Bleeding
au.edu.unimelb.swen.oop.mokepon.effect

TURN: int

■ HALF_STRENGTH: double

■ attacker: Fighter

● Bleeding(Fighter)

● applyLingeringDamage(Fighter): List<String>

UMLDoclet 2.3.2

A

StatusEffect
au.edu.unimelb.swen.oop.mokepon.effect

turnsLeft: int

◆ StatusEffect(int)

● getDamageMultiplier(): double

● getMissChance(): double

● applyLingeringDamage(Fighter): List<String>

● countDown(): void

● isOver(): boolean

◆ isLastTurn(): boolean

UMLDoclet 2.3.2

A

StatusEffect
au.edu.unimelb.swen.oop.mokepon.effect

C

Blinded
au.edu.unimelb.swen.oop.mokepon.effect

TURN: int

■ MISS_CHANCE: double

● getMissChance(): double

UMLDoclet 2.3.2

A

StatusEffect
au.edu.unimelb.swen.oop.mokepon.effect

C

Stunned
au.edu.unimelb.swen.oop.mokepon.effect

TURN: int

■ DAMAGE_MULTIPLIER: double

● getDamageMultiplier(): double

UMLDoclet 2.3.2

au.edu.unimelb.swen.oop.mokepon.item

C

Inventory
au.edu.unimelb.swen.oop.mokepon.item

□ counts: Map<Item, Integer>

● Inventory(List<Item>)

● getItems(): List<Item>

● getCount(Item): int

● has(Item): boolean

● add(Item): void

● remove(Item): void

UMLDoclet 2.3.2

A

Item
au.edu.unimelb.swen.oop.mokepon.item

□ typeName: String

□ name: String

□ imageName: String

◆ Item(String, String, String)

● getTypeName(): String

● getName(): String

● getImageName(): String

● use(Fighter, Fighter): String

◆ describeUse(Fighter, String): String

UMLDoclet 2.3.2

C

ItemCatalogue
au.edu.unimelb.swen.oop.mokepon.item

□ HEALTH TONIC: Item

□ COFFEE: Item

□ POISON: Item

□ ITEMS: List<Item>

□ BATTLE ORDER: List<Item>

● getAll(): List<Item>

● getBattleOrder(): List<Item>

● find(String): Item

UMLDoclet 2.3.2

A

Item
au.edu.unimelb.swen.oop.mokepon.item

● use(Fighter, Fighter): String

C

Poison
au.edu.unimelb.swen.oop.mokepon.item

● use(Fighter, Fighter): String

UMLDoclet 2.3.2

A

Item
au.edu.unimelb.swen.oop.mokepon.item

● use(Fighter, Fighter): String

C

Coffee
au.edu.unimelb.swen.oop.mokepon.item

● use(Fighter, Fighter): String

UMLDoclet 2.3.2

A

Item
au.edu.unimelb.swen.oop.mokepon.item

● use(Fighter, Fighter): String

C

HealthTonic
au.edu.unimelb.swen.oop.mokepon.item

● use(Fighter, Fighter): String

UMLDoclet 2.3.2

au.edu.unimelb.swen.oop.mokepon.model

C

Mokepon
au.edu.unimelb.swen.oop.mokepon.model

□ name: String

□ imagePath: String

□ type: MokeponType

□ stats: Map<Stat, Integer>

□ skills: List<Skill>

● Mokepon(String, String, MokeponType, Map<Stat, Integer>, List<Skill>)

● getName(): String

● getImagePath(): String

● getType(): MokeponType

● getStat(Stat): int

● getSkills(): List<Skill>

UMLDoclet 2.3.2

E

MokeponType
au.edu.unimelb.swen.oop.mokepon.model

○ NORMAL

○ FIRE

○ ICE

○ WATER

○ LIGHTNING

□ strongTerrain: Terrain

□ weakTerrain: Terrain

■ MokeponType(Terrain, Terrain)

● getEffectOn(Terrain): TerrainEffect

UMLDoclet 2.3.2

E

TerrainEffect
au.edu.unimelb.swen.oop.mokepon.model

○ ADVANTAGE

○ NEUTRAL

○ DISADVANTAGE

□ multiplier: double

□ description: String

■ TerrainEffect(double, String)

● getDescription(): String

● adjust(int): int

UMLDoclet 2.3.2

E

Stat
au.edu.unimelb.swen.oop.mokepon.model

○ HP

○ ENERGY

○ STRENGTH

○ ACCURACY

○ DEFENCE

○ DODGE

□ label: String

□ minimum: int

□ maximum: int

■ Stat(String, int, int)

● getLabel(): String

● limit(int): int

UMLDoclet 2.3.2

E

Terrain
au.edu.unimelb.swen.oop.mokepon.model

○ BARRIER

○ FOREST

○ GRASS

○ OCEAN

○ PATH

○ ROCKS

○ WATER

□ symbol: char

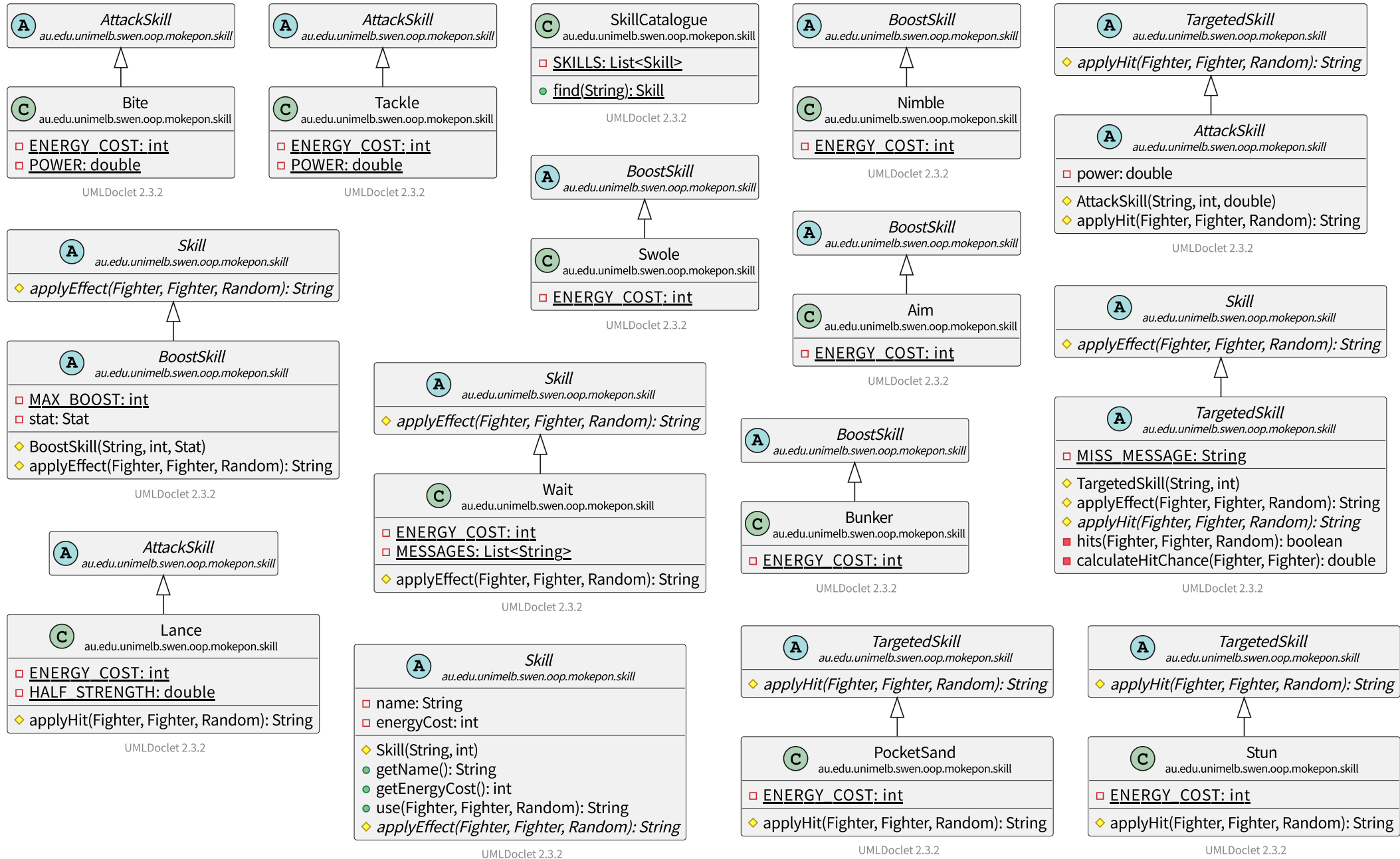
□ walkable: boolean

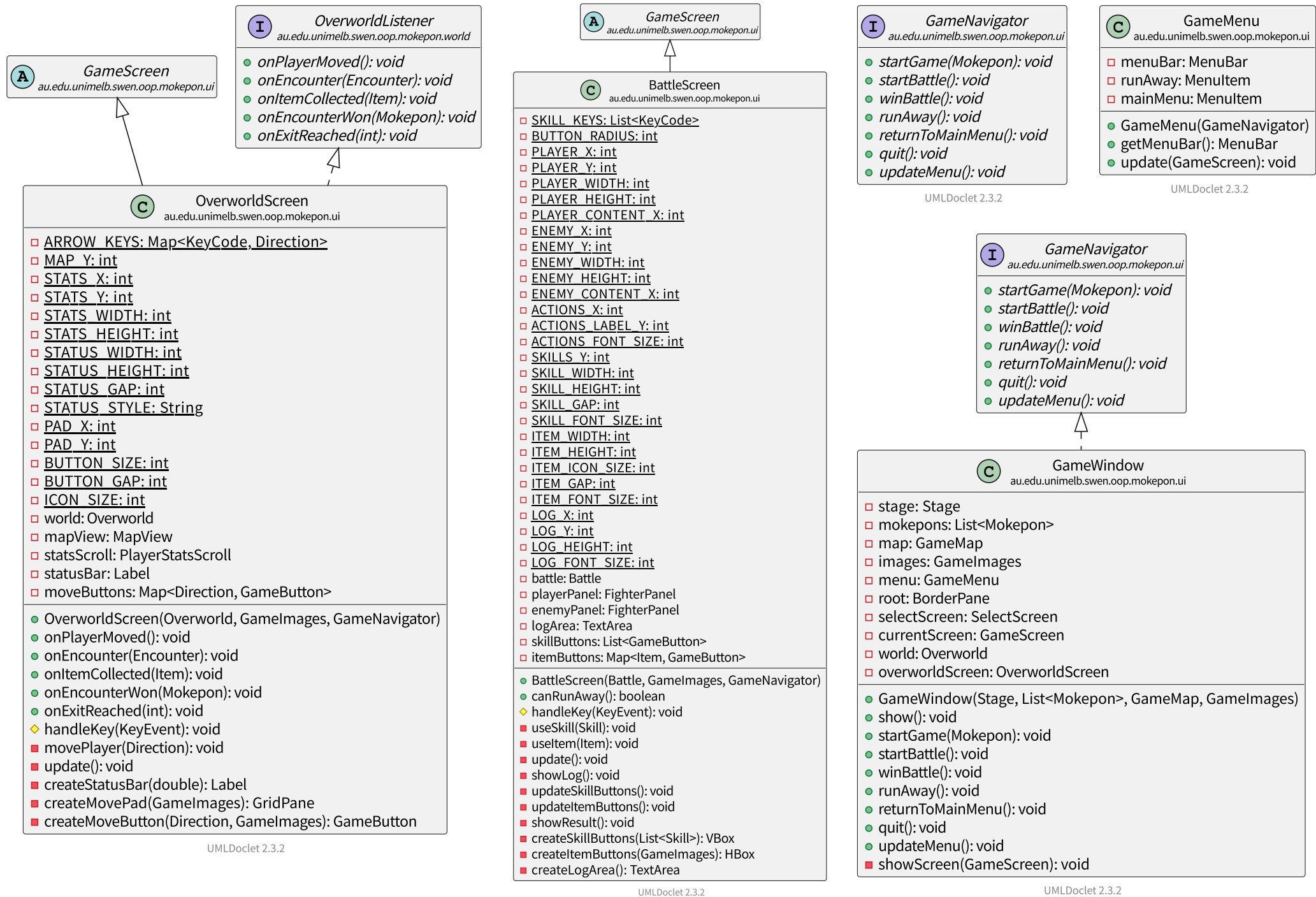
■ Terrain(char, boolean)

● isWalkable(): boolean

● fromSymbol(char): Terrain

UMLDoclet 2.3.2





A

GameScreen

au.edu.unimelb.swen.oop.mokepon.ui

□ navigator: GameNavigator

□ view: Pane

□ dialog: ModalDialog

◆ GameScreen(GameNavigator)

● getView(): Pane

● handleKeyPress(KeyEvent): void

● canRunAway(): boolean

● canReturnToMainMenu(): boolean

◆ handleKey(KeyEvent): void

◆ getNavigator(): GameNavigator

◆ addToView(Node...): void

◆ addToView(List<Node>): void

◆ showDialog(ModalDialog): void

◆ closeDialog(): void

◆ isDialogOpen(): boolean

◆ createGameEndDialog(String): ModalDialog

UMLDoclet 2.3.2

C

Button

javaafx.scene.control

□ NORMAL: double

□ HIGHLIGHTED: double

□ DISABLED_OPACITY: double

□ cornerRadius: int

● GameButton(String, Font, int)

● GameButton(ImageView, int)

● addPressEffect(): void

● setHighlighted(boolean): void

● setUsable(boolean): void

■ setBackgroundOpacity(double): void

UMLDoclet 2.3.2

C

StatBar

au.edu.unimelb.swen.oop.mokepon.ui

□ SEPARATOR: String

□ FONT_SIZE: int

□ numbers: Label

● LabelledStatBar(double, double, String, int, int)

● update(int, int): void

● getNodes(): List<Node>

UMLDoclet 2.3.2

A

GameScreen

au.edu.unimelb.swen.oop.mokepon.ui

C

SelectScreen

au.edu.unimelb.swen.oop.mokepon.ui

□ PROMPT: String

□ TITLE_X: int

□ TITLE_Y: int

□ TITLE_FONT_SIZE: int

□ LIST_X: int

□ LIST_Y: int

□ LIST_WIDTH: int

□ LIST_HEIGHT: int

□ LIST_BUTTON_WIDTH: int

□ LIST_FONT_SIZE: int

□ LIST_GAP: int

□ BUTTON_RADIUS: int

□ CIRCLE_CENTRE_X: int

□ CIRCLE_CENTRE_Y: int

□ CIRCLE_RADIUS: int

□ PORTRAIT_SIZE: int

□ CIRCLE_COLOUR: Color

□ NAME_Y: int

□ NAME_WIDTH: int

□ NAME_FONT_SIZE: int

□ TYPE_ICON_SIZE: int

□ TYPE_ICON_GAP: int

□ SCROLL_X: int

□ SCROLL_Y: int

□ SCROLL_WIDTH: int

□ SCROLL_HEIGHT: int

□ STATS_X: int

□ STATS_Y: int

□ LABEL_WIDTH: int

□ VALUE_WIDTH: int

□ ROW_HEIGHT: int

□ STAT_FONT_SIZE: int

□ CHOOSE_X: int

□ CHOOSE_Y: int

□ CHOOSE_SIZE: int

□ CHOOSE_FONT_SIZE: int

□ CHOOSE_RADIUS: int

□ images: GamelImages

□ portraitView: ImageView

□ nameLabel: Label

□ typelcon: ImageView

□ statValues: Map<Stat, Label>

□ skillsValue: Label

□ typeValue: Label

□ chooseButton: GameButton

□ selectedButton: GameButton

□ selectedMokepon: Mokepon

● SelectScreen(List<Mokepon>, GamelImages, GameNavigator)

● canReturnToMainMenu(): boolean

● clearSelection(): void

■ select(Mokepon, GameButton): void

■ showTypelcon(boolean): void

■ formatSkills(Mokepon): String

■ createTitle(DropShadow): Label

■ createPortraitCircle(): Circle

■ createNameRow(): HBox

■ createMokeponList(List<Mokepon>): ScrollPane

■ createChooseButton(): GameButton

■ createStatsGrid(): GridPane

■ addStatRow(GridPane, int, String, Label, int): void

■ createValueLabel(): Label

■ createRowLabel(String, Font): Label

UMLDoclet 2.3.2

C

FighterScroll

au.edu.unimelb.swen.oop.mokepon.ui

□ FAINT_TINT: Color

□ scroll: ImageView

□ faintMask: Rectangle

● FighterScroll(double, double, double, double)

● getScroll(): ImageView

● getFaintMask(): Rectangle

● setFainted(boolean): void

■ createFaintMask(Image, double, double, double, double): Rectangle

UMLDoclet 2.3.2

C

UiStyle

au.edu.unimelb.swen.oop.mokepon.ui

○ GAME_WIDTH: int

○ GAME_HEIGHT: int

○ SCROLL_FILE: String

□ BACKGROUND_FILE: String

□ FONT_FILE: String

□ SANS_SERIF: String

□ MONOSPACE: String

□ STRAIGHT_DOWN: double

● loadGameFont(int): Font

● createSansSerifFont(FontWeight, int): Font

● createMonospaceFont(FontWeight, int): Font

● loadImage(String): Image

● createImageView(Image, double, double, double, double): ImageView

● createIcon(Image, double): ImageView

● createBackground(): ImageView

● tint(ImageView, String): void

● createWhiteOutline(): DropShadow

● createCenteredLabel(String, Font, double, double, double): Label

● createLabel(String, Font): Label

UMLDoclet 2.3.2

C

StatBar

au.edu.unimelb.swen.oop.mokepon.ui

◆ WIDTH: int

◆ HEIGHT: int

□ MIN_VISIBLE_RATIO: double

□ bar: ProgressBar

● StatBar(double, double, String, int, int)

● update(int, int): void

● getNodes(): List<Node>

■ calculateRatio(int, int): double

UMLDoclet 2.3.2

C

ModalDialog

au.edu.unimelb.swen.oop.mokepon.ui

SHADE_STYLE: String

BOX_STYLE: String

BUTTON_STYLE: String

HEADING_SIZE: int

SPACING: int

BUTTON_GAP: int

PADDING: Insets

layer: StackPane

messages: VBox

buttons: HBox

ModalDialog(String)

getLayer(): StackPane

addMessage(String): void

addButton(String, EventHandler<ActionEvent>): void

focusFirstButton(): void

UMLDoclet 2.3.2

C

PlayerStatsScroll

au.edu.unimelb.swen.oop.mokepon.ui

FONT_SIZE: int

ROW_HEIGHT: int

LABEL_WIDTH: int

SEPARATOR_WIDTH: int

TEXT_X: int

TEXT_Y: int

player: Player

nameFont: Font

numberFont: Font

pane: Pane

fightValue: Label

itemValues: Map<Item, Label>

PlayerStatsScroll(Player, double, double, double, double)

getNode(): Pane

update(): void

addRow(GridPane, int, String, Label): void

createLabel(String, Font): Label

UMLDoclet 2.3.2

C

FighterPanel

au.edu.unimelb.swen.oop.mokepon.ui

HP_COLOUR: String

ENERGY_COLOUR: String

PORTRAIT_SIZE: int

NAME_FONT_SIZE: int

NAME_OFFSET: int

PORTRAIT_OFFSET: int

HP_BAR_OFFSET: int

ENERGY_BAR_OFFSET: int

fighter: Fighter

scroll: FighterScroll

portrait: ImageView

name: Label

hpBar: StatBar

energyBar: StatBar

FighterPanel(Fighter, Rectangle2D, double, boolean)

getNodes(): List<Node>

getFaintMask(): Rectangle

update(): void

UMLDoclet 2.3.2

C

MapView

au.edu.unimelb.swen.oop.mokepon.ui

TILE_SIZE: int

world: Overworld

pane: Pane

objectLayer: Pane

playerView: ImageView

loadedImages: Map<String, Image>

objectViews: Map<MapObject, ImageView>

MapView(Overworld, GamelImages, double, double)

getNode(): Pane

update(): void

createTileLayer(GameMap, GamelImages): Pane

createObjectView(MapObject): ImageView

createTileView(Image, Position): ImageView

placeOnTile(ImageView, Position): void

getImage(String): Image

UMLDoclet 2.3.2

C

Overworld

au.edu.unimelb.swen.oop.mokepon.world

map: GameMap

mokepons: List<Mokepon>

player: Player

objects: List<MapObject>

listeners: List<OverworldListener>

random: Random

encounter: Encounter

Overworld(GameMap, List<Mokepon>, Mokepon)

Overworld(GameMap, List<Mokepon>, Mokepon, Random)

addListener(OverworldListener): void

getMap(): GameMap

getPlayer(): Player

getObjects(): List<MapObject>

canMove(Direction): boolean

move(Direction): void

createBattle(): Battle

runAway(): void

winEncounter(): void

startEncounter(Marker): void

collect(MysteryBox): void

reachExit(): void

notifyListeners(Consumer<OverworldListener>): void

UMLDoclet 2.3.2

C

Encounter

au.edu.unimelb.swen.oop.mokepon.world

marker: Marker

playerMokepon: Mokepon

enemy: Mokepon

terrain: Terrain

Encounter(Marker, Mokepon, Mokepon, Terrain)

getMarker(): Marker

getEnemy(): Mokepon

getPlayerEffect(): TerrainEffect

getEnemyEffect(): TerrainEffect

getTerrainMessages(): List<String>

createBattle(Inventory): Battle

addTerrainMessage(List<String>, String, TerrainEffect): void

UMLDoclet 2.3.2

au.edu.unimelb.swen.oop.mokepon.world

