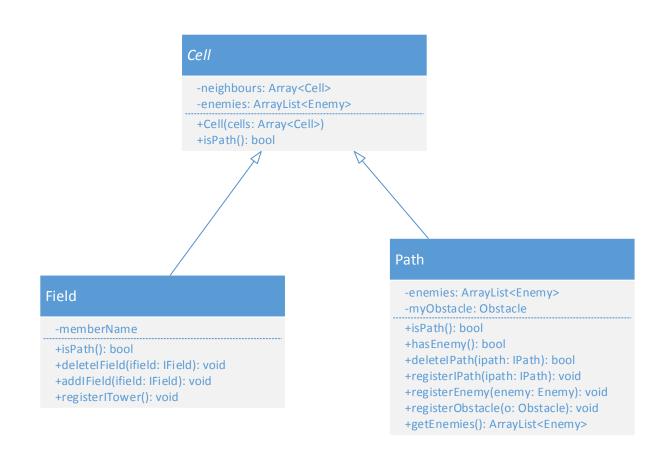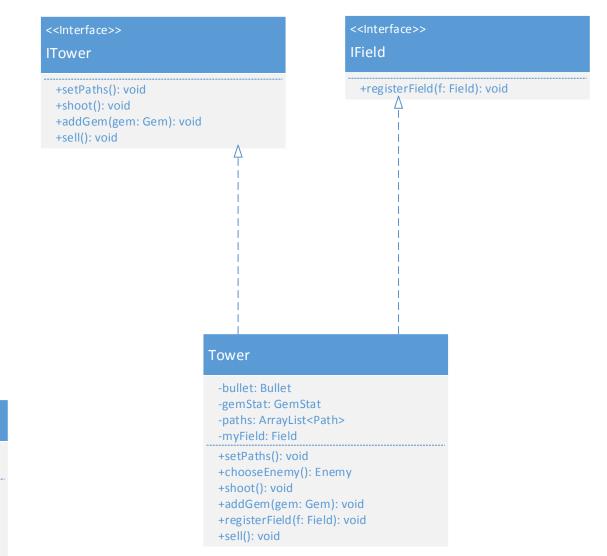## Game

-map: Map
-enemies: ArrayList<Enemy>
-towers: ArrayList<Tower>
-mana: int

+update(): void
+initialize(): void
+tick(): void

Map

-name: String
-level: int
-grid: Array<Array<Cell>>

+Map(name: String)
+load(name: String)

Cell

-neighbours: Array<Cell>
-enemies: ArrayList<Enemy>

+Cell(cells: Array<Cell>)
+isPath(): bool

Field

-memberName

+isPath(): bool
+deleteIField(ifield: IField): void
+addIField(ifield: IField): void
+registerITower(): void

Path

-enemies: ArrayList<Enemy>
-myObstacle: Obstacle

+isPath(): bool
+hasEnemy(): bool
+deleteIPath(ipath: IPath): bool
+registerIPath(ipath: IPath): void
+registerEnemy(enemy: Enemy): void
+registerObstacle(o: Obstacle): void
+getEnemies(): ArrayList<Enemy>

<<Interface>>
ITower

+setPaths(): void
+shoot(): void
+addGem(gem: Gem): void
+sell(): void

<<Interface>>
IField

+registerField(f: Field): void

<<Interface>>
IObstacle

+slow(intensity: int, p: Path): void
+amortization(): void

<<Interface>>
IPath

+eliminate(p: Path): void
+registerPath(p: Path): void

Bullet

-damage: int
-enemyType: Enemy

+refresh(gem: Gem): void
+getDamageHobbit: int
+getDamageElf(): int
+getDamageHuman(): int
+getDamageDwarf(): int

Tower

-bullet: Bullet
-gemStat: GemStat
-paths: ArrayList<Path>
-myField: Field

+setPaths(): void
+chooseEnemy(): Enemy
+shoot(): void
+addGem(gem: Gem): void
+registerField(f: Field): void
+sell(): void

Obstacle

-slowIntens: int
-myPath: Path
-amort: int

+slow(intensity: int, p: Path): void
+amortization(): void
+eliminate(p: Path): void
+registerPath(p Path): void

Enemy

-speed: int
-modSpeed: int
-health: int

+registerPath(p Path): void
+eliminate(p: Path): void
+hurt(Bullet): void

Elf

-memberName

Hobbit

-memberName

+hurt(Bullet): void

Dwarf

-memberName

+hurt(Bullet): void

Human

-memberName

+hurt(Bullet): void

Gem

-damage: int
-range: int
-speed: int
-enemyType: Enemy

GemStat

-memberName

+ refresh(gem: Gem): void
+getValue(): int