Model::Main

Color

«enumeration» Code
FG_RED
FG_BLUE
FG_GREEN
...
BG_RED
BG_BLUE
BG_GREEN
...

«enumeration» StaticCode
STORM
HILL
BERSERK
TRACER
AMBUSHER
PACIFIST
DUMMY

Modifier
-code: Code

+opStream(os: ostream&, mod: const Modifier&)

Entity

«enumeration» Move
NOOP
ATTACK
UP
DOWN
LEFT
RIGHT

«enumeration» Weapon
NOWEAPON
SHOVEL
PISTOL
SEMIAUTO

«enumeration» Armor
NOARMOR
WOODEN
BRICK
METAL

«enumeration» Constants
STORM_WIDTH_COEFF
STORM_DMG_PER_LVL

Berserk
+Berserk(id: uint, x: int, y: int)
+const override getWeapon(): PISTOL
+const override getArmor(): WOODEN
+const override getPriorityList(): {ATK*UP*LEFT*DOWN*RIGHT}

Tracer
+Tracer(id: uint, x: int, y: int)
+const override getWeapon(): SHOVEL
+const override getArmor(): BRICK
+const override getPriorityList(): {UP*LEFT*DOWN*RIGHT*ATK}

Ambusher
+Ambusher(id: uint, x: int, y: int)
+const override getWeapon(): SEMIAUTO
+const override getArmor(): NOARMOR
+const override getPriorityList(): {ATK}

Pacifist
+Pacifist(id: uint, x: int, y: int)
+const override getWeapon(): NOWEAPON
+const override getArmor(): METAL
+const override getPriorityList(): {UP*LEFT*DOWN*RIGHT}

Dummy
+Dummy(id: uint, x: int, y: int)
+const override getWeapon(): NOWEAPON
+const override getArmor(): NOARMOR
+const override getPriorityList(): {NOOP}

Player
#id: const uint
#coordinate: Coordinate
#HP: int

+Player(id: uint, x: int, y: int)
+const getID(): uid
+const getCoord(): const Coordinate&
+const getHP(): int
+const getBoardID(): string
+const getArmor(): Armor
+const getWeapon(): Weapon
+const getPriorityList(): vector<Move>
+const getFullName(): string
+const isDead(): bool
+const getColorID(): Code
+executeMove(): void
+attackTo(player: Player*): bool

EntityHelper
+stormWidthForRound(currentRound: uint): int
+stormDamageForRound(currentRound: uint): int
+damageForWeapon(weapon: Weapon): int
+damageReductionForArmor(armor: Armor): int

GameParser
+parseFileWithName(filename: const string&): boardSize players

BoardPrinter
+printBoard(board: const Board&, shouldPrintColors: bool): void

GameEngine
-currentRound: uint
-board: Board

+GameEngine(boardSize: uint, vector_Player *_ *players)
+const getBoard(): const Board&
+operator[](id: uint): Player*
+const isFinished(): bool
+takeTurn(): void
+takeTurnForPlayer(playerID: uint): Move
+const decideWinnerPlayer(): Player*

Coordinate
+x: int
+y: int

+Coordinate(int x, int y)
+opGreater(rhs: const Coordinate&): bool
+opEqual(rhs: const Coordinate&): bool
+opNotEqual(rhs: const Coordinate&): bool
+opPlus(move: Move): Coordinate
+opMinus(rhs: const Coordinate&): int

Board
-boardSize: const uint
-players: vector<Player *> *

+Board(uint boardSize, vector_Player*_ *players)
+const getSize(): uint
+const isCoordInBoard(rhs: const Coordinate&): bool
+const isStormInCoord(rhs: const Coordinate&): bool
+const isCoordHill(rhs: const Coordinate&): bool
+const operator[](rhs: const Coordinate&): Player*
+const calculateCoordWithMove(move: Move, rhs: const Coordinate&): Coordinate
+const visibleCoordsFromCoord(rhs: const Coordinate&): vector_Coordinate_
+updateStorm(uint currentRound): void