Character <<abstract>>

#board: Board
#_direction: int = 1
#_alive: boolean = true
#_moving: boolean = false
timeAfter: int = 40

+<<abstract>>update(): void +<<abstract>>render(Screen screen): void #<<abstract>>calculateMove(): void #<<abstract>>kill(): void +<abstract>>afterKill(): void #<<abstract>>canMove(): boolean #getXMessage(): double #getYMessage(): double

Bomber

-_bombs:List<Bomb> #_input: Keyboard -planted: Sound # timeBetweenPutBombs: int = 0

+<<constructor>>Bomber(int x, int y, Board board) +update(): void +render(Screen screen): void +calculateXOffset(): void -detectPlaceBomb(): void #placeBomb(int x, int y): void -clearBombs(): void +kill(): void #afterKill(): void #calculateMove(): void +canMove(double x, double y): boolean +move(double xa, double ya): void +collide(Entity e): boolean -chooseSprite(): void + updateBombCollision(): void

Enemy <<abstract>>

Balloon

+<<constructor>>Balloon(int x, int y, Board board) #chooseSprite(): void