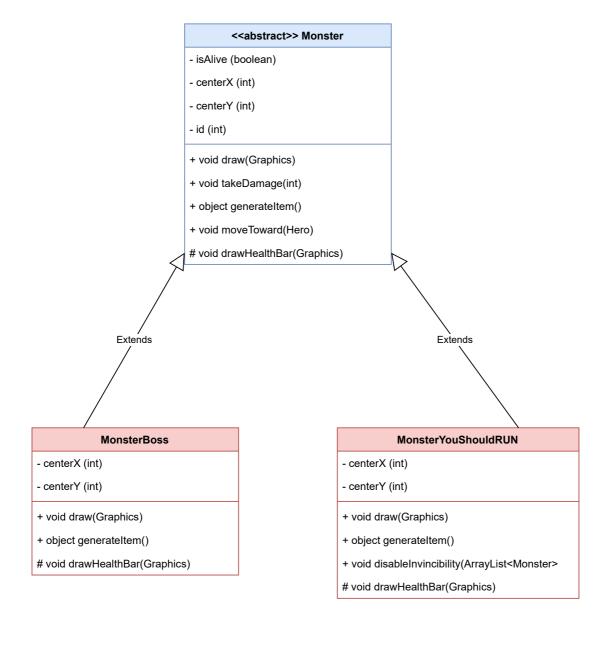
#### **PLAYER**

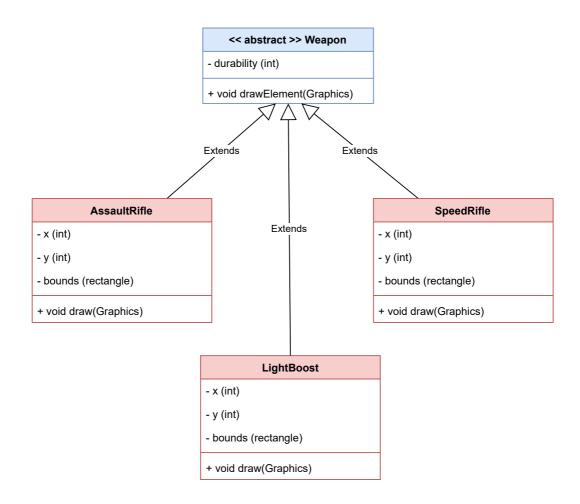
<< abstract >> Entity

## - x (int) - y (int) - centerX (int) - centerY (int) Hero - alive (boolean) - health (int) - invincible (boolean) - collisionHero (rectangle) + void draw(Graphics) - centerX (int) + void takeDamage(int) - centerY (int) # void activateInvincibility() - isSpeedRifleEquipped (boolean) # void disableInvicibility() Extends - isAssaultRifleEquipped (boolean) # void drawHealthBar(Graphics) - invincibilityTimeRemaining (int) + void applySpeedBoost(int, int) <<abstract>> Monster + void activateInvincibility() Extends - isAlive (boolean) + void unequipSpeedRifle() - centerX (int) + void equipAssaultRifle() - centerY (int) + void equipSpeedAssaultRifle() - id (int) + void draw(Graphics) + void unequipAssaultRifle() + void draw(Graphics) + void heal(int) + void takeDamage(int) + boolean isNear(Monster, int) + object generateItem() + void stopMovement() + void moveToward(Hero) + void move(set<Integer>) # void drawHealthBar(Graphics) - void updateFrame() # drawHealthBar(Graphics) <<Interface>> Player + void takeDamage(int) + int x() + int y() + int centerX() + int centerY() + boolean alive()

## **PLAYER**

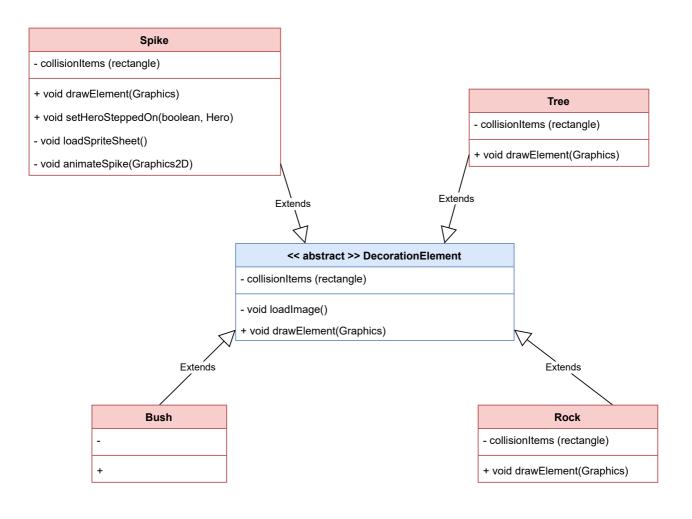


## **WEAPON**

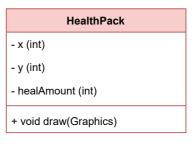


# Projectile

- x (int)
- y (int)
- dx (int)
- dy (int)
- bounds (rectangle)
- bounds (rectangle)
- + void draw(Graphics)



# SpeedBoost - x (int) - y (int) - duration (int) - speedIncrease (int) + void draw(Graphics)



#### MAIN

# GameOverScreen

- -
- + void saveScore()
- +string readScores()
- + list<int[]> loadScores()
- + void end(Graphics)

#### MenuPanel

- -
- void startGame()
- # void paintComponent(Graphics)

# GamePanel

- gameSpeed (boolean)
- + void keyTyped(KeyEvent)
- + void gestionDamage(int)
- + void startGame()
- Monster findClosestMonsterWithinRange(int)
- void handleCollision(Hero, DecorationElement)
- void nextLevel()
- void delayBeforenextShot(int)
- + void actionPerformed(ActionEvent)
- + void keyReleased(KeyEvent)
- + void addMonster(Monster)
- + void keyPressed(KeyEvent)
- void checkCollision()
- # void paintComponent(Graphics)

#### Level

- rocks (ArrayList<Rock>)
- monsters (ArrayList<Monster>)
- trees (ArrayList<Tree>)
- spikes (ArrayList<Spike>)
- bigBoss (MonsterYouShouldRun)
- bushes (ArrayList<Bush>)
- void addRandomDecorations(Hero, int)
- void generateBaseDecorations()
- void removeRandomDecoration()
- + point generateMonsterPosition()
- + boolean lvl up()
- + void generateRandomDecorations()
- + void generateRandomMonster()
- + void updateDecorations(Hero)