*Package: templeraider.entity*

**Entity** *(abstract)*

+private int xpos

+private int ypos

+Entity(int xcoord, int ycoord): ctor

+*abstract*getClassID(): String

+getImage(): Image

+isPassable(): Boolean

+onCollide(Entity ent): void

+getX(): int

+getY(): int

+distance(Entity ent): double

+think(long time): void

+setNextThink(long time): void

+remove(): void

*Package: templeraider.gui*

**GameCanvas**

+loadTileImages(): void

+setTileImage(int tileId, String str): void

+titleScreenShown(): boolean

+setTitleScreenShown(boolean shown): void

+paint(Graphics g): void

*Package: templeraider.gui*

**RPGMap**

+mapGrid: int[][]

+RPGMap(int level): ctor

+isPassable(int x, int y): boolean

*Package: templeraider.gui*

**RPGGUI**

+private GameCanvas gameCanvas

+isKeyDown(int key): boolean

+printLine(String line): void

+keyPressed(KeyEvent e): void

+keyReleased(KeyEvent e): void

+keyTyped(KeyEvent e): void

+redrawMap(): void

+getGameCanvas(): GameCanvas

+enableButton(int a): void

+disableButton(int a): void

+isGamePaused(): boolean

*Package: templeraider*

**Game**

+GAME\_TITLE=”Temple Raider”

+getInstance(): Game

+startNewGame(): void

+nextMap(): void

+getTimerSpeed(): int

+pauseTimer(boolean pause): void

+getEntities(): Entities

+getGUI(): RPGGUI

+getMap(): RPGMap

+getCombat(): Combat

+getEndmessage(): String

+isGameEnd(): boolean

+getTime(): long

+update(): void

+main(String[] args): void

+actionPerformed(ActionEvent event): void

+loseGame(): void

+winGame(): void

*Package: templeraider.entity*

**Entities**

+private ArrayList<Entity> entityList

Holds all the entities

+addEntity(Entity): void

+removeEntity(Entity): void

+removeAll(): void

+removeAll(String classid): void

+getAll(): Entity[]

+countByClass(String classid): int

+getByClass(String classid): Entity[]

+isEntInPos(x,y): Boolean

+getEntInPos(x,y): Entity

+allThink(): void

*Package: templeraider.combat*

**Combat**

+isInCombat(): boolean

+getPlayer(): EntityPlayer

+getEnemy(): EntityCombat

+startCombat(EntityPlayer player, EntityCombat enemy, boolean playerFirst): void

+startTurn(): void

+endTurn(): void

**BatUpDown**

*extends EnemyBat*

+think(long time): void

Enemy AI

**BatLeftRight**

*extends EnemyBat*

+think(long time): void

Enemy AI

**EnemyBat**

*extends EntityEnemy*

+EnemyBat(int x, int y): ctor

Set up stats

+doTurn(): void

Combat Behaviour

**EnemyMummy**

*extends EntityEnemy*

+EnemyMummy(int x, int y): ctor

Set up stats

+think(long time): void

Enemy AI

+doTurn(): void

Combat Behaviour

**EnemyTempleBandit**

*extends EntityEnemy*

+EnemyTemleBandit(int x, int y): ctor

Set up stats

+think(long time): void

Enemy AI

+doTurn(): void

Combat Behaviour

**EnemyForestBandit**

*extends EntityEnemy*

+EnemyForestBandit(int x, int y): ctor

Set up stats

+think(long time): void

Enemy AI

+doTurn(): void

Combat Behaviour

**EnemyTreeBandit**

*extends EntityEnemy*

+EnemyTreeBandit(int x, int y): ctor

Set up stats

+think(long time): void

Enemy AI

+doTurn(): void

Combat Behaviour

*Package: templeraider.combat*

**EnemyBarbarian**

*extends EntityEnemy*

+EnemyBarbarian(int x, int y): ctor

Set up stats

+think(long time): void

Enemy AI

+doTurn(): void

Combat Behaviour

**EnemyWolf**

*extends EntityEnemy*

+EnemyWolf (int x, int y): ctor

Set up stats

+think(long time): void

Enemy AI

+doTurn(): void

Combat Behaviour

**EnemyCultist**

*extends EntityEnemy*

+EnemyCultist(int x, int y): ctor

Set up stats

+think(long time): void

Enemy AI

+doTurn(): void

Combat Behaviour

**EnemyBoss**

*extends EntityEnemy*

+EnemyBoss(int x, int y): ctor

Set up stats

+think(long time): void

Enemy AI

+doTurn(): void

Combat Behaviour

**Combat**

+isInCombat(): boolean

+getPlayer(): EntityPlayer

+getEnemy(): EntityCombat

+startCombat(EntityPlayer player, EntityCombat enemy, boolean playerFirst): void

+startTurn(): void

+endTurn(): void

**EntityEnemy**

*extends EntityCombat*

+onCollide(Entity ent): void

Start Combat

+think(long time): void

Enemy AI

**EntityPlayer**

*extends EntityCombat*

+onCollide(Entity ent): void

Start Combat

+think(long time): void

Get Player Input

+giveXP(int xp): void

+getPlayerName(): String

+levelUp(): void

+healhPlayer(): void

+getImage(): Image

Player Animations

**EntityCombat**

*extends EntityMovable*

+getCombatName(): String

+getMaxHealth(): int

+getHealth(): int

+setHealth(int): void

+getRewardedXP(): int

+inflictDamage(int): void

+getHitChance(): double

+getCriticalChance(): double

+getDefense(): double

+getDamage(EntityCombat defender): int

+attack(EntityCombat defender): void

+defend(): void

+doTurn(): void

+drawHealthBar(Graphics g,int xpos,int ypos,int width,int height, float colR, float colG, float colB): void

*Package: templeraider.combat.enemies*

*Package: templeraider.entity*

**EntityPlayerSpawn**

*extends Entity*

Used to specify where the player spawns on a map

**EntityHealthFountain**

*extends Entity*

+onCollide(Entity ent)

Heal the player

**EntityStairs**

*extends Entity*

+onCollide(Entity ent)

Load the next level

**EntityMovable**

*extends Entity*

+moveTo(int x,int y): Boolean

+moveUp(): Boolean

+moveDown(): Boolean

+moveLeft(): Boolean

+moveRight(): Boolean

+isMoving(): Boolean

+getMoveSpeed(): int

+setMoveSpeed(int speed): void

+getInterpolationX(): double

+getInterpolationY(): double

**Entities**

+private ArrayList<Entity> entityList

Holds all the entities

+addEntity(Entity): void

+removeEntity(Entity): void

+removeAll(): void

+removeAll(String classid): void

+getAll(): Entity[]

+countByClass(String classid): int

+getByClass(String classid): Entity[]

+isEntInPos(x,y): Boolean

+getEntInPos(x,y): Entity

+allThink(): void

**Entity** *(abstract)*

+private int xpos

+private int ypos

+Entity(int xcoord, int ycoord): ctor

+*abstract*getClassID(): String

+getImage(): Image

+isPassable(): Boolean

+onCollide(Entity ent): void

+getX(): int

+getY(): int

+distance(Entity ent): double

+think(long time): void

+setNextThink(long time): void

+remove(): void

*Package: templeraider.gui*

**RPGGUI**

+private GameCanvas gameCanvas

+isKeyDown(int key): boolean

+printLine(String line): void

+keyPressed(KeyEvent e): void

+keyReleased(KeyEvent e): void

+keyTyped(KeyEvent e): void

+redrawMap(): void

+getGameCanvas(): GameCanvas

+enableButton(int a): void

+disableButton(int a): void

+isGamePaused(): boolean

**RPGMap**

+mapGrid: int[][]

+RPGMap(int level): ctor

+isPassable(int x, int y): boolean

**GameCanvas**

+loadTileImages(): void

+setTileImage(int tileId, String str): void

+titleScreenShown(): boolean

+setTitleScreenShown(boolean shown): void

+paint(Graphics g): void