

| Entity (abstract)                     |   |
|---------------------------------------|---|
| # currentCell: Cell                   | the Cell that this entity is currently occupying      |
| + Entity(Cell)                        | constructor which takes in a Cell to make currentCell |
| + getCurrentTile():<br>void           | returns this Entity's currentTile                     |
| + setCurrentTile(Tile<br>currentTile) | sets this Entity's currentTile to the parameter       |

| Character (extends Entity)                  |  |
|---|--|
| # health: double                            | this Character's current health  |
| # maxHealth: double                         | this Character's maximum health  |
| # attack: double                            | this Character's attack  |
| # defence: double                           | this Character's defence   |
| # armor: Item                               | this Character's equipped armor Item   |
| # weapon: Item                              | this Character's equipped weapon Item  |
| # race: Race                                | this Character's Race  |
| # effects: Effect[]                         | an array of this Character's current Effects   |
| # gold: int                                 | the amount of gold held by this character  |
| # potionPower: double                       | increases the effectiveness of potions, influenced by Race, Items, and Effects             |
| # healthOnKill: double                      | the amount of health gained when killing an enemy, influenced by Race, Items, and Effects  |
| # goldOnKill: int                           | the amount of gold gained when killing an enemy, influenced by Race, Items, and Effects    |
| # turnController:<br>TurnConroller          | an TurnController object that Character's use to start and end their turns                 |
| + Character(Cell<br>currentCell, Race race) | constructor which takes in a Cell to be passed into super() and a Race, which race becomes |
| +   | takes in a TurnController object so the  |

|   |  |
|---|--|
| startTurn(TurnController): void   | Character can pass the turn to the next Character at the end of their turn, calls updateStats() and applyEffects()   |
| + endTurn(): void   | called when this Character is done with their turn, calls decrementEffectDurations(effects), indicates to turnController that their turn is over                     |
| + static doDamage(Character caster, Character target, DamageType damageType, double damage): void | static method that handles the dealing of damage between characters, passes different values into takeDamage() depending on damageType when it's called              |
| + takeDamage(double amount, Character damager): void  | decrements health by the amount parameter, if this kills the character, attempt to call targetKilled(damager)  |
| + targetKilled(): void  | apply this character's xxxOnKill variables   |
| + takeHeal(double amount): void   | increments health by the amount parameter, up to maxHealth   |
| + addEffect(Effect effect): void  | Adds the given effect Effect to this character's effects array   |
| + updateStats(): void   | update this character's stats based on base values, racial bonuses, item bonuses, and stat-changing effects, calls effects[i].applyStatChange() for all i in effects |
| + applyEffects(): void  | call effects[i].applyEffect() for all i in effects   |
| + decrementEffectDurations(Effect[] effects): void  | call effects[i].decrementDuration() for all i in effects, destroy effects with duration 0  |
| + modStat(double amount, Stat stat): void   | increment the given Stat by the given amount   |
| + getAttack(): double   | returns this character's attack  |
| + getDefence(): double  | returns this character's defence   |

| Item (extends Entity)  |   |
|--|---|
| ItemType {WEAPON, ARMOR, POTION}: enum   | enumerator indicating whether this item is a weapon, armor, or potion   |
| itemType: ItemType   | this Item's ItemType  |
| healthMod: double  | the change in health this Item causes   |
| attackMod: double  | the change in attack this Item causes   |
| defenceMod: double   | the change in defence this Item causes  |
| + Item(Cell currentCell, ItemType itemType, double healthMod, double attackMod, double defenceMod) | constructor that takes in the cell this item should occupy, this Item's ItemType, and its health, attack, and defenceMods, calls super(currentCell) |

| Race  |  |
|---|--|
| # raceName: String                              | the name of this Race  |
| # healthMod: double                             | the change in health this Race causes  |
| # attackMod: double                             | the change in attack this Race causes  |
| # defenceMod: double                            | the change in defence this Race causes   |
| # raceEffect: Effect                            | the Effect applied to Characters of this Race  |
| + getRaceName(): String                         | returns the name of this Race  |
| + getHealthMod(): double                        | returns this character's healthMod   |
| + getAttackMod(): double                        | returns this character's attackMod   |
| + getDefenceMod(): double                       | returns this character's defenceMod  |
| + <i>getEffect(Character character): Effect</i> | returns a new Effect, implemented differently for each Race, character is the character effected |

| Effect                             |  |
|------------------------------------|--|
| # affectedCharacter: Character     | the Character that this Effect is applied to   |
| # caster: Character                | the Character that caused this Effect to be applied  |
| # effectName: String               | the name of this Effect  |
| # magnitude: double                | the magnitude of this Effect   |
| # duration: int                    | the remaining turns that this Effect will be active for  |
| # isPermanent: boolean             | signifies whether this Effect is permanent or not, set to true if a duration of -1 was passed into the constructor |
| # healthMod: double                | the change in health this Effect causes  |
| # attackMod: double                | the change in attack this Effect causes  |
| # defenceMod: double               | the change in defence this Effect causes   |
| + <i>applyEffect(): void</i>       | applies any non-stat-change components of this Effect  |
| + <i>applyStatChange(): void</i>   | applies any stat-changes caused by this Effect   |
| + <i>decrementDuration(): void</i> | decrements duration by 1 if not isPermanent  |
| + <i>getDuration(): int</i>        | returns duration   |

| Stat (enum)    |                             |
|----------------|-----------------------------|
| MAX_HEALTH     | corresponds to maxHealth    |
| ATTACK         | corresponds to attack       |
| DEFENCE        | corresponds to defence      |
| POTION_POWER   | corresponds to potionPower  |
| HEALTH_ON_KILL | corresponds to healthOnKill |
| GOLD_ON_KILL   | corresponds to goldOnKill   |

| DamageType (enum) |  |
|-------------------|--|
| PHYSICAL          | doDamage() calls<br>target.takeDamage(Math.ceil((100 / (100 + target.getDefence())) * damage), caster) |
| ELEMENTAL         | doDamage() calls target.takeDamage(damage, caster) until some sort of magic defence is implemented     |

| Cell                                    |   |
|---|---|
| - gridX: int                            | this cell's x position on the Floor grid  |
| - gridY: int                            | the cell's y position on the Floor grid   |
| - tileTexture: BufferedImage            | the texture gained from the Tile absorbed by this Cell  |
| - occupiable: boolean                   | whether or not this cell can be occupied by entities, gained from absorbed Tile   |
| - occupant: Entity                      | the Entity occupying this Cell  |
| + Cell(int gridX, int gridY, Tile tile) | constructor that takes in an x and y grid position and a Tile from which tileTexture and occupiable are absorbed  |
| + getOccupant(): Entity                 | returns occupant  |
| + setOccupant(Entity occupant): void    | sets occupant to the parameter Entity if this Cell doesn't already have an occupant and if this Cell is occupiable, calls occupant.setCurrentTile(this) |
| + getSpaceOpen(): boolean               | returns true if this Cell doesn't have an occupant and this Cell is occupiable, otherwise returns false   |
| + getOccupiable(): boolean              | returns occupiable  |
| + getGridX(): int                       | returns gridX, may not be needed  |
| + getGridY(): int                       | returns gridY, may not be needed  |

