Dungeons Without Dragons Explained

Austin Brown and Daniel Paredes

The **Character** class is the class for any character in the game. Every character has a:

* State
* Weapon (optional)
* List of modifiers (is empty if the character has none)
* Damage attack() which is the total damage it can deal to an enemy (the weapon's damage + character's individual damage)
* Upkeep function that applies any Boons or Curses the Character has
* Heal function
* takeDamage function changes the characters state depending on the input damage parameter

The **Player** class is derived from the character class, this is the main character/player of the game.

This class has everything that the Character class as well as a:

* name, can be chosen, or by default it is “Stevie”
* Level, an int variable that keeps track of the players level and increases when the experience bar is full
* Experience, an int variable that tracks the amount of experience a player has,
* Boolean hasKey that is True if the Player has the key, and false otherwise, if this Boolean becomes true, the game is won
* levelup() function that is called when player levels up, this function randomly gives the player the key
* Upkeep is the same as the Parent class but also updates player level and can call levelup()

The **NPC** (non-player character) class is derived from the character class, this is any character that is not the player or a monster. The NPC class has everything a character has as well as name, by default it is “Gustav”. All NPC’s are allies to the player.

The **Monster** class is derived from the character class. It is an enemy of the player. The monster class has everything that a character has as well as a name, by default it is “Carl”.

The **Weapon** class is the class for any weapon in the game. This class has:

* Damage attack() which is the total damage it can deal
* Type Advantage which does bonus damage against certain enemies

The **Damage** class is an amount of damage, this class has:

* Integer damage, the minimum amount of damage
* A list of Modifiers, typeadvantages. These modifiers can increase the damage passed to certain objects that share the same modifiers
* Damage +Operators, this allows multiple Damages to be combined for one attack, for example the Player’s Damage + Player’s Weapon Damage + Ally’s Damage.
* Integer numMods, the number of modifiers a damage has.

The **State** class contains information that describes the character’s health and damage that it can deal (without a weapon). This class has:

* Integer baseHealth, the health that a character currently has
* Integer maxHealth, the maximum health a character can have
* Integer healthMod, the amount of health that is given or taken from the upkeep() function
* Damage that the character can deal without a weapon
* Integer damageMod, the amount of damage reduced or added to the character

The **Modifier** class: The base class for all modifiers. This class has:

* Integer Effect that determines how much this modifies a stat
* Integer Duration that determines how many “Turns” it will stay in effect
* Upkeep function that decreases duration by 1 and returns true if the duration is less than 0. This is called inside of a character’s upkeep function.
* Apply function that modifies an integer of a state class, called in character upkeep functions

The **DoT** class: This class is a modifier that will harm a character over time. This class has:

* Overridden apply function

The **Curse** class: This class is a modifier that will lower the attack of a character. This class has:

* Overridden apply function

The **Boon** class: This class is a modifier that will heal a character over time. This class has:

* Overridden apply function

The **TypeAdvantage** class: This class is a modifier that will increase the damage of a character. This class has:

* Overridden apply function
* A string member called type that determines if the effect is applied.

The **Main** is where the game mechanics happen and all the classes are actually implemented. The main has the following basic functions:

* Int Menu, this function displays a menu of options to the character and allows them to choose between 1. Look around the room. 2. Go to the next room. 3. Display stats for the party. The choice is stored in an integer.
* Void Fight(), this function simulates a fight in the game. First the player attacks the monster, then the player’s ally attacks. Then the game checks to see if the monster is dead. If it is, then the fight ends and player gains experience. If not, the monster attacks either the player or the ally. This is repeated until the monster dies.

The main starts with an introduction text at the start of the game. The player chooses his own name. The player starts with no allies by default. The player is then given the menu options. If the player decides to look around the room, a player may either find an item, monster, or nothing. If the player finds an item, the player will be healed. If the player finds a monster, the fight function will be called. If the player chooses to go another room, the player may gain an ally, find a monster, or nothing may happen. If the player chooses to view the stats, the game will output the statistics (health, damage, etc.) for the player and its allies. The player gains experience and levels up every time he kills a monster. The objective of the game is to keep playing and leveling up until the player finds the key. Once the key is found, the player wins the game.