

# dungeon\_dudes design

WO1 Clayton E. Williams

October 2023

## 1 Overview

**dungeon\_dudes** is a text-based RPG that users interact with at the command line. A majority of the game has already been written. The goal of this project is to use the provided guide for how different characters and monsters interact, and develop a module to implement a character or monster for game play.

The intent is to develop a monster class that can be seamlessly integrated to the overall project, with no degradation to gameplay. Crucial to the design plan will be to not touch any shared modules or packages, and only make changes to the `humanoid.py` as the base class for humanoid monsters, and the humanoid subclass `monsters`, `bandit.py`, `murloc.py`, and `ogre.py`

## 2 Class Requirements

There are 4 classes that will need to be written to implement the humanoid monsters: `humanoid`, `ogre`, `bandit`, and `murloc`.

- **Humanoid**

Humanoid is the base class for three humanoid monsters, which will inherit from the `Monster` class, which inherits from the `Combatant` class. Because each humanoid monster is able to use healing potions, `Humanoid` will need to initialize with 1 healing potion, and have functionality for healing. The healing function will update the hit points, as well as update number of healing potions the monster has.

The `take_turn`, `take_damage`, `get_skills`, and `get_skills_list` functions will need to be written based on the manual as a subclass of `Combatant`.

All basic attacks from the Humanoids result in Physical damage type.

For actions that are based on percentages, use a `random.randint(0, 100)` and compare against the decimal of the percentage. EX. if an attack has a 30 percent chance, the logic would be: `if random.randint(0, 100) < 30`

- **Ogre**  
 Ogres can be of two types, 'Blood-Thirster' and 'Ogre-Magi' and will be selected randomly using `random.choice`. Both types share basic skills, Ogre Strategic Thinking, Spiked Potions, Improved Spiked Potions, and Attack, as well as Strength and Agility stats growth. However, they each possess their own special actions that require determining which type of Ogre the instance belongs to.
  - **Blood-Thirster**  
 Unique actions to the Blood-Thirster Ogre are Frenzy, Risky Blow, and Improved Frenzy, at levels 1, 1, and 10 respectively. Risky Blow is the only non-passive action, so Blood-Thirsty Ogre actions will be chosen from Attack or Risky Blow.
  - **Ogre-Magi**  
 Unique actions to the Ogre-Magi Ogre are Wild-Magics, Of 2 Minds, and Improved Ogre Strategic Thinking. Of these, Improved Ogre Strategic Thinking is the only passive action, so Ogre-Magic action will be chosen from a list of Attack, Wild-Magics, and Of 2 Minds.
- **Murloc**  
 The Murloc monster represents a group of monsters, starting at 4, where 1 member executes a special action while the remaining attack. To achieve this, a list will be constructed with  $n - 1$  attacks, and a random choice from the list of available actions, Ice Bolt, Forage, Poisons, and Holy Nova. The only passive action results in an increase in Murlocs to 5 at level 10. When an action is selected, it will be popped from the list to prevent it from being executed again.
  - **Ice Bolt**  
 This action can be executed twice. To easily replicate this, the list of actions will contain 2 callables to `ice.bolt()`.  
 This action deals Ice damage based on Intelligence.
  - **Forage**  
 This action can be executed twice per battle as well and follows the same setup as Ice Bolt. The result of this is an additional healing potion, which will need to be updated in the Humanoid healing\_potions.
  - **Poisons**  
 This action can be executed only once per battle. Once used, all subsequent attacks have a 30 percent chance of doing 30 percent more damage as Poison damage. This action is available at level 3.
  - **Holy Nova** At level 10, this action becomes available. It can only be used once per battle, heals the tribe by 30 percent and does Holy damage equal to Intelligence.
- **Band of Bandits**  
 Similar to Murloc, the Band of Bandits also simulates a group, or band

of similar monsters, starting at 3, where one Bandit attacks, and the remaining bandits execute an action. The list of available actions are Fireball, Dirty Tricks, Rallying Cry, and Bless. Each are able to be used once or twice per battle and follow the same algorithm for determining if it has been used and removing it from the list of available actions.

- Fireball  
This action can be used twice per battle and results in an Attack dealing Fire damage based on Intelligence.
- Dirty Tricks  
This action can be used only once per battle. It is an Attack at 75 percent power, but reduces the Physical Damage Defensive Multiplier. This action requires CombatAction to have two tuples, one for Attack, and one for Aura to modify the Defensive Multiplier.
- Rallying Cry  
This action can be used only once per battle and increases the tribes Physical Damage Offensive Multiplier by 10. The CombatAction required for this action is Battle Cry. This action is available at level 8.
- Bless  
This action can be used only once per battle and improves all Defensive Damage Modifiers by 10 and heals the band by 50 percent. Bless is achieved with a Battle Cry CombatAction. This action is available at level 15.

All actions are accomplished using the CombatAction. Attacks with the "Attack" action, modifiers to characters with "Aura", and modifiers to the Monsters with "Battle Cry".

### 3 UML

Humanoid
healing_potions hit_points printer : CombatPrint
attack(): CombatAction base_att_def_power() healing_potion(character) humanoid_damage(damage: float): int modify_damage(damage): int take_damage(damage: int, dmg_type, message: str): bool take_turn(): CombatAction

Figure 1: Enter Caption

Bandit	Murloc	Ogre
damage_modifiers defense_modifiers hit_points stats_structure : Dict[str, Tuple[int]]  bless() dirty_tricks(): Tuple fireball(): Tuple get_action() get_skills(): Dict[str, 'function'] get_skills_list(): list rallying_cry() take_turn(): CombatAction	base_attack damage damage_modifiers damage_type defense_modifiers hit_points poisoned stats_structure : Dict[str, Tuple[int]] tribe_size  forage() get_action() get_base_att() get_skills() get_skills_list() holy_nova() ice_bolt() poisons() take_turn()	damage damage_modifiers defense_modifiers ogre_type ogre_types : list stats_structure : Dict[str, Tuple[int]]  attack() blood_thirster_attack() bloodlust() frenzy() get_skills() get_skills_list(): list improved_frenzy() risky_blow(): CombatAction special_skill() strategic_thinking() take_turn(): CombatAction wild_magics()

Figure 2: Enter Caption