Terminal RPG

Simple terminal-based role-playing game

What is an RPG?

Features

- Shop to buy/sell in-game items
- Saving progress
- Unique locations
- Battle mechanics

Walkthrough

Controls

Navigate the menu using

- Arrow-keys
- Vim keys (J & K)

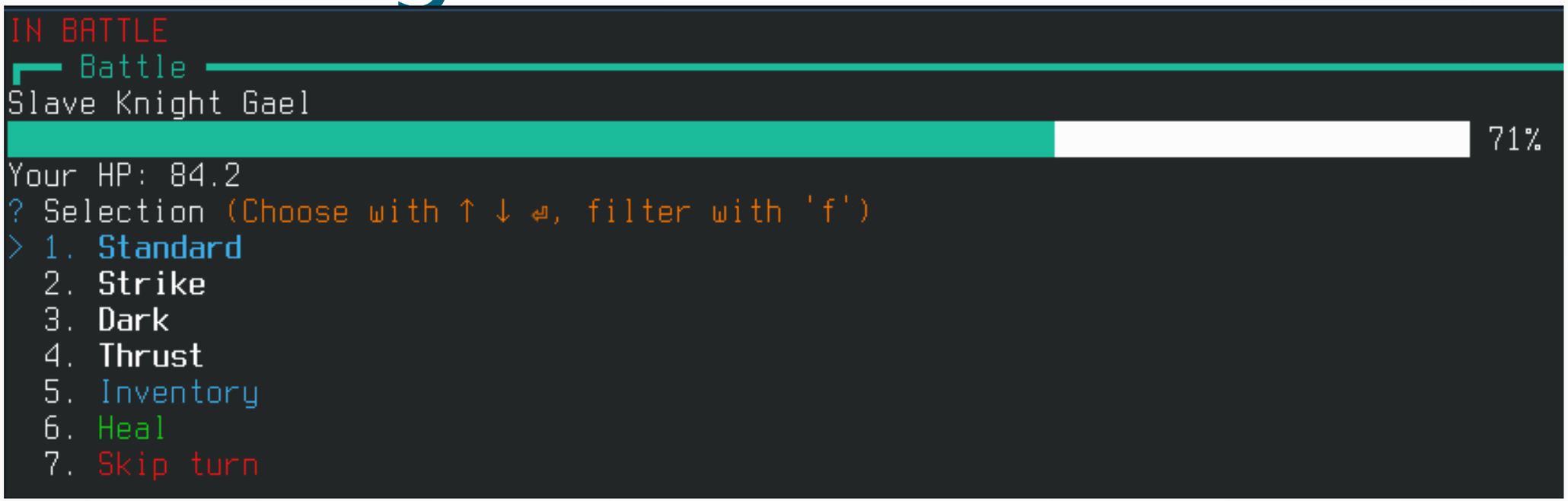
How to play

- Players start at the Hub where they can view their inventory or access the warp menu to travel between locations.
- Players can access the Battle option from the Arena location where they can then initiate battle with a chosen opponent.
- Players can access the Shop NPC after warping to the Shop location to buy and sell items to use in battle.

Battle mechanics

- Boss attack is selected at random.
- Different attacks have different attack damage multipliers.
- Attacks can miss.
- Healing or viewing your intventory during battle will take up one turn.

Battling



- Players can choose an attack, with each attack having a unique damage multiplier.
- Players may heal during battle, consuming one health-potion each time.

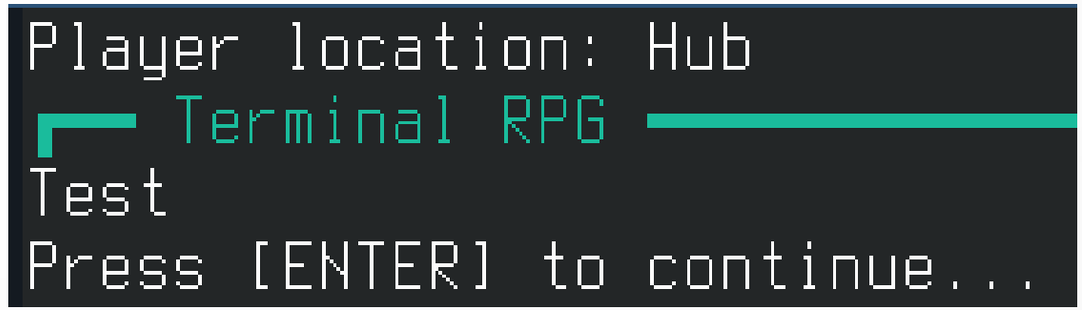
UI structure

```ruby
def main_frame()
    clear_screen()
    # we should print player location for each new main frame
    puts "Player location: " + @player.location.to_s().blink()
    # we're going to open the frame in blockless mode
    CLI::UI::Frame.open(title)
    # past this point is technically dangerous if we do not rem
        yield()

    # if we get here
    pop_frame()
end
```

```ruby
main_frame {
    puts "Test"
    continue_prompt()
}
```

- CLI/UI frames are created and destroyed each time main_frame is called and passed a code-block. The code-block is then what gets displayed in the frame.



Player inventory

Player and NPC inventory is stored in a hash.

```
@inventory = {
  # item => { item_str => stock }
  :sword => { "Sword" => 1 },
  :health_potion => { "Health Potion" => 5 }
}
```

Next Slide....

Shop logic

```ruby
def show_shop_screen()
    pop_frame()
    # we should iterate over the npc's inventory to display options
    main_frame {
        # prompt shop items
        CLI::UI::Prompt.ask("Shop Items") { |handler|
            # the inventory hash is nested. eg. { :health_potion => { "Health Potion" => 5 } }
            @shop_npc.inventory.each() { |key_sym, value|
                # access nested hash { "Health Potion" => 5 } with the key_sym
                @shop_npc.inventory[key_sym].each { |key, value|
                    # display each item
                    handler.option("Item: #{key} | Quantity: #{value} | Price: #{@shop_npc.price_list[key_sym]}") { |selection|
                        # if player gold - the price of item is greater than or equal to 0 && item is in stock
                        allowed_to_buy = (@player.gold - @shop_npc.price_list[key_sym] >= 0 && @shop_npc.inventory[key_sym][key] > 0)
                        if(allowed_to_buy)
                            @player.gold -= @shop_npc.price_list[key_sym]
                            @player.inventory[key_sym] += 1
                            @shop_npc.inventory[key_sym][key] -= 1
                            puts "Remaining balance: #{@player.gold} | Stock left: #{@shop_npc.inventory[key_sym][key]}".blink()
                            sleep(2)
                            show_main_screen()
                        else
                            puts "Item out of stock!"
                            sleep(2)
                            show_shop_screen()
                        end
```

Don’t forget the flow chart

Improvements

- Attack cooldown and/or tied to a mana / stamina system.

- More opponents to battle.

- Range of new weapons and items to use at your disposal.

- Peer to peer battles.