Mark Boyle
Terminal App

Lands of Aralia

Game

- Inspired by ‘Choose your own adventure’ and ‘Dungeons and Dragons’
- They go on a quest for the Legendary Ruby Gemstone.
- On the quest they have to battle enemies and solve riddles and math problems.

Intro Text and Character Selection

Level 1

Battle:
Goblin

Level 2

Battle:
Troll

Battle:
Orc

Level 3

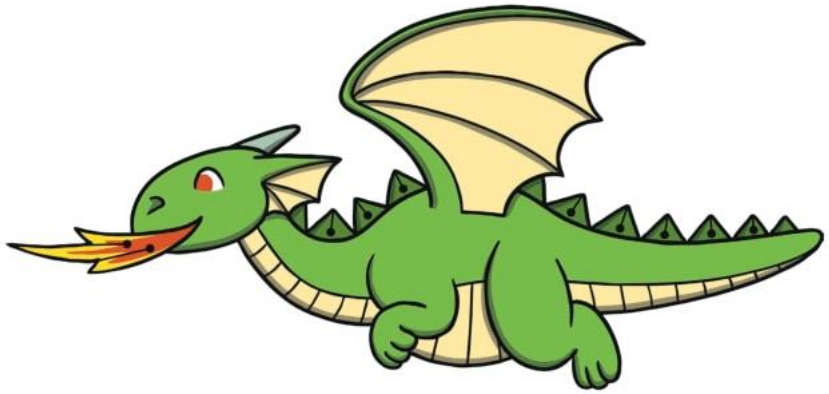
Riddles:
Witch

Math Problems:
Leprechaun

Level 4

Battle:
Dragon

How it works



- A Character class
- An Enemy class
- An object is created for each class with default values

- The user makes a name for the character and chooses the character type (Elf, Dwarf or Warrior)
- When they make the selection, a method is called to update the stats of the character.

```ruby
def select_character(prompt)
    puts ' '
    prompt.select('Please Select Your Character Type:') do |menu|
        menu.choice 'Elf'
        menu.choice 'Dwarf'
        menu.choice 'Warrior'
    end
end
```

Character Selection

```ruby
character_confirmation = 'No'

while character_confirmation == 'No'
    system 'clear'
    character_choice = select_character(prompt)
    case character_choice
    when 'Elf'
        character.update_elf_stats
    when 'Dwarf'
        character.update_dwarf_stats
    when 'Warrior'
        character.update_warrior_stats
    end
    character.display_character_info
    puts ' '
    character_confirmation = prompt.select('Are you happy with this choice?') do |menu|
        menu.choice "Yes"
        menu.choice "No"
    end
end
```

```ruby
def update_elf_stats
    @character_type = 'Elf'
    @sword_skill = 2
    @archery_skill = 5
    @armour_rating = 2
end

def update_dwarf_stats
    @character_type = 'Dwarf'
    @sword_skill = 4
    @archery_skill = 1
    @armour_rating = 4
end
```

- Variables are set at the start

- Level = 1
- Path = 1
- Lives = 3

- This determines which battle/riddle scene loop to go into.
- These variables are updated after each battle or pathway choice.
- Once in the right loop a method is called to update the enemy information.

Battle Scene

```
while level == 2 && lives > 0 && path == 1
    display_troll_intro
    enemy.update_troll_stats
    character.restore_health(level)

    while enemy::enemy_health > 0 && character::character_health > 0
        input = prompt.select("Choose your action:") do |menu|
            menu.choice "Use Sword"
            menu.choice "Shoot Arrow"
            menu.choice "Search Area"
            if level > 1 then menu.choice "Throw Spear" end
            if level > 2 then menu.choice "Swing Battleaxe" end
          end
        system 'clear'
        puts ' '
        case input
        when "Use Sword"
            enemy::enemy_health = character_attack(enemy::enemy_health, character.generate_sword_damage)
            puts ' '
        when "Shoot Arrow"
            enemy::enemy_health = character_attack(enemy::enemy_health, character.generate_archery_damage)
            puts ' '
        when "Throw Spear"
            enemy::enemy_health = character_attack(enemy::enemy_health, character.generate_spear_damage)
            puts ' '
        when "Swing Battleaxe"
            enemy::enemy_health = character_attack(enemy::enemy_health, character.generate_battleaxe_damage)
            puts ' '
        when "Search Area"
            character.search_area
        end
```

If the user decides to attack

```
def generate_sword_damage
    attack_damage = rand(30) + (@sword_skill * 3)
    attack_damage
end

def generate_archery_damage
    attack_damage = rand(30) + (@archery_skill * 3)
    attack_damage
end

def generate_spear_damage
    attack_damage = rand(10..40)
    attack_damage
end

def generate_battleaxe_damage
    attack_damage = rand(15..45)
    attack_damage
end
```

If the user decides to search area

```ruby
    def search_area
        search_number = rand(100)
        case search_number
        when (1..10)
            find_extreme_health_potion
        when (11..60)
            find_health_potion
        when (61..85)
            find_improved_armour
        when (86..100)
            puts "Sorry, you didn't find anything."
        end
    end

    def find_extreme_health_potion
        puts "You found an Extreme Health Potion! Your health goes up by 50."
        @character_health += 50
        puts "Your health is now at #{@character_health}."
        puts ' '
    end

    def find_health_potion
        puts "You found a Health Potion! Your health goes up by 25."
        @character_health += 25
        puts "Your health is now at #{@character_health}."
        puts ' '
    end

    def find_improved_armour
        puts "You found some better armour! Your armour rating goes up by 5."
```

Riddle Scene

```
while level == 3 && lives > 0 && path == 1
    system 'clear'
    display_witch_intro
    incorrect_answers = 0
    correct_answers = 0
    riddle_number = [1, 2, 3, 4, 5, 6]
    while incorrect_answers < 3 && correct_answers < 3
        puts ' '
        puts "Next Riddle:\n\n" unless incorrect_answers == 0 && correct_answers == 0

        riddle_selection = riddle_number.sample
        riddle_number.delete(riddle_selection)
        case riddle_selection
        when 1
            puts "What has legs but can't walk?"
            answer = gets.chomp.downcase.split
            puts answer
            puts ' '
            if answer.include?('chair')
                puts 'Correct!'
                correct_answers += 1
            elsif answer.include?('table')
                puts 'Correct!'
                correct_answers += 1
            else
                puts 'Incorrect!'
                incorrect_answers += 1
            end
```

User Engagement

- Can choose multiple pathways
- Randomly generated numbers means the game is always different
- Character selection and action selection add a level of strategy
- Choose name of character
- Earning new weapons encouraging user to play to the end

Features

- Combat Style
- Level Up (new weapons become available)
- Limited Lives
- Multiple Pathways
- Character Selection (Different character have different stats)

Challenges

- Keeping code DRY and efficient
- Using numbers that made the game fun to play

Favourite Parts

- Working out the right numbers to use
- Creating story content.

Gems Used

- Tty-prompt
- Artii
- Colorize

Tips/Strategies

- Collect armour early on against the goblin because it’ll improve the chances of beating the dragon at the end.