



# ARKANE ORKZ

RPG turned Idle turned  
Offline multiplayer  
Battler

# AN OVERVIEW

Arkane Orkz was originally conceived as an idea as text based adventure game inspired by Zork, using the rules of a recent TTRPG ruleset called Arcane Ugly, designed by Youtuber Miscast.



# AN OVERVIEW

Using the System Reference Document provided by Miscast for free use, I adapted the rules to fit what I would need for the game I had in mind.

THE ARCAINE UGLY  
SYSTEM REFERENCE  
DOCUMENT



AKA  
FREE STUFF TO HACK!

# THE CONCEPT



After multiple revisions of my concept during implementation, I settled on a more simple idea of a game where players can log in and fight other offline saves to level up and become the strongest.

# THE MAIN MENU

Utilises TTY Prompt

Battlin Orkz

(Battle other saves)

Save Ork

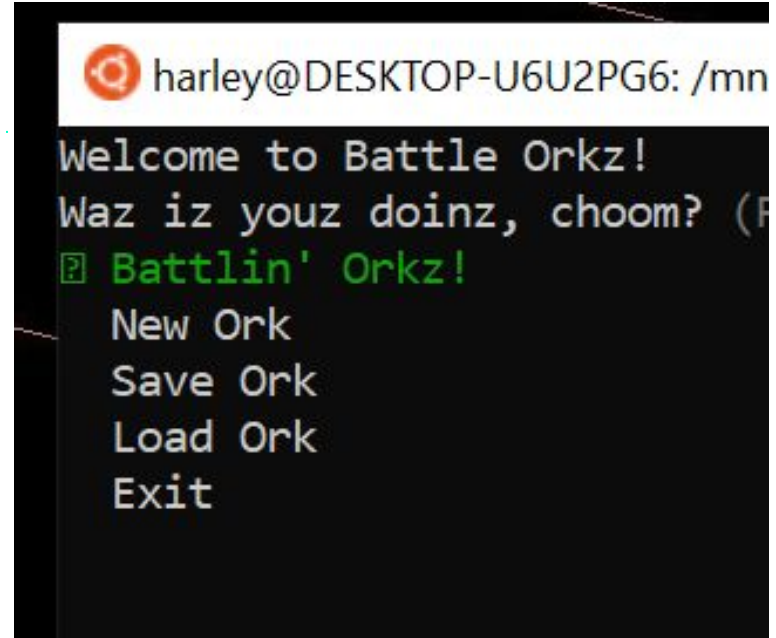
(Save current character)

Load Ork

(Load character from file)

Exit

(Exit the program)



```
harley@DESKTOP-U6U2PG6: /mn
Welcome to Battle Orkz!
Waz iz youz doinz, choom? (P
[?] Battlin' Orkz!
New Ork
Save Ork
Load Ork
Exit
```

# CHARACTER CREATION

```
What is your name?
```

```
Example's Stats:
```

Strength		5
Quickz		2
Smartz		2
Wackness		4
Hurtpoints		8

```
Are you happy with these stats? (Press ↑/↓ and Enter)
```

```
❑ No
```

```
Yes
```

Prompts user for name  
(to be implemented: if name  
is in use, rejects it and  
tells user to load instead)

Allows user to reroll stats  
indefinitely before  
accepting them

# CHARACTER CREATION CONT.

Pick your startin weapon: (Press 1)

<b>Zword</b>	<b>1d8</b>	<b>Strength</b>
Zpear	1d8	Dexterity
Ax'em	1d6+1	Strength
Stabber	1d6+1	Dexterity

Prompts user to pick from selection of starting weapons.

Currently the only difference between weapons is the damage and the stat used for attacks.

Example created.

Waz iz youz doinz, choom? (Press 1)

**Battlin' Orkz!**

New Ork  
Save Ork  
Load Ork  
Exit

Tells user the new character has been created and loads the main menu again

# SAVE CHARACTER

If the current character has never been saved, saves current character to the end of the save file

If there is a save with the same name present, the game prompts you to overwrite it.

Saying no prompts the user to give their character a new name.

```
Example waz zaved.  
Waz iz youz doinz, choom? (  
[?] Battlin' Orkz!  
    New Ork  
    Save Ork  
    Load Ork  
    Exit
```

```
Example waz zaved.  
Waz iz youz doinz, choom? Save Ork  
Do you want to save over Example | Level: 1  
(Press ↑/↓ arrow to move and Enter to select)  
[?] No  
    Yes
```

```
Example waz zaved to da phial.  
Please choose a new name:
```



# LOAD CHARACTER

```
Try againz when you'ze a real Ork.  
Waz iz youz doinz, choom? Load Ork  
Grog | Level: 1  
Forgromer Bashsword | Level: 1  
Polly Eifelhammer | Level: 1  
Example | Level: 1  
FF | Level: 1  
Oi! Which Orkz iz you?  
_
```

Lists all the current saves in the save file and prompts user to input the name of a file to load.

If a name is submitted that isn't in the save file, resets to main menu

(currently can load "" save when fed random characters)

## MAIN MENU CODE

Utilises TTY Prompt  
to get input

Processed using  
When  
Case

```
main_menu
```

```
main_selection = $prompt.select("W
```

```
    menu.choice "Battlin' Orkz!"
```

```
    menu.choice "New Ork"
```

```
    menu.choice "Save Ork"
```

```
    menu.choice "Load Ork"
```

```
    menu.choice "Exit"
```

```
end
```

```
return main_selection
```

```
class Character
```

```
  attr_reader :name, :str, :dex, :int, :w  
  attr_accessor :hp
```

```
  def initialize(name, str, dex, int, wrd  
    @name = name  
    @str = str  
    @dex = dex  
    @int = int  
    @wrđ = wrđ  
    @hp = hp  
    @armor = armor  
    @ward = ward  
    @damage = damage  
    @atk = atk  
    @level = level  
    @exploration = exploration  
    @money = money  
  end
```

## CHARACTER.RB / CHARACTER OBJECT

Holds all relevant data for player characters and monsters for all current purposes.

When player loads a character, they are loaded to \$player and become globally accessible.

## CREATING A CHARACTER OBJECT

Def create\_character() is within index.rb.

Manual parameter is used to determine whether full character creation is needed.

Named parameter is used to give quick generated characters names.

```
def create_character(manual, named)  
  #Manually make the character
```

```
  if manual == -1  
    #Manually make the character.
```

```
  elsif manual == 0  
    #Quickly autogenerate a character
```

```
  #Use name from parameter if provided, otherwise get a random one.  
  if named != ""  
    name = named  
  else  
    name = Faker::Games::ElderScrolls.name  
  end
```

# CHARACTER.RB / CHARACTER OBJECT

Basic Functions:

To\_s

File\_show

Stats

Assorted uses for showing different stats of the character object to the player.

```
def to_s
  return @name
end

def file_show
  return "#{@name} | Level: #{@level}"
end

def stats
  puts "Strength\t|\t#{@str}"
  puts "Quickz\t|\t#{@dex}"
  puts "Smartz\t|\t#{@int}"
  puts "Wackness\t|\t#{@wrđ}"
  puts "Hurtpoints\t|\t#{@hp}"
  puts "Armoredness\t|\t#{@armor}"
  puts "Magikked\t|\t#{@ward}"
  puts "Ork Level\t|\t#{@level}"
end
```

## CHARACTER.RB / CHARACTER OBJECT

Save game function saves the entire character to file.

First half of the function loads saves.yml and iterates through save files to check for a matching name.

If a matching name is found, file\_found flag is set to true.

```
def save_game
  file_found = ""
  overwrite = ""
  save = $player.to_yaml
  File.open("./saves.yml") do |file_iter|
    YAML.load_stream(file_iter) do |line|
      if line.to_s == $player.to_s
        file_found = true
        $slot = line
        puts "Do you want to save over #{line.file_show}"
        overwrite = yesno
        system("clear")
      end
    end
  end
end
```

In addition, the function prompts the player if they want to overwrite the previous save.

## CHARACTER.RB / CHARACTER OBJECT

Second half saves to file.

If there is a previous file and overwrite flag is yes, we load the whole save file and use gsub to swap in our new save.

If the overwrite flag is false, we prompt the user to provide a new name.

```
if file_found == true && overwrite == "Yes"
  temp_file = File.read("./saves.yml").gsub($slot.to_yaml, save)
  File.write("./saves.yml", temp_file, mode: "w")
  puts "#{@name} was zaved to da phial."
elsif file_found == true && overwrite == "No"
  print "Please choose a new name: "
  named = gets.chomp
  @name = named
  save_game
else file_found == false
  File.write("./saves.yml", save, mode: "a")
  puts "#{@name} was zaved to da phial."
end
end
```

If there was no file found, we just amend our save to the end of the file.

## SAVE TO FILE

Gem YAML is used to convert Character objects into easily stored and retrieved data.

Later on, YAML can read the file object by object without having to split up the file into readable bits manually.

```
--- !ruby/object:Character
name: Grog
str: 2
dex: 1
int: 5
wrđ: 4
hp: 10
armor: 11
ward: 11
damage: rand(1..8)
atk: dex
level: 1
exploration: 0
money: 20
```



## CHARACTER.RB / CHARACTER OBJECT

Attack function is in the works

Parameters will determine the attacker and the defender

First step is determine what stat the attacker uses, dex or str.

After that, a random number between 1-20 is chosen and the relevant stat is added.

```
def attack(atakr,dfndr)
  if atkr.atk == "dex"
    attack_stat = atkr.dex
  elsif atkr.atk == "str"
    attack_stat = atkr.str
  end
  if rand(1..20) + attack_stat >= dfndr.armor
    dfndr.hp -= eval(atakr.damage)
  end
end
```

If that number is equal to or beats the defender's armor, the attack hits and the attackers damage value is run using eval, which outputs a numerical value to subtract from the defenders hitpoints.

## CHARACTER.RB / CHARACTER OBJECT

Improve function is yet to be started

Will be called when a character has reached the xp cap, levels the character up and resets xp.

It will prompt the player to choose between 3 of a random list of improvements.

```
def attack(atkr,dfndr)
  if atkr.atk == "dex"
    attack_stat = atkr.dex
  elsif atkr.atk == "str"
    attack_stat = atkr.str
  end
  if rand(1..20) + attack_stat >= dfndr.armor
    dfndr.hp -= eval(atkr.damage)
  end
end
```

Due to stats being individual and damage being eval'd, we can just increase the stats and even change the entire damage calculation as part of an improvement.

```
def load_game(load_file)
  if load_file != ""
    File.open("./saves.yml") do |file_iter|
      YAML.load_stream(file_iter) do |line|
        if line.to_s == load_file
          $player = line
        end
      end
    end
  else
    File.open("./saves.yml") do |file_iter|
```

```
end
    File.open("./saves.yml") do |file_iter|
      YAML.load_stream(file_iter) do |line|
        puts line.file_show
      end
    end
    puts "Oi! Which Orkz iz you?"
    lf = gets.chomp
    if lf.length > 0
      load_game(lf)
      system("clear")
      puts "#{$player.name} loaded."
    else
      system("clear")
      puts "Try againz when you'ze a real Ork."
    end
  end
end
```

## LOADING A CHARACTER

Load\_game function is inside index.rb, and handles loading a character object to the global \$player variable.

If a parameter is included, it will search the save file (using similar method to save) and if a matching file is found it is automatically loaded.

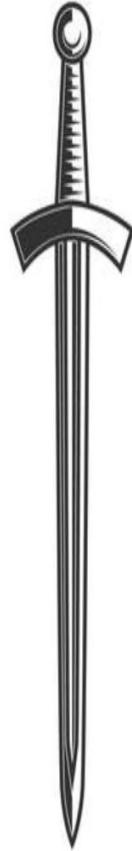
Otherwise, all saves are printed and the user is prompted to input the name of their Ork.

# BATTLE

The battle function is currently unwritten due to changes in scope of the application during development.

The plan for development is to finish the `Character.attack` and `Character.improve` function, and use those as the basis for the battle feature.

It will allow the currently loaded character face off against other save files to earn xp.



Battle feature should work as follows:

Function is called with 2 parameters (attacker, defender)

Roll  $1d6 + 2$

If the roll  $> 4$ , player makes attack before loop is initiated.

While player AND enemy health  $\neq 0$   
Enemy attacks player,  
Checks for victory  
Player attacks enemy  
End

At the end of the battle,  
Check player for level up  
If so, run `Improve()`

# ARGUMENTS

I decided to use arguments as tools to run different functions inside the app quickly, for both testing purposes and the ability to quickly load into the game.

The first argument determines how the second argument is used:

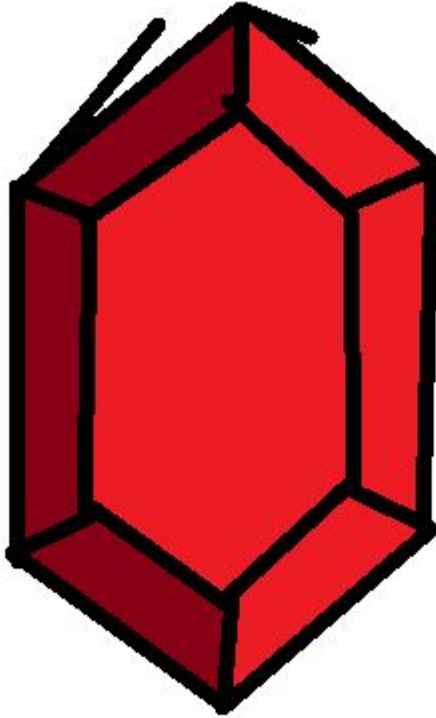
“Load” – Loads character provided in parameter 2, otherwise shows the load menu.

“New” – Opens character creation with parameter 2 as the name.

“Quick” – Quickly generates a character with parameter 2 as the name.

```
cmd1 = ARGV[0]
cmd2 = ARGV[1]
ARGV.clear
if cmd1 != nil
  if cmd1 == "load"
    if cmd2 != nil
      load_game(cmd2)
    else
      load_game("")
    end
  elsif cmd1 == "new"
    if cmd2 != nil
      create_character(-1, cmd2)
    else
      create_character(-1, "")
    end
  elsif cmd1 == "quick"
    if cmd2 != nil
      create_character(0, cmd2)
    else
      create_character(0, "")
    end
  else
    puts "Woopzie!"
  end
end
```

# GEMS



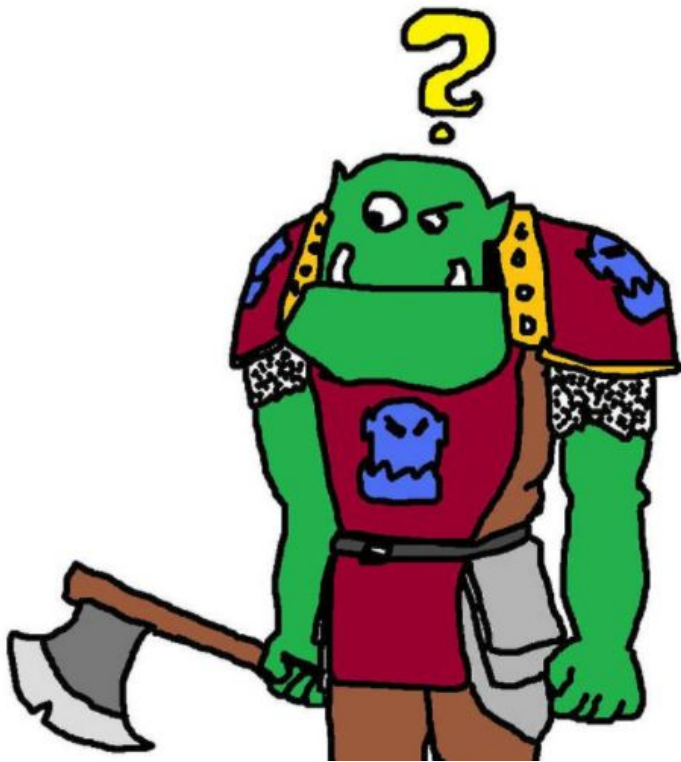
TTY-Prompt - Used for menus

Faker - Used for automatically generating names

YAML - Used to convert objects to easily read/written data for saving to file

Colorize - Not used yet, but will be implemented for decoration and readability in battles.

Possibly more to be used for styling the application. (Terminal-table, maybe ascii)



“Mr.Orc the Confused”

By user Tingest on DeviantArt

# ETHICAL ISSUES

Little to no ethical  
issues:

---

Faker generates names from  
Elder Scrolls, but they are  
pretty much generic fantasy  
names.

Source Material for game  
(Arcane Ugly) mechanics are  
open source.

# CHALLENGES

Challenges for this  
assessment include:

Time management

Figuring out how to use  
YAML to handle save file

Scope of the application

