

TERMINAL APP: 'GAME OF DEATH'



BY CONRAD CLOSE

1

WHAT IS 'GAME OF DEATH'?

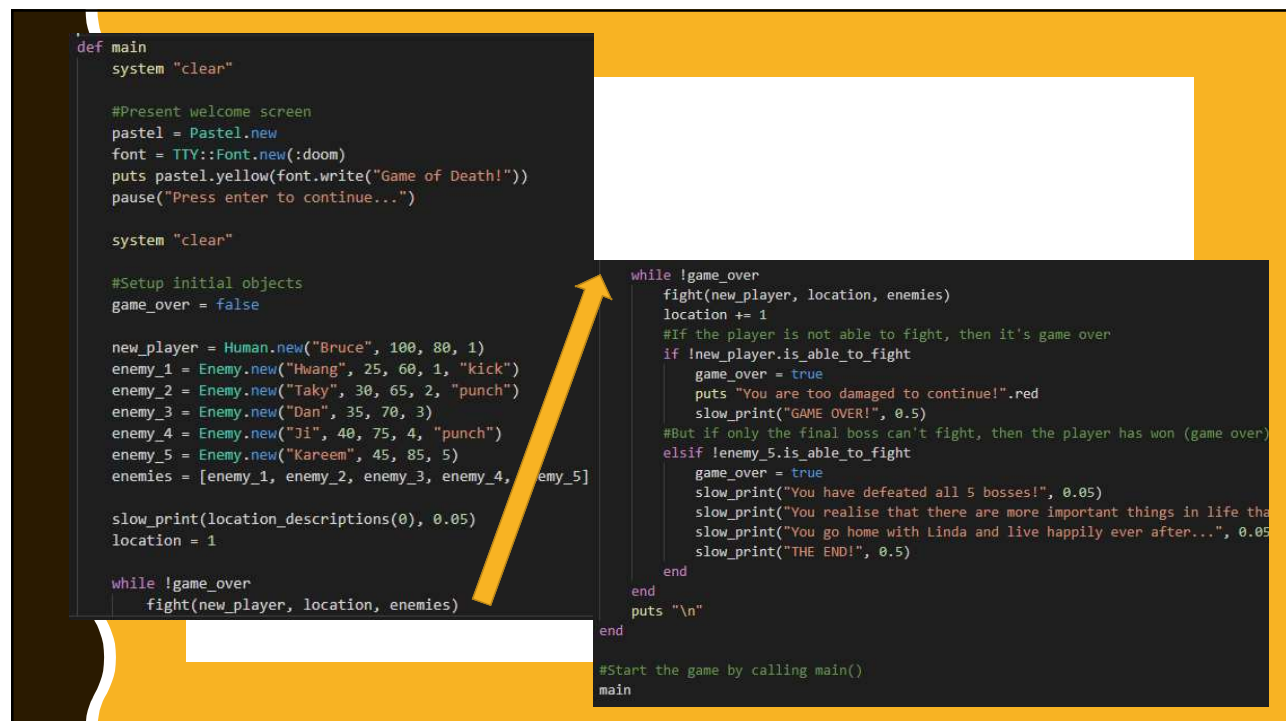
- A colourful, interactive and light-hearted text-based adventure based on the unreleased version of the movie 'Game of Death', written by Bruce Lee. Also inspired by games like Dwarf Fortress and verbose games like the original Fallout's and the old Sierra titles.
- It's a game which takes a single player on a brief adventure into enemy territory to save their girlfriend from 5 evil bosses.
- It allows the player to make direct choices which have a direct impact on the other characters' physicality and ability to fight. 2 crippled arms means a person can't punch, for instance. Damaging a person's head means that person becomes less accurate...
- Challenges? Deciding where to use class-methods or just methods, deciding how to structure the flow of the game in general...
- Ethical issues?

2

HOW IS IT STRUCTURED?

- Core functionality is split up into 4 files:
 - classes.rb – handles the player and opponent classes with their various attributes
 - methods.rb – stores the primary methods for executing gameplay and handling user input
 - data.rb – stores data and arrays which hold large amounts of data (e.g. verbose taunts and location descriptions)
 - tests.rb – test cases for testing the class methods
 - game_of_death.rb – stores the main function and player/enemy class initializations

3



4

```

class Human
  attr_accessor :name, :health, :accuracy, :location, :limbs, :is_able_to_fight

  def initialize(name, health, accuracy, location=false, prefers_attack=false)
    @name = name
    @health = health
    @accuracy = accuracy #Used for calculating chance to hit an opponent
    @location = location #The physical location they are in (rooms 1-5)
    @is_able_to_fight = true #changes to false if certain limbs get too damaged

    #Holds the person's limbs with a verbose descriptor and health
    @limbs = {
      "la" => ["left arm", 25],
      "ra" => ["right arm", 25],
      "ll" => ["left leg", 30],
      "rl" => ["right leg", 30],
      "h" => ["head", 25],
      "g" => ["groin", 15],
      "t" => ["torso", 50]
    }
  end

  #Returns a random limb, used for generating a weak initial limb or a random target limb
  def get_random_limb
    random_limb = @limbs.values.sample #Get a random sample from the limbs hash
    return random_limb
  end
end

```

5

```

#Enemies can (and do) have a weakness and sometimes have a preferred attack
class Enemy < Human
  attr_reader :weakness, :prefers_attack

  def initialize(name, health, accuracy, location=false, prefers_attack=false)
    super
    @prefers_attack = prefers_attack
    @weakness = self.get_random_limb #Set a random limb to be a weak limb
  end

  def talk(struck_limb) #When player hits the enemy, talk() is called
    return "#{@name}: #{speak(@location-1, struck_limb, @weakness[0])}"
  end
end

```

6

```
def location_descriptions(index)
  descriptions = [
    "Your girlfriend, Linda, has been kidnapped! The local mafia are keeping her in the top level of the mysterious pagoda.",
    "You step inside, the room is a dojo of some sort, there is a man meditating. He hears you enter and approaches you.",
    "You catch your breath and head upstairs. It's a dark room with a single chair, where another angry man is sitting.",
    "Exhausted, you continue upstairs. Into an even darker room with an even angrier, louder man, get ready!",
    "You crawl upstairs. There is no one here and you are safe. Just kidding, you are being attacked!",
    "You have reached the top floor! Your girlfriend is playing Monopoly with Kareem, the mafia boss. He attacks you!"
  ]
  return descriptions[index]
end

#Enemy taunts, said when you miss
def taunts
  taunts = [
    "You fight as well as you smell! Which is to say, bad!",
    "You suck!",
    "No pain, no gain!",
    "Linda is ours now! Get over it!",
    "You are a bad person!",
    "Get out! This is our pagoda!",
    "You're from that place I don't like!",
    "I trained for years just to kick your butt!",
    "If you were a better boyfriend, you wouldn't be in this situation!",
    "You look like a weakling! Oh! It's because you are!",
    "Hahahah you missed!",
    "Missed me again!",
  ]
end
```

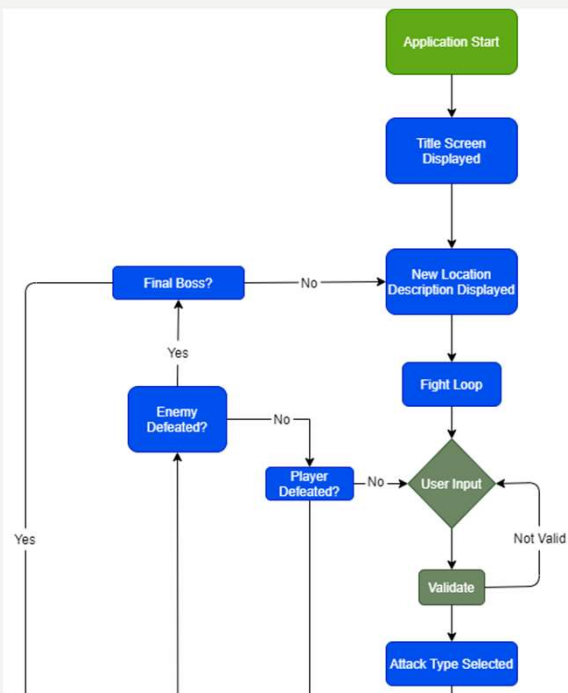
7

```
def speak(index, struck_limb, weak_limb)
  dialogues = [
    ["OW!", "Ow! My #{struck_limb}!", "I love my #{weak_limb}...", "Don't hit my weak #{weak_limb}!!", "Nooooo! My #{weak_limb}!"],
    ["Oof!", "I'm removing you from Facebook...", "*Strokes his #{weak_limb}", "Why'd you have to hit my #{struck_limb}?", "I'm a bleeder"],
    ["Oh my God that smarts!", "Owie owie owie!!", "If I put on my glasses would you stop hitting me?", "I'm a bleeder"],
    ["*cries*", "What did my #{struck_limb} ever do to you?", "My God you're strong! I see why your girlfriend likes you"],
    ["OW! I should've become a programmer...", "I won't let my family down!", "I was about to pass GO, too!", "I feel better"]
  ]
  return dialogues[index][rand(0..dialogues[index].length-1)]
end
```

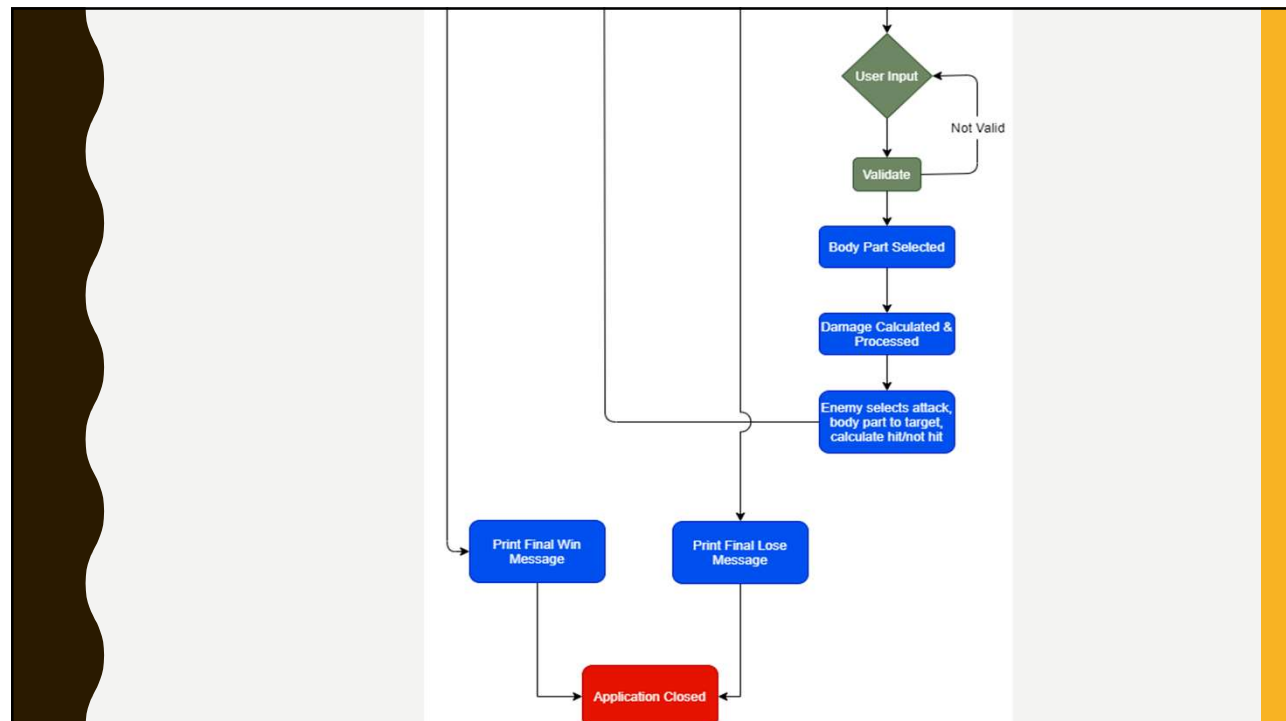
My favourite part of the code.
 The random speech generator for the 5 enemies.
 Sometimes they reveal their “weak_limb” and sometimes
 they refer to the limb you just struck.
 Adds some fun to the game.

8

CODE OVERVIEW



9



10

