Snake in Neural Nets

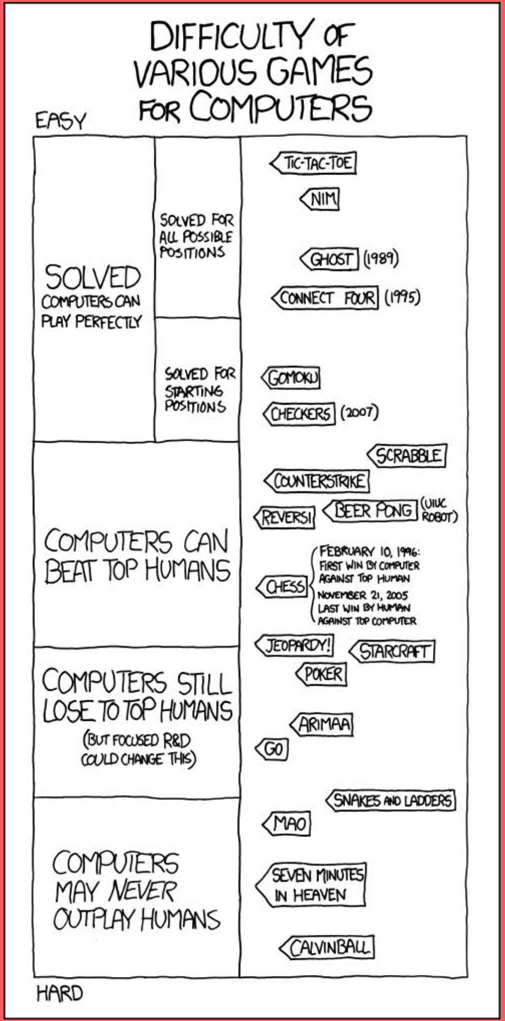
Pearson Mewbourne

https://www.youtube.com/watch?v=q6s99eH9EC8&list=FLJsOcGrlnD2Uv4PlFTV4dlA

Where it all began

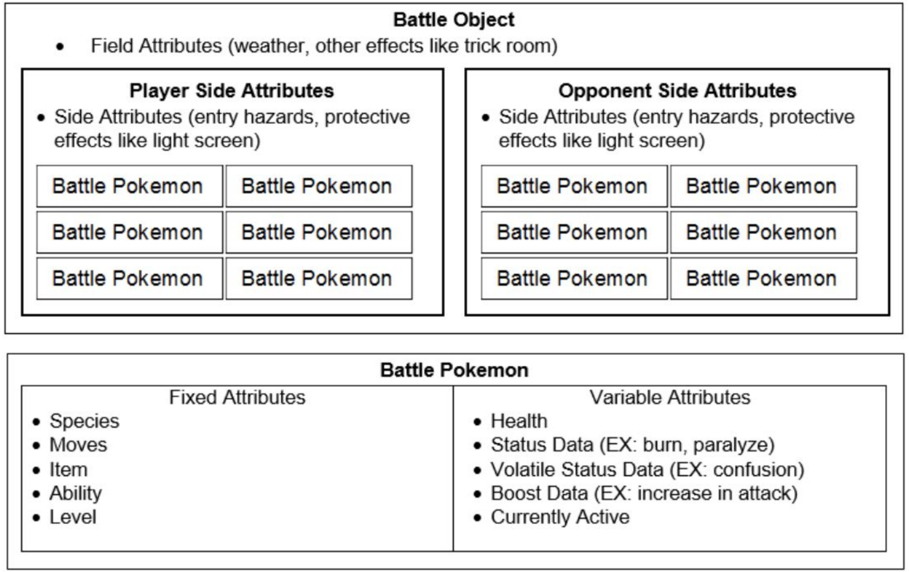
- I was inspired by hw3
    - Inside, there's the file pokemon.csv
- It really got me thinking
- Data driven?
- Creates new pokemon?
- But then I was inspired
- NEURAL NETS!

It was this image that changed my mind


DIFFICULTY OF VARIOUS GAMES FOR COMPUTERS
EASY
SOLVED
COMPUTERS CAN PLAY PERFECTLY
SOLVED FOR ALL POSSIBLE POSITIONS
TIC-TAC-TOE
NIM
GHOST (1989)
CONNECT FOUR (1995)
SOLVED FOR STARTING POSITIONS
GOMOKU
CHECKERS (2007)
COMPUTERS CAN BEAT TOP HUMANS
SCRABBLE
COUNTERSTRIKE
REVERSI
BEER PONG (UIUC ROBOT)
CHESS
FEBRUARY 10, 1996: FIRST WIN BY COMPUTER AGAINST TOP HUMAN
NOVEMBER 21, 2005 LAST WIN BY HUMAN AGAINST TOP COMPUTER
JEOPARDY!
STARCRAFT
POKER
COMPUTERS STILL LOSE TO TOP HUMANS
(BUT FOCUSED R&D COULD CHANGE THIS)
ARIMAA
GO
SNAKES AND LADDERS
MAO
COMPUTERS MAY NEVER OUTPLAY HUMANS
SEVEN MINUTES IN HEAVEN
CALVINBALL
HARD

And this paper

https://varunramesh.net/content/documents/cs221-final-report.pdf



Battle Object

- Field Attributes (weather, other effects like trick room)

Player Side Attributes

- Side Attributes (entry hazards, protective effects like light screen)

Battle Pokemon	Battle Pokemon
Battle Pokemon	Battle Pokemon
Battle Pokemon	Battle Pokemon

Opponent Side Attributes

- Side Attributes (entry hazards, protective effects like light screen)

Battle Pokemon	Battle Pokemon
Battle Pokemon	Battle Pokemon
Battle Pokemon	Battle Pokemon

Battle Pokemon

Fixed Attributes

- Species
- Moves
- Item
- Ability
- Level

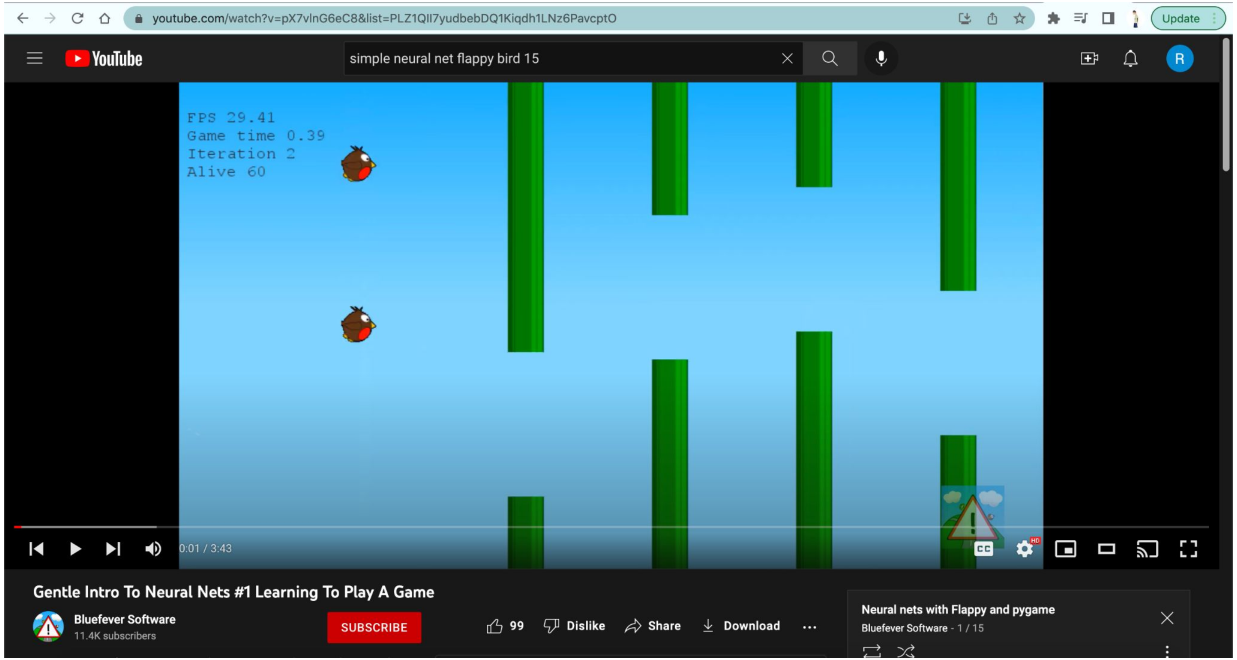
Variable Attributes

- Health
- Status Data (EX: burn, paralyze)
- Volatile Status Data (EX: confusion)
- Boost Data (EX: increase in attack)
- Currently Active

Too much data!

So... let’s scale it back

I wanted to learn how to make a neural net play a video game… so google lead me to this simple series



youtube.com/watch?v=pX7vlnG6eC8&list=PLZ1QIl7yudbebDQ1Kiqdh1LNz6PavcptO

simple neural net flappy bird 15

```
FPS 29.41
Game time 0.39
Iteration 2
Alive 60
```

0:01 / 3:43

Gentle Intro To Neural Nets #1 Learning To Play A Game

Bluefever Software
11.4K subscribers

Neural nets with Flappy and pygame
Bluefever Software - 1 / 15

Libraries

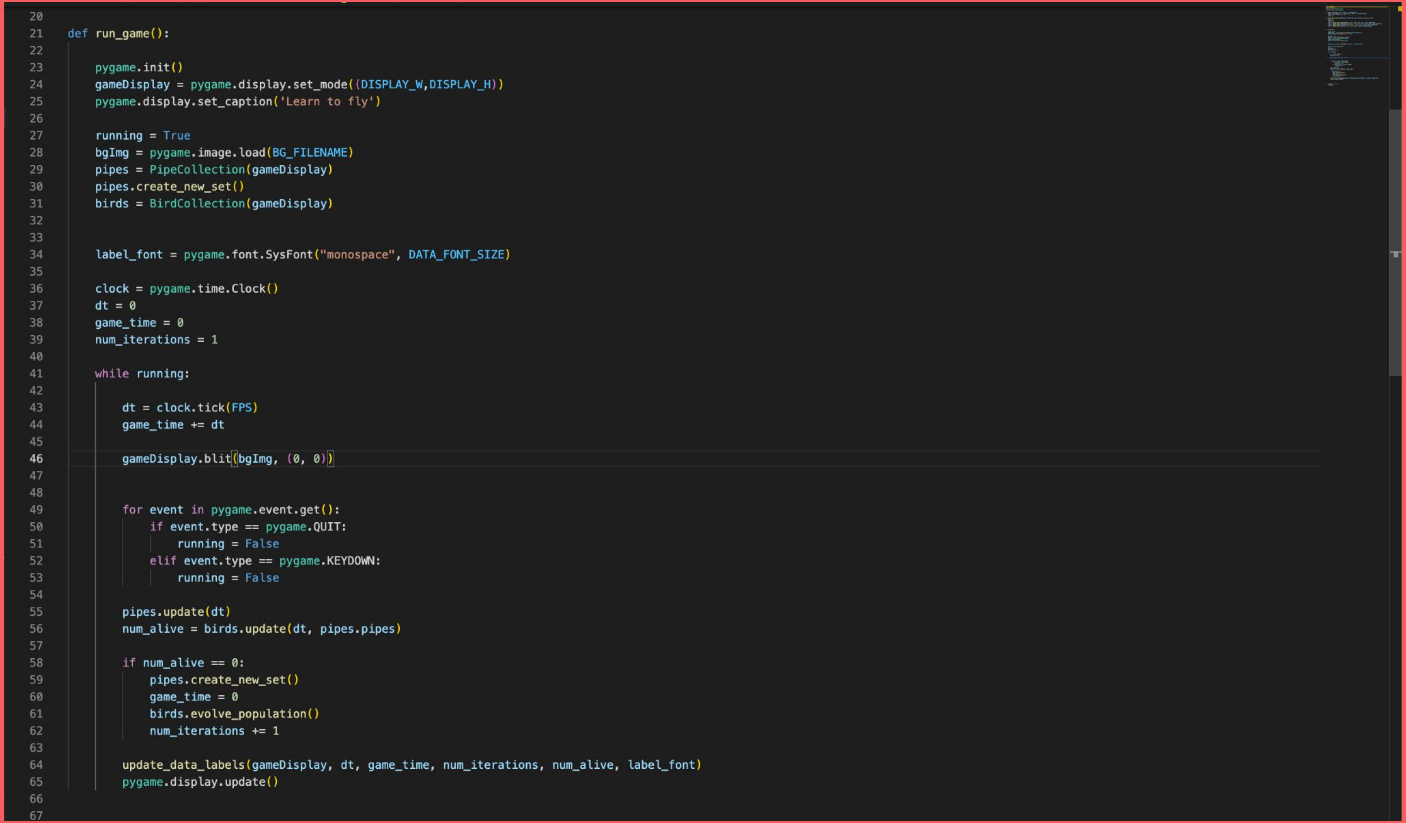
- Pygame - making the games
- NumPy
- SciPy

So I read through his code, commenting for my understanding, and I was able to make this!

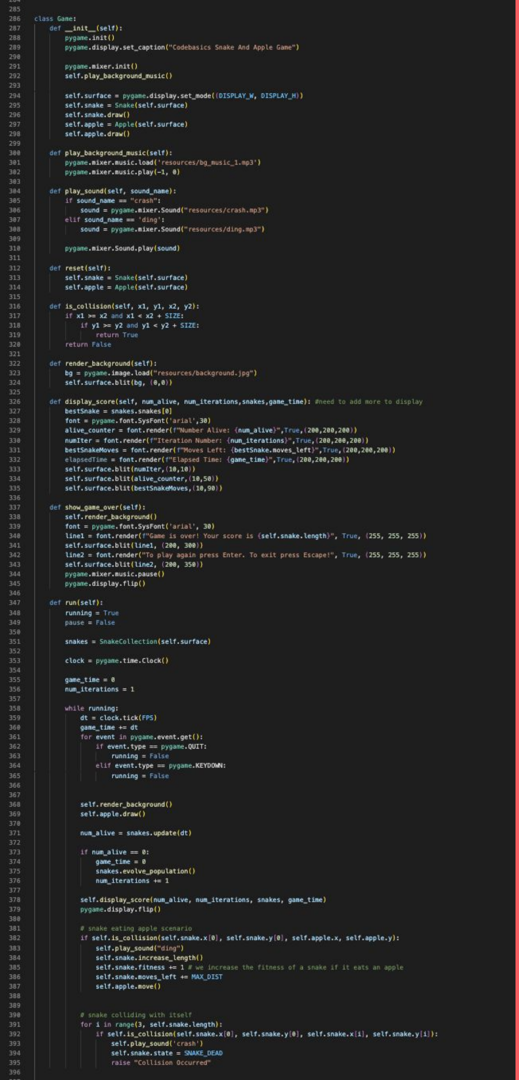
But that’s not good enough!

- I want it to learn another game
- And I thought snake would work
  - Simple data
  - Simple actions
  - Simple goals
- So I found a version of snake made in pygame and tried to jam the two codes together
- Which lead to this…

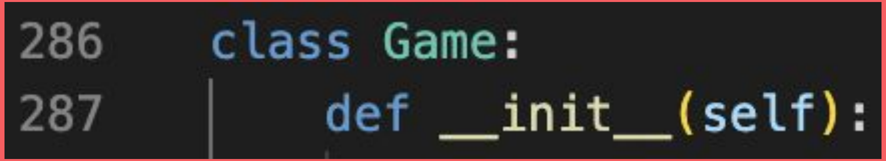
Stylistic differences



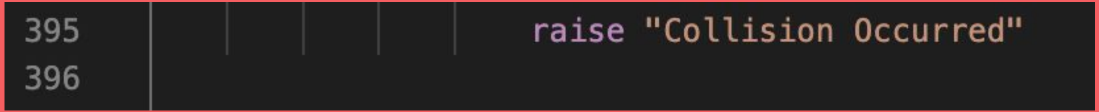
```python

def run_game():

    pygame.init()
    gameDisplay = pygame.display.set_mode((DISPLAY_W,DISPLAY_H))
    pygame.display.set_caption('Learn to fly')

    running = True
    bgImg = pygame.image.load(BG_FILENAME)
    pipes = PipeCollection(gameDisplay)
    pipes.create_new_set()
    birds = BirdCollection(gameDisplay)


    label_font = pygame.font.SysFont("monospace", DATA_FONT_SIZE)

    clock = pygame.time.Clock()
    dt = 0
    game_time = 0
    num_iterations = 1

    while running:

        dt = clock.tick(FPS)
        game_time += dt

        gameDisplay.blit(bgImg, (0, 0))


        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                running = False
            elif event.type == pygame.KEYDOWN:
                running = False

        pipes.update(dt)
        num_alive = birds.update(dt, pipes.pipes)

        if num_alive == 0:
            pipes.create_new_set()
            game_time = 0
            birds.evolve_population()
            num_iterations += 1

        update_data_labels(gameDisplay, dt, game_time, num_iterations, num_alive, label_font)
        pygame.display.update()


```



```python

class Game:
    def __init__(self):
        pygame.init()
        pygame.display.set_caption("Codebasics Snake And Apple Game")

        pygame.mixer.init()
        self.play_background_music()

        self.surface = pygame.display.set_mode((DISPLAY_W, DISPLAY_H))
        self.snake = Snake(self.surface)
        self.snake.draw()
        self.apple = Apple(self.surface)
        self.apple.draw()

    def play_background_music(self):
        pygame.mixer.music.load('resources/bg_music_1.mp3')
        pygame.mixer.music.play(-1, 0)

    def play_sound(self, sound_name):
        if sound_name == "crash":
            sound = pygame.mixer.Sound("resources/crash.mp3")
        elif sound_name == 'ding':
            sound = pygame.mixer.Sound("resources/ding.mp3")

        pygame.mixer.Sound.play(sound)

    def reset(self):
        self.snake = Snake(self.surface)
        self.apple = Apple(self.surface)

    def is_collision(self, x1, y1, x2, y2):
        if x1 >= x2 and x1 < x2 + SIZE:
            if y1 >= y2 and y1 < y2 + SIZE:
                return True
        return False

    def render_background(self):
        bg = pygame.image.load("resources/background.jpg")
        self.surface.blit(bg, (0,0))

    def display_score(self, num_alive, num_iterations,snakes,game_time): #need to add more to display
        bestSnake = snakes.snakes[0]
        font = pygame.font.SysFont('arial',30)
        alive_counter = font.render(f"Number Alive: {num_alive}",True,(200,200,200))
        numIter = font.render(f"Iteration Number: {num_iterations}",True,(200,200,200))
        bestSnakeMoves = font.render(f"Moves Left: {bestSnake.moves_left}",True,(200,200,200))
        elapsedTime = font.render(f"Elapsed Time: {game_time}",True,(200,200,200))
        self.surface.blit(numIter,(10,10))
        self.surface.blit(alive_counter,(10,50))
        self.surface.blit(bestSnakeMoves,(10,90))

    def show_game_over(self):
        self.render_background()
        font = pygame.font.SysFont('arial', 30)
        line1 = font.render(f"Game is over! Your score is {self.snake.length}", True, (255, 255, 255))
        self.surface.blit(line1, (200, 300))
        line2 = font.render("To play again press Enter. To exit press Escape!", True, (255, 255, 255))
        self.surface.blit(line2, (200, 350))
        pygame.mixer.music.pause()
        pygame.display.flip()

    def run(self):
        running = True
        pause = False

        snakes = SnakeCollection(self.surface)

        clock = pygame.time.Clock()

        game_time = 0
        num_iterations = 1

        while running:
            dt = clock.tick(FPS)
            game_time += dt
            for event in pygame.event.get():
                if event.type == pygame.QUIT:
                    running = False
                elif event.type == pygame.KEYDOWN:
                    running = False


            self.render_background()
            self.apple.draw()

            num_alive = snakes.update(dt)

            if num_alive == 0:
                game_time = 0
                snakes.evolve_population()
                num_iterations += 1

            self.display_score(num_alive, num_iterations, snakes, game_time)
            pygame.display.flip()

            # snake eating apple scenario
            if self.is_collision(self.snake.x[0], self.snake.y[0], self.apple.x, self.apple.y):
                self.play_sound("ding")
                self.snake.increase_length()
                self.snake.fitness += 1 # we increase the fitness of a snake if it eats an apple
                self.snake.moves_left += MAX_DIST
                self.apple.move()


            # snake colliding with itself
            for i in range(3, self.snake.length):
                if self.is_collision(self.snake.x[0], self.snake.y[0], self.snake.x[i], self.snake.y[i]):
                    self.play_sound('crash')
                    self.snake.state = SNAKE_DEAD
                    raise "Collision Occurred"

```



```python
class Game:
    def __init__(self):
```



```python
                    raise "Collision Occurred"

```

Not very succinct

What next?

Well obviously, I need to fix the snake code itself.

But I also want to return to my original idea!

I’ve already found APIs:

The API to scrape Pokemon Showdown

The API to scrape Smogon, a website with Pokemon stats and data

More things I need for that

A better understanding of PyGame or another way to represent the Game

Knowledge of how to make a neural net that can interact with websites

A neural net that can play against itself to improve

Dedication- this will take time

Thank You!

Sources

Percymon: A Pokemon Showdown Artifical Intelligence

Neural Nets with Flappy and PyGame

Snake in PyGame