Proyecto 2

Pokeminmax!

Sofia Castillo Giraldo
Merly Velásquez Cortez

Faculta de Ingeniera, Universidad del Valle

Inteligencia Artificial

Joshua David Triana

12 de mayo del 2025

1. **Definición de Estructuras**
   Se creo una clase para el Pokémon y para el movimiento donde se definió los atributos de cada una, para posteriormente representar el combate entre los pokemones.

```python
class Movimiento:
    def __init__(self, nombre, tipo, poder):
        self.nombre = nombre
        self.tipo = tipo
        self.poder = poder


class Pokemon:
    def __init__(self, nombre, tipo, ps, movimientos):
        self.nombre = nombre
        self.tipo = tipo
        self.ps = ps
        self.movimientos = movimientos
```

2. Definiciòn de Funciones

2.1 Funciòn esta_vivo

Es una función que verifica en cada movimiento hecho si el jugador o el defensor tienen PS > 0.

```python
def esta_vivo(self):
    return self.ps > 0
```

2.2 Funcion aplicar_daño

En esta función se obtendrá el cálculo del daño, para así restar puntos de salud (ps) en cada ataque al contrincante y debilitarlo mientras está vivo. Además, el daño se reflejará en la interfaz mediante un cambio de color en la barra.

```python
def aplicar_daño(atacante, defensor, movimiento):
    daño = calcular_daño(movimiento.tipo, defensor.tipo, movimiento.poder)
    defensor.ps -= daño

    if not defensor.esta_vivo():
        print(f"{defensor.nombre} ha sido debilitado!")
```

2.3 Funcion calcular_daño

En esta función se calcula el daño considerando el tipo de movimiento (ataque) y el tipo de pokemón. El cálculo se hace con el poder (puntaje) del movimiento multiplicado por la efectividad (super efectividad: 2.0, efectiva:0.5, no efectiva: 0). Cuando no existe ninguna efectividad, devuelve uno.

```python
def calcular_daño(tipo_ataque, tipo_defensa, poder):

    efectividad = EFECTIVIDAD.get(tipo_ataque.lower(), {}).get(tipo_defensa.lower(), 1)

    return poder * efectividad
```

```
EFECTIVIDAD = {
    'agua':     {'agua': 0.5, 'planta': 0.5, 'fuego': 2.0, 'dragon': 0.5, 'roca': 2.0, 'Tierra': 0.5},
    'fuego':    {'bicho': 2.0, 'planta': 2.0, 'agua': 0.5, 'fuego': 0.5, 'hielo':2.0, 'acero': 2.0, 'roca': 0.5, 'dragón': 0.5},
    'planta':   {'agua': 2.0, 'tierra': 2.0, 'roca': 2.0, 'fuego': 0.5, 'planta': 0.5, 'veneno': 0.5, 'volador': 0.5, 'bicho': 0.5, 'dr
    'electrico':{'agua': 2.0, 'electrico': 0.5, 'tierra': 0.0, 'planta': 0.5, 'volador': 2.0, 'dragón': 0.5},
    'normal':   {'roca': 0.5, 'acero': 0.5, 'fantasma': 0.0},
    'bicho':    {'planta': 2.0, 'psíquico': 2.0, 'siniestro': 2.0, 'fuego': 0.5, 'volador': 0.5, 'lucha': 0.5, 'veneno': 0.5, 'hada': 0
    'lucha':    {'normal': 2.0, 'hielo': 2.0, 'roca': 2.0, 'siniestro': 2.0, 'acero': 2.0, 'veneno': 0.5, 'volador': 0.5, 'psíquico': 0
    'volador':  {'eléctrico': 0.5, 'roca': 0.5, 'planta': 2.0, 'lucha': 2.0, 'bicho': 2.0, 'acero': 0.5},
    'veneno':   {'planta': 2.0, 'hada': 2.0, 'veneno': 0.5, 'tierra': 0.5, 'roca': 0.5, 'fantasma': 0.5, 'acero': 0.0},
    'tierra':   {'fuego': 2.0, 'eléctrico': 2.0, 'veneno': 2.0, 'roca': 2.0, 'acero': 2.0, 'planta': 0.5, 'bicho': 0.5, 'volador': 0.0}
    'fantasma': {'normal': 0.0, 'psíquico': 2.0, 'fantasma': 2.0, 'siniestro': 0.5},
    'siniestro':{'psíquico': 2.0, 'siniestro': 0.5, 'lucha': 0.5, 'hada': 0.5, 'fantasma': 2.0},
    'psíquico': {'lucha': 2.0, 'veneno': 2.0, 'siniestro': 0.0, 'psíquico': 0.5, 'acero': 0.5},
    'dragón':   {'dragón': 2.0, 'acero': 0.5, 'hada': 0.0},
    'hielo':    {'dragón': 2.0, 'planta': 2.0, 'agua': 0.5, 'fuego': 0.5, 'volador': 2.0, 'tierra': 2.0, 'hielo': 0.5, 'acero': 0.5},
    'acero':    {'roca': 2.0, 'hielo': 2.0, 'volador': 2.0, 'agua': 0.5, 'planta': 0.5, 'fuego': 0.5, 'acero': 0.5, 'lucha': 0.5, 'psíq
    'roca':     {'fuego': 2.0, 'hielo': 2.0, 'volador': 2.0, 'bicho': 2.0, 'lucha': 0.5, 'tierra': 0.5, 'acero': 0.5},
    'hada':     {'dragón': 2.0, 'lucha': 2.0, 'siniestro': 2.0, 'veneno': 0.5, 'acero': 0.5, 'fuego': 0.5}
}
```

2.4 Funcion minimax_con_poda

En esta función se calcula desde el punto de vista de la IA. Donde evalúa todas las jugadas posibles. Existen dos casos:

a. Turno de la IA: calcula el daño de cada uno de sus ataques y analiza cual es ataque realiza el mayor daño.
b. Turno del jugador: calcula la respuesta que podría tener el jugador a los ataques de la IA y además calcula el daño que cada uno de sus ataques le podría realizar a la IA.

En general, este algoritmo analiza posibles mejores ataques para simular cual llevaria a ganar.

```python
def minimax_con_poda(estado, profundidad, alpha, beta, es_max_turno):
    if profundidad == 0 or not estado['jugador'].esta_vivo() or not estado['ia'].esta_vivo():
        return funcion_evaluacion(estado), None

    mejor_movimiento = None

    if es_max_turno:
        max_eval = float('-inf')
        for movimiento in estado['ia'].movimientos:
            estado_copia = copiar_estado(estado)
            aplicar_daño_simulado(estado_copia['ia'], estado_copia['jugador'], movimiento)
            evaluacion, _ = minimax_con_poda(estado_copia, profundidad - 1, alpha, beta, False)
            if evaluacion > max_eval:
                max_eval = evaluacion
                mejor_movimiento = movimiento
            alpha = max(alpha, evaluacion)
            if beta <= alpha:
                break
        return max_eval, mejor_movimiento
    else:
        min_eval = float('inf')
        for movimiento in estado['jugador'].movimientos:
            estado_copia = copiar_estado(estado)
            aplicar_daño_simulado(estado_copia['jugador'], estado_copia['ia'], movimiento)
            evaluacion, _ = minimax_con_poda(estado_copia, profundidad - 1, alpha, beta, True)
```

2.5 Funciòn evaluacion_estado

Esta funcion evalua el estado de combate donde si retorna un puntaje positivo la IA tendra ventaja sobre el usario y en caso contrario, con un valor negativo la IA esta desventaja.
Obtiene los puntos de salu de cada entrenador y determina cual es el daño potencial que pueden generar a su contricante.
Esto lo hace sumando el daño que hace el tipo de moviento a el tipo de pokemon que es su contricante y luego diviendo por el numero de movimientos.
Por ultimo retorna un valor que indica si la IA esta en ventaja o desventaja.

```python
def funcion_evaluacion(estado):

    ia = estado['ia']
    jugador = estado['jugador']

    vida_ia = ia.ps
    vida_jugador = jugador.ps

    daño_potencial_ia = sum(calcular_daño(m.tipo, jugador.tipo, m.poder) for m in ia.movimientos) / len(ia.movimient
    daño_potencial_jugador = sum(calcular_daño(m.tipo, ia.tipo, m.poder) for m in jugador.movimientos) / len(jugador

    return (vida_ia - vida_jugador) + (daño_potencial_ia - daño_potencial_jugador)
```

2.6 Funciones de simulación

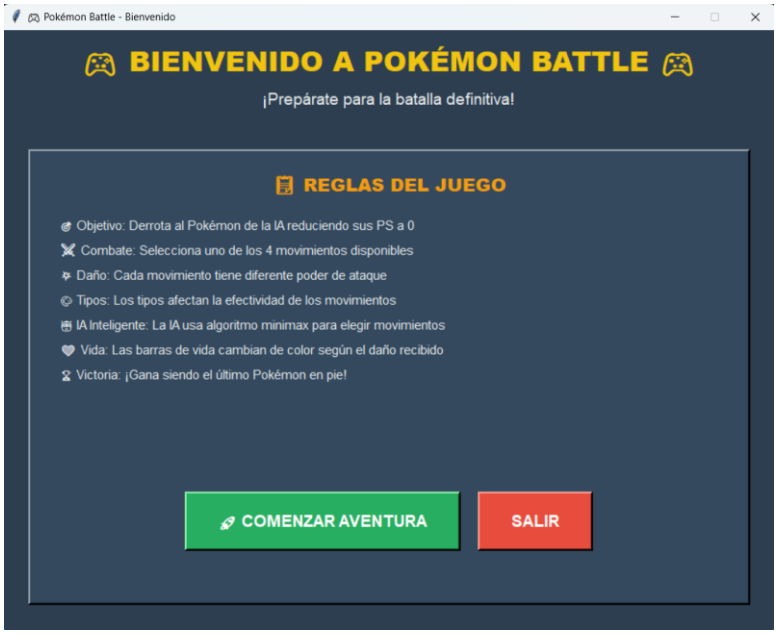
Estas funciones son utilizadas para no interferir en el combate real, simplemente es para evaluar las situaciones y posibilidades.

a. Funcion aplicar daño simulado: de igual manera calcula el daño y lo utiliza en el algorimto de minimax pero no modifica el combate real.
b. Copiar_Pokemon: se genera una copia del pokemon con el estado que le pasa la función copiar_estado.
c. Copiar_estado: esta función genera una copia del estado para pasársela al minimax y no alterar el combate real.

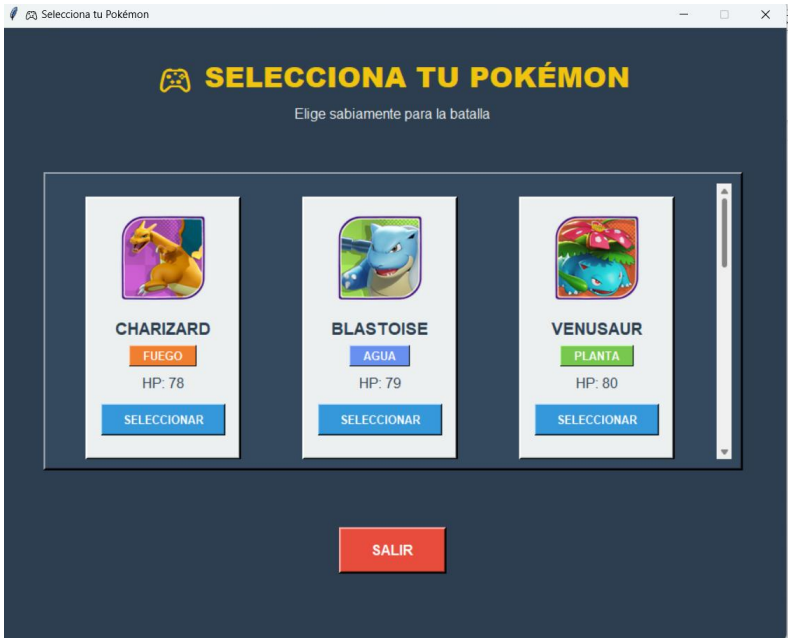
```python
def copiar_pokemon(pkm):
    return Pokemon(
        pkm.nombre,
        pkm.tipo,
        pkm.ps,
        [Movimiento(mov.nombre, mov.tipo, mov.poder) for mov in pkm.movimientos]
    )


def copiar_estado(estado):
    return {
        'jugador': copiar_pokemon(estado['jugador']),
        'ia': copiar_pokemon(estado['ia'])
    }
```

3. Interfaz de usuario

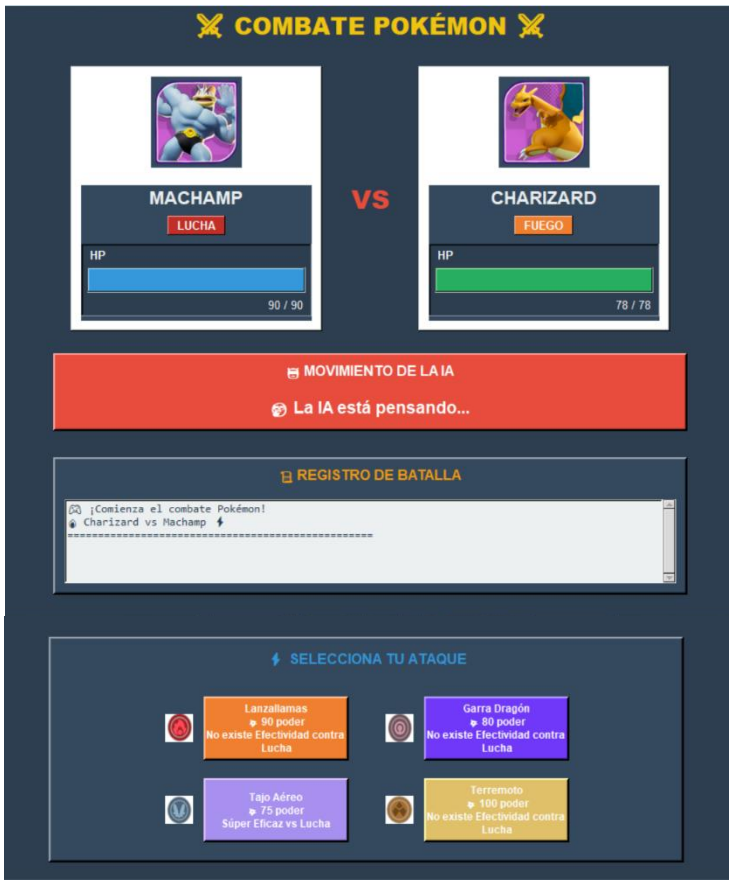
3.1 Panel 1 (principal-welcome)



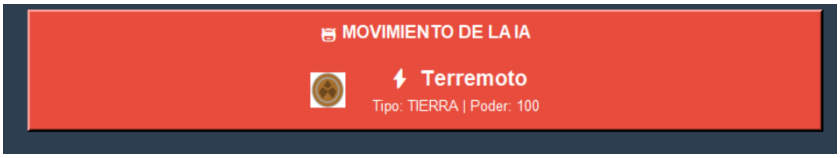
3.2 Panel 2 (selección de pokemón)



3.3 Panel 3 (campo de batalla)



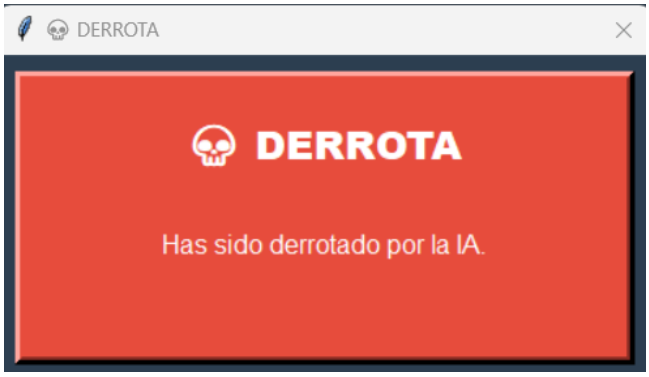
1. Se muestra el Pokémon elegido por la IA y el elegido por el usuario, con cada una de sus características y su respectiva barra de HP.
2. Un recuadro rojo donde se mostrará el ataque que realizo la IA.

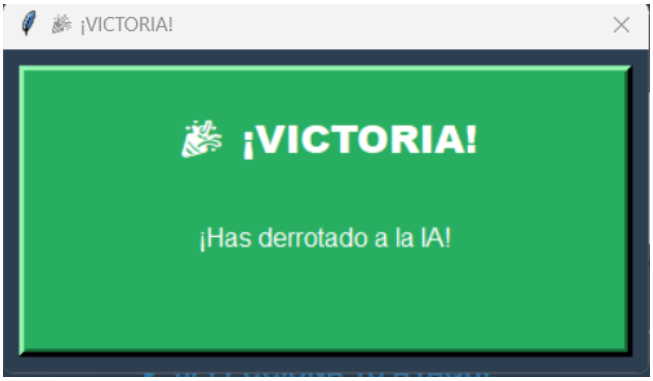


3. Un registro de batalla, que mostrara el daño infligido a su contrincante en cada ataque.

4. Los ataques disponibles del jugador para derrotar a la IA, donde indicará los niveles de efectividad de acuerdo con el tipo de Pokémon de la IA.

3.4 ventanas de derrota y victoria





4. Repositorio GitHub

https://github.com/Mvc2004/Proyecto2IA.git