



## Battle Arena

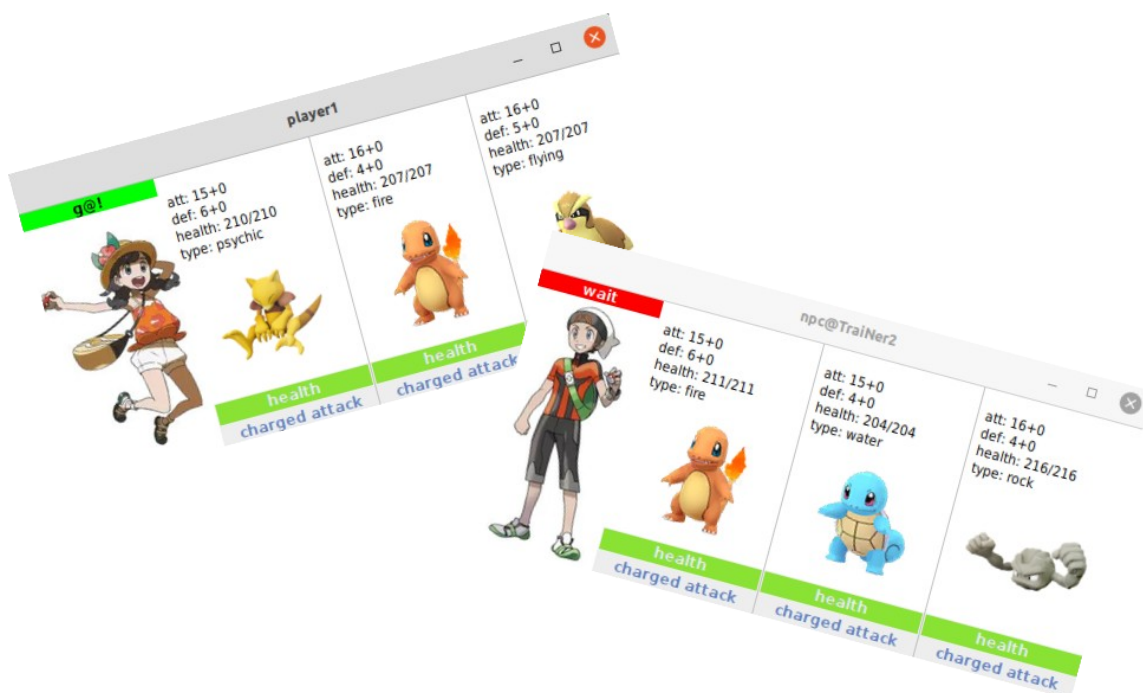
En este proyecto se plantea implementar una variante de los juegos Pokémon. Su dinámica básica es similar a los juegos habituales y se describe brevemente a continuación, las interfaces que se muestran a continuación son orientativas:

1. En este juego se efectuaran batallas de  $n$  jugadores y  $m$  bots (npc), con  $p$  Pokémons cada uno.  $n$ ,  $m$  y  $p$  se definirán mediante la pantalla de inicio:



1. Pantalla de inicio.

2. Una vez se hayan elegido  $n$ ,  $m$  y  $p$  se iniciara la batalla. En este caso se generaran las correspondientes  $n+m$  ventanas (una por cada jugador, dos en el ejemplo):



2. Ventanas de batalla.

3. Para cada Pokémon se definirán los atributos de ataque, defensa, vida y tipo. Estos atributos se inicializarán de la siguiente manera:
  - ataque:  $11 + \text{un valor aleatorio entre } 1-7$ .
  - defensa:  $3 + \text{un valor aleatorio entre } 1-4$ .
  - vida:  $200 + \text{un valor aleatorio entre } 1-20$ .
  - tipo: aleatorio entre **Fuego**, **Agua**, **Planta** y **Eléctrico**.
    - Nota: Se pueden incluir más tipos de Pokémon.
4. La batalla comenzara automáticamente, y se asignaran los turnos para combatir. Los turnos serán aleatorios, es decir un jugador puede repetir turno.
5. En cada turno, el jugador elegirá para cada uno de sus Pokémons a que Pokémon oponente atacará (a cualquier Pokémon del resto de  $n+m-1$  jugadores). Además, se podrá atacar al mismo Pokémon de un oponente más de una vez por turno (con Pokémons diferentes). Cada Pokémon solo puede atacar una vez por turno.
  - La vida del Pokémon que ha sido atacado se reducirá en  **$\text{vida} = \text{vida} - (\text{ataque} * \text{multiplicador}) - \text{defensa}$** . La vida nunca podrá incrementarse.
    - El multiplicador sera 1 por defecto, pero si:
      - El atacante es tipo **Fuego** y el atado es tipo **Planta** el multiplicador será 2.
      - El atacante es tipo **Agua** y el atado es tipo **Fuego** el multiplicador será 2.
      - El atacante es tipo **Eléctrico** y el atado es tipo **Agua** el multiplicador será 2.
      - El atacante es tipo **Planta** y el atado es tipo **Eléctrico** el multiplicador será 2.
      - Nota: Se pueden incluir más tipos de Pokémon.
6. Una vez hayan atacado todos los Pokémon de un jugador, se pasará de turno. Hay que tener en cuenta que puede volver a tocarle.
  - El jugador deberá clicar en un botón para efectuar el cambio de turno (botón g@! en la imagen 2).
  - El bot lo hará automáticamente.
7. **Evoluciones:** durante la batalla los Pokémon evolucionarán, hasta un máximo de dos veces.
  - Primera evolución: los atributos ataque y defensa obtendrán una bonificación permanente de +5 +3 respectivamente. Cuando su vida este por debajo del 50%.
  - Segunda evolución: los atributos ataque y defensa obtendrán una bonificación permanente de +7 +5 respectivamente. Cuando su vida este por debajo del 20%.



3. Ejemplos de evolución.

8. **Estado Euforia:** Cuando un Pokémon haya recibido  $x$  ataques, entrara en un estado de euforia. Su ataque y su defensa se incrementarían temporalmente en +100 hasta que realice su siguiente ataque.  $x$  será un valor aleatorio entre 3 y 7. Una vez haya realizado el ataque, volverá a su estado habitual y podrá volver a este estado especial si recibe otros  $x$  ataques. En los ejemplos la euforia se representa mediante la barra “Charged attack”:



4. Ejemplos de euforia.

## Aspectos importantes

- Este proyecto se realizará en grupo, siguiendo un proceso inspirado en la metodología Scrum, en el que se establecerán tres sprint. Antes de comenzar a realizar el proyecto, los miembros del grupo deben leer detenidamente este enunciado general y establecer los valores que desean asignar a cada propiedad para la que se indica que tiene un valor inicial predeterminado.
- Como valor añadido, en este proyecto hay que implementar la estrategia que utiliza bot para atacar. Se puede hacer de forma aleatoria, pero se valorará positivamente la implementación de estrategias más ‘inteligentes’.
- La aplicación debe aplicar el patrón arquitectónico Modelo-Vista-Controlador. **Si este patrón no se aplica adecuadamente, la calificación del proyecto será 0.**
- El trabajo en grupo es esencial. **La calificación de aquellos proyectos en los que se observe que alguno de los miembros no participe se multiplicará por 0,5.** El equipo es responsable de su gestión y **podrán expulsar del grupo a un miembro si este no participa en el proyecto, notificándoselo al profesor de la asignatura.**
- Junto con este enunciado se entregaran los sprites en .png del juego (Pokémon y Jugadores) y una demo. Las especificaciones del enunciado y la demo pueden ser diferentes, hay que seguir las instrucciones del enunciado.