# Informe ejecutivo

1. de enero de 2024

Bonelli del Hoyo Santiago Nicolás

Guerrero Tomás

Librandi Gonzalo

Rautto Braian

Rodriguez Tomás Agustín

Romero Mateo

El presente informe tiene como objetivo presentar la autonomía de las clases asociadas a los jugadores y a la pelota del diseño del juego. La autonomía se logra mediante la implementación de estrategias, la utilización de un mediador y la herencia de clases. Este informe explora cómo y por qué estas clases son autónomas.

Para lograr la autonomía de las clases, se adopta el concepto de *Sprites* de *Pygame*, que son objetos gráficos representativos de entidades dentro del juego, incluyendo la pelota, los jugadores de campo y el arquero en nuestro escenario. Cada *Sprite* implementa su propio método *update* que les permite gestionar sus estados y comportamientos correspondientes de manera independiente. La activación automática del método *update* en cada fotograma del juego, gracias al sistema de *Pygame*, es posible debido a que las clases *Ball* y *Player* heredan de la clase *pygame.sprite.Sprite*.

Las clases *PlayerField* y *GoalKeeper* heredan de la clase base *Player*, y ambas implementan el método abstracto *update* que contiene la lógica de movimiento del jugador, el pase y el disparo de la pelota. La elección entre una de estas acciones mediante el método *with\_ball* y la obtención de la nueva posición a través del método *where\_to\_pass* en caso de pasar la pelota o el llamado al método *getProxPos* si el jugador desea moverse, se delega a la implementación de la estrategia de cada equipo. Concretamente con respecto al movimiento, se envía un mensaje al método *move* el cual realiza la animación de desplazamiento desde donde se haya el jugador hasta la nueva posición indicada por la estrategia, siempre y cuando el movimiento sea válido, es decir, que el jugador no sobrepase los límites de la cancha. La autonomía se observa en cómo cada clase decide su siguiente posición sin depender directamente de instrucciones externas.

Por otra parte, la autonomía de movimiento presente en la clase *Ball* se rige en función de su interacción con los jugadores. Al colisionar con un jugador, este último se hace de su posesión y puede decidir dirigirla a un compañero, al arco o simplemente moverse con ella como se mencionó anteriormente. En su método *update*, además de chequear los límites de la cancha, ya sea para identificar un gol o posiciones inválidas como salirse fuera de los límites de la misma (realiza un efecto rebote sobre los límites, y si los sobrepasa se efectúa un saque desde el arco más cercano), realiza el llamado al método *animation\_of\_move* que implementa la animación de desplazamiento desde donde se haya la pelota hasta donde le indique el jugador que tiene su posesión. Existe un caso especial que es cuando la pelota colisiona con dos o más jugadores. Para evitar que los mismos peleen por la posesión de la pelota por un tiempo indefinido, lo que ocurre es que el método *fighting\_for\_ball* que se ejecuta en el *update* de la clase *Ball* hace que la pelota salga disparada hacia una dirección aleatoria.

Saludos cordiales.