# Descripción de las funciones implementadas:

Funciones de movimiento:

Funcionas de habilidad:

Nota: Las funciones implementadas del algoritmo genético se describen más adelante en la sección de algoritmos desarrollados.

# Descripción de las estructuras de datos desarrolladas:

Estructura de datos para la implementación del algoritmo genético

* Para modelar la población, la cual corresponde a los 22 jugadores; 11 en cada equipo se procedió a representarla como una lista con dos sublistas. Cada sublista es un equipo y cada equipo están compuestas por jugadores. Donde cada jugador se representa como una lista. La composición de un jugador consta de tres atributos codificados en binario y un ID para identificar si se trata de un portero, defensa, media cancha y delantero. En la figura 1 se puede apreciar la estructura de datos empleada para la población.

Diagrama

Descripción generada automáticamente

**Figura 1**. Estructura de datos para población

Para la implementación de movimientos

Estructura de datos para implementación de interfaz

# Descripción de los algoritmos desarrollados:

Algoritmo genético:

El algoritmo genético implementado en el proyecto consta de 5 etapas y una etapa inicial.

Primero se genera una población inicial con atributos aleatorios, una vez creada esta población inicia el algoritmo genético.

En la etapa “Population” se toma todos los elementos generados y les asigna un ID de player (1: portero, 2: defensa, 3: media cancha, 4: delanteros). Luego esta generación entra a la etapa de la aptitud donde se calcula su potencial.

En la etapa “Calculate fitness” se procede a recorrer cada equipo para analizar sus jugadores. La idea es que el jugador ideal sea aquel con la mayor cantidad de unos en sus atributos. Por consiguiente, se va a contar los bits

En la etapa “Selection” se procede a seleccionar jugadores con mejor aptitud. Se va a seleccionar 6 jugadores con mayor cantidad de unos. Para este caso no se toma en cuenta el ID del jugador, para esta sección es irrelevante el tipo de jugador y todos los individuos se tratan como iguales.

En la etapa “Reproduction” se requiere reproducir los jugadores con mejor aptitud. Entonces de los 6 jugadores seleccionados en la etapa anterior se procede a realizar un algoritmo para reproducir 5 jugadores más. El criterio de reproducción es tomar los primeros dos bits de cada atributo de cada jugador y combinarlo para así obtener un nuevo jugador. Por lo tanto, cada par de jugadores da como resultado un nuevo jugador, tal y como se observa en la figura 2.

Diagrama

Descripción generada automáticamente

**Figura 2.** Diagrama de reproducción

En la etapa “Mutation”, se procede a cambio un bit random de los atributos por su contrario. Eso corresponde a sumas un bit y descartar el overflow. En esta ultima etapa se recorre cada jugador y sus atributos y se muta en cada uno de ellos.

Se llega nuevamente a la etapa “Population”, aquí se alcanza la etapa recursiva donde se asignan los IDs a la población entrante.

Diagrama

Descripción generada automáticamente

**Figura 3**. Diagrama de flujo de algoritmo genético

# Problemas conocidos:

# Actividades realizadas por estudiantes:

# Problemas encontrados:

1. Uno de los problemas encontrados es que el algoritmo genético variaba las características de los jugadores, mas no en el juego en sí, es decir; el algoritmo genético no refleja los movimientos de los jugadores. Por lo que la parte grafica debía proponer una manera de que se visualizara un partido de manera práctica.
   1. Intentos de solución:
      1. Se realiza de manera independiente, y de esta manera no se considera viable la conexión entre esto y la población.
      2. Los movimientos se toman por aparte sin considerar el algoritmo genético. Pero de esta manera no se puede conectar la información del algoritmo genético, pues no se puede guardar como variable la población.
   2. Solución encontrada:
      1. Implementación de la función de movimiento en el flujo del algoritmo genético contrario a hacerlo de manera independiente en la interfaz.
   3. Recomendaciones:
      1. Considerar la implementación de las posiciones dentro del algoritmo genético, para mayor facilidad en el manejo del frontend.
   4. Conclusiones:
      1. Un buen manejo de parámetros para las funciones es beneficioso para tratarlas como variables y la información no se pierda.
2. La visualización de los movimientos de cada jugador ocasionó problemas cuando se quería actualizar el movimiento de un jugador. Para actualizar el movimiento de un jugador, se debe saber la posición de todos los demás jugadores.
   1. Intentos de solución:
      1. Se intenta pintar un jugador a la vez, pero en cada iteración lo jugadores desaparecen. Son imágenes intermitentes.
   2. Solución encontrada:
      1. Se debe repintar el canvas de la ventana y para ello, se deben guardar las posiciones antiguas de los jugadores para repintarlas o moverlos al mismo tiempo.
   3. Recomendaciones:
      1. Dado que se deben mover uno o más jugadores a la vez, se recomienda considerar todas las posiciones de los jugadores para repintar la imagen.
   4. Conclusiones:
      1. En Racket la animación se realiza pintando pixeles, por lo tanto, se debe considerar eliminar el rastro de pixeles.
3. El movimiento de los jugadores presenta una vibración cuando se están moviendo, ya que nunca alcanzan sus puntos finales debido no se considera un punto de culminación adecuado.
   1. Intentos de solución:
      1. Se consideraron varias condiciones de terminación, pero ninguna se adaptaba de la mejor manera.
   2. Solución encontrada:
      1. Se implementó una fórmula matemática la cual se fue modificando hasta llegar a la condición de terminación adecuada. Donde se consideran las diferencias de pixeles y las distancias limitantes de cada jugador.

**AGREGAR FUNCION MATEMATICA**

* 1. Recomendaciones:
     1. Considerar las diferencias de pixeles cuando se implementa el movimiento y considerar que un jugador es un conjunto de pixeles pintados.
     2. Considerar una formula de movimiento generar e irla modificando según las necesidades.
  2. Conclusiones:
     1. Se concluye que la mejor manera encontrada en el proyecto para sobrellevar el movimiento de jugadores es bajo un modelo matemático que considere la condición de terminación del algoritmo.

1. Implementación de algoritmo genético y

# Conclusiones y recomendaciones

# Bitácora: