**Instituto Tecnológico de Costa Rica**

**Área de Ingeniería en Computadores**

CE-3104: Lenguajes, Compiladores e Intérpretes

**Profesor:**

Marco Rivera Meneses

**Tarea 1: BlaCEjack**

Paradigma funcional

**Integrantes:**

Carlos Adrian Araya Ramirez

Michael Shakime Richards Sparks

José Andrés Solano Mora

**Primer semester 2021**

1. **Breve descripción del Proyecto**

BlaCEjack se es un juego basado en el juego de casino llamado Black Jack o también comúnmente llamado 21. Consiste en obtener una mejor puntuación que el crupier a partir de pedir cartas y aproximarse lo más posible a una puntuación de 21 sin pasarse.

* 1. **Descripción detallada de los algoritmos de solución desarrollados con sus respectivos diagramas.**

La acción o función principal de este juego consiste en repartir una carta aleatoria dentro de un mazo de 52 cartas diferentes a un jugador en momento dado, y analizar la condición de dicho jugador a partir de las cartas que posee.

En base a esta función, el flujo del programa consiste en primero seleccionar el numero de jugadores, según la cantidad de jugadores seleccionada se creará una lista de jugadores que funcionará de estructura de datos durante todo el juego.

Una vez creada la lista de jugadores, el algoritmo inmediatamente revuelve la cartas del mazo (hace un shuffle a la lista de cartas) y reparte dos cartas a cada jugador y al crupier, donde la primera carta repartida equivale a la carta boca abajo. Durante el reparto de estas primeras dos cartas no se hace ningún calculo sobre la puntuación de ningún jugador, solo se hace un cdr de la lista de cartas para evitar que se repitan.

Cuando todos los jugadores y el crupier tienen las primeras dos cartas comienza el primer turno, donde queda a la espera de que el jugador indique si desea solicitar una carta más o plantarse, si el jugador decide plantarse continuará el turno del siguiente jugador y si decide pedir una carta se evaluarán las siguientes condiciones.

* **Si la puntuación del jugador con las cartas visibles para todos es mayor a 21,** el jugador habrá perdido y continua el turno del siguiente jugador.
* **Si la puntuación del jugador con las cartas visibles para todos es menor a 21,** se le da una carta al jugador, es decir se agrega a su lista de cartas el car del mazo y se le hace un cdr al mazo para sacar la carta repartida.
* **Si es el turno del crupier,** se evalúan las mismas dos condiciones anteriores pero esta vez si se toma en cuenta la carta volteada.

El proceso de pedir una carta de se repite hasta que todos los jugadores se hayan plantado o hayan perdido y llegue el turno del crupier.

El turno del crupier es el turno final, por lo tanto, a partir de aquí cada vez que el crupier pide una carta, se analiza si obtiene un blackjack ya que en este caso ganaría irrefutablemente.

Finalmente, cuando el crupier decide plantarse o pierde, se hace un análisis de los puntajes de todos los jugadores y el crupier y se calcula quien es el ganador.

* 1. **Descripción de las funciones implementadas.**

1. (bCEj X)
2. (dar-carta jugador mazo)
3. (dar-carta-jugador jugador listaJugadores mazo)
4. (reparte-cartas listaJugadores mazo)
5. Funciones de shakime
6. (turno-crupier crupierDeck mazo)
   1. **Descripción de la ejemplificación de las estructuras de datos desarrolladas.**

La estructura de datos de los jugadores es una lista de jugadores, donde cada jugador es una lista, en la que su primer elemento (el car) es el nombre del jugador, el segundo elemento, es la puntuación actual equivalente a sus cartas, y su tercer y ultimo elemento es la lista de cartas.

**'(("Player1" 0 ((A 1) (A 2) (A 7))) ("Player2" 0 ((A 6) (A 5) (T 1))) ("Player3" 0 ((A 1) (A 7))))**

La lista de cartas de cada jugador sigue la regla de las cartas que se utiliza en el mazo del juego, donde cada carta es un par ordenado, donde el primer elemento es el palo de la carta y el segundo el valor de la carta.

* 1. **Problemas sin solución: En esta sección se detalla cualquier problema que no se ha podido solucionar en el trabajo.**
  2. **Problemas sin solución: En esta sección se detalla cualquier problema que no se ha podido solucionar en el trabajo**
  3. **Problemas sin solución: En esta sección se detalla cualquier problema que no se ha podido solucionar en el trabajo**
  4. **Conclusiones.**
  5. **Recomendaciones.**
  6. **Bibliografía consultada en todo el proyecto**