

Universidad Icesi

Proyecto Final – Algoritmos y Programacio II

CB- Soccer.

Requerimientos funcionales:

RF1. Inicialización del juego, Permite al usuario jugar una partida, logrando cargar los datos necesarios y crear una nueva partida con respectiva estructura del torneo haciendo referencia a las etapas del torneo (octavos, cuartos y final), Resultado esperado poder iniciar el torneo correctamente.

RF2. Seleccionar el equipo y el uniforme del jugador, teniendo en cuenta que primero se escoge el equipo y después el uniforme, Resultado esperado es que el jugador pueda seleccionar el equipo y uniforme deseado y poder así seguir con la secuencia del juego correctamente.

RF3. Generar los puntos del juego, Permite guardar los puntos del jugador teniendo en cuenta en donde quedó del torneo, con estos puntos el jugador tenga la oportunidad de competir para quedar en los mejores 10 jugadores del juego, finalmente se espera que se guarde correctamente en el juego el jugador con sus respectivos puntos.

RF4. Ordenamiento por nombres de los récords, este método permite comparar los jugadores ordenándolos en orden ascendente con respecto al abecedario. Resultado esperado los 10 mejores ordenados correctamente.

RF5. Ordenamiento por puntuación de los récords, este método permite comparar los jugadores ordenándolos en orden ascendente. Resultado esperado los 10 mejores ordenados correctamente.

RF6. Ordenamiento por nombres de los equipos, este método permite comparar los equipos ordenándolos en orden ascendente con respecto al abecedario. Resultado esperado es que cuando el jugador vaya a escoger los equipos estos están ordenados correctamente.

RF7. Ordenamiento por nombres de los uniformes, este método permite comparar los uniformes ordenándolos en orden ascendente con respecto al abecedario. Resultado esperado es que cuando el jugador vaya a escoger los uniformes estos están ordenados correctamente.

RF8. Generar resultados de los partidos de los demás oponentes, Permite generar aleatoriamente los resultados de los partidos de los oponentes del torneo y así determinar el oponente del jugador dependiendo si gano o perdió el partido jugado o a punto de jugar. Resultado esperado es tener correctamente generada la estructura de los resultados del torneo, donde se continuará con los equipos que pasaron la ronda.

RF9. Generar el oponente, este método permite al comienzo del torneo generar el oponente del jugador. Resultado esperado es que el jugador pueda tener un oponente y que el oponente

no se genere en otros partidos, cumpliendo así correctamente el objetivo de general el oponente.

RF10. Generar movimiento del Balón, permite generar que rebote tanto en los límites de la pantalla como en el jugador y el oponente, Resultado esperado es que el balón rebote correctamente respondiendo a la dirección de los rebotes y respetando los límites de la pantalla.

RF11. Generador de tiempo del partido, es llevar el tiempo del partido para poder darle un límite y poder que el jugador tenga en cuenta el tiempo para poder así hacer sus estrategias de juego. Resultado esperado que el cronómetro del juego funcione correctamente y que en su debido límite termine el partido para definir el ganador y el perdedor.

RF12. Permite llevar el marcador del partido, es llevar la cuenta de los goles marcados por los dos oponentes y así llevando el marcador donde uno pueda ver quien va ganando o perdiendo, Resultado esperado es que el marcador lleve correctamente la cuenta de los goles de cada equipo.

RF13. Generar movimiento del jugador, Permite que el jugador tenga un movimiento vertical en la pantalla permitiendo que el balón no entre en su arco y poder hacerla rebotar para que efectúe un golazo, Resultado esperado es que el usuario pulse las flechas haciendo que el jugador pueda tener movimiento en la cancha, teniendo en cuenta que tenga coherencia con la flecha que hundi6 el usuario y que cuando el jugador toque el balón que este rebote correctamente hacia el otro lado de la cancha.

RF14. Generar movimiento del jugador oponente, el oponente que es la “computadora” (el equipo a enfrentar) tenga un movimiento vertical logrando detener ocasionalmente las pelotas para que no le hagan gol haciendo real una simulaci6n de un contrincante bueno para el jugador, Resultado esperado es que el oponente tenga movimientos verticales y que cuando toque el balón este rebote hacia el otro lado de la cancha.

RF15. Generador de lista de puntajes, donde se sacan los 10 mejores jugadores hasta el momento en el juego, Resultado esperado es que la lista tenga coherencia con los puntajes de forma ascendente y que si sean los mejores 10, hay que tener en cuenta que si no se han jugado los 10 jugadores el método solo imprimirá los que estén hasta el momento, finalmente imprimiéndolos con un formato determinado que es primero el número de la posici6n en la lista, después el nombre y por último el puntaje del jugador.

RF16. Salir del juego, este método permite salir del juego, Resultado esperado que salga del juego correctamente.

RF17. Persistencia de los datos, este método permite guardar el puntaje de los jugadores mediante serialización, Resultado esperado es que los datos de antes de cerrar la aplicación se conservan a la hora de volver a abrir la aplicación.

RF18. Generar siguiente partido, Determinar si el jugador puede seguir jugando o no sigue en el torneo, entonces solo si gana genera el siguiente partido, Resultado esperado que el jugador gane el partido y pueda continuar tu torneo correctamente.

RF19. Impresión de árbol mostrando la etapa del torneo, este método permite darle al usuario una visualización del torneo donde imprime el árbol de la secuencia del torneo y sus etapas, Resultado esperado es que el árbol pueda imprimirse correctamente y pueda tener una buena coherencia con el torneo y los equipos que ganan y pasan a la siguiente etapa.

RF20. Cargar los archivos de texto, este método permite cargar los nombres de los equipos a el programa, Resultado esperado es que cuando el jugador vaya a escoger un equipo si hayan cargado correctamente.

Requerimientos no funcionales:

Req-1. Usabilidad de la aplicación. La aplicación deber ser intuitiva al uso y las tareas deben estar correctamente explicadas, siendo fácil de usa a simple vista.

Req-2. La plataforma no puede ser accedida directamente, sino a través de una interfaz diseñada para estos propósitos.

Req-3. Las excepciones deben ser reportadas por la propia aplicación en la medida de las posibilidades y no por el sistema operativo. Los mensajes del sistema deben estar en el idioma apropiado.

Req-4. El sistema deberá responder en el mínimo tiempo posible ante las solicitudes de los usuarios y el procesamiento de la información. La eficiencia de la aplicación estará determinada en gran medida por el aprovechamiento de los recursos que se disponen en el modelo.

Req-5. El sistema debe ser tolerante a los fallos inesperados por parte del sistema y los posibles errores de los diferentes usuarios.

Req-6. La aplicación debe poderse ejecutar en diferentes entornos, como Windows, Linux, MacOS, etc.

Req-7. La aplicación debe estar en la capacidad de almacenar los datos suministrados por el usuario, aun cuando la aplicación se haya cerrado y vuelva a ser iniciada tiempo después.

Req-8. Inicializar la aplicación no debería de tardar más de 1 minuto.

Req-9. Todas las operaciones búsqueda, selección, entre otras, deben realizadas por el software.