Bitacora

|  |  |  |
| --- | --- | --- |
| Fecha | Actividad | Participantes |
| 27/2/22 | Nos reunimos en llamada y creamos los documentos iniciales | Axel – Jan - Pablo |
| 1/3/22 | Iniciamos la programación creando carpeta RS y creando funciones | Axel - Jan |
| 5/3/22 | Se añada documento con utilidades para el algoritmo de listas | Pablo - Jan |
| 6/3/22 | Se añadieron más funciones y validaciones en el archivo de Utilidades.rkt | Jan |
| 6/3/22 | Se agrega archivo de gui con funciones para generar cubo | Pablo |
| 6/3/22 | Se agregan condiciones de verificación en archivos RS.rkt y en Utilidades.rkt | Axel |
| 7/3/22 | Se añade fondo, función de click y función de pintar en el GUI.rkt | Pablo |
| 7/3/22 | Se añade función de verificar condiciones sobre columnas y filas en RS.rkt | Axel |
| 7/3/22 | Se añade en condiciones para rotación de filas y columnas | Jan |
| 7/3/22 | Se añade función de rotar caras en el GUI.rkt | Pablo |
| 8/3/22 | Se crea función de cambiar filas en Utilidades.rkt | Jan – Pablo - Axel |
| 8/3/22 | Se crea función de update que se conecta con la parte grafica | Jan – Pablo - Axel |
| 8/3/22 | Función de rotación de filas-columnas-caras | Jan – Pablo - Axel |
| 9/3/22 | Documentación Interna | Jan – Pablo - Axel |
| 9/3/22 | Creación de manual y documentación técnico | Jan – Pablo - Axel |

* 1. Descripción de las funciones implementadas.

**Función RS:**

La función RS va a ser la primera función que llamaría el programa lo cual verifica condiciones para ver si las entradas que da el jugador están correctos y que puedan proceder a la parte grafica.

Las verificaciones se hacen en subfunciones que tienen que dar falso o verdadero los cuales son

(lenFilCol)

(verificarMovs)

(largo Matriz)

Estos se explicarán en algoritmos desarrollados.

**Función Update:**

La función update revisa primero que el tamaño del cubo este dentro del rango, segundo que la variable Cubo sea una lista y que no esté vacía, tercero que Movs sea una lista y que no esté vacía.

**Función fila-col?:**

Lee la primera instrucción que viene en la variable Movs(lista de movimientos) y la separa por fila o columna y dirección, luego verifica si coincide con alguna de las condiciones y si coincide, envía los parámetros de fila y columna a la siguiente función.

**Función fil-col-central?**:

La función determina si la fila/columna que se quiere mover se ubica en el "centro" o es una "esquina". Cuando hablamos de centro nos referimos a que solo cuatro de las seis caras sufren un cambio y esquinas cuando cinco de las seis caras sufren un cambio. Esta validación se hace con el número que viene en la primera instrucción dentro de la lista Movs.

**Función GUI:**

* 1. Descripción de las estructuras de datos desarrolladas.

Listas:

Matrices:

1.3. Descripción detallada de los algoritmos desarrollados.

**lenFilCol:** Función que verifica lo largo de las filas, las columnas y si el cubo tiene 6 caras.

Esta función se implementa mediante el algoritmo de **recursión de cola** y tiene que dar de salida un falso o un verdadero.

**verificarMovs**: Función que verifica que los movimientos tengan formato de (ej. (F1D)).

Esta función se implementa mediante el algoritmo de **recursión de pila** y tiene que dar de salida un falso o un verdadero.

**actualizar-fila:** Hace el cambio de fila ya sea hacia la izquierda o hacia la derecha

La función de actualizar-fila, primero a que corresponde el parámetro “dir” rota una fila indicada hacia la izquierda

**rotarDer:** Realiza una rotación hacia la derecha del cubo

Esta función se encarga de rota una de las “esquinas” indicadas, se llama tres veces a la función de actualizar-fila(la cual mueve una fila hacia la izquierda), esto se debe a que, en un cubo, rotar una vez a la derecha es lo mismo que rotar 3 veces a la izquierda y viceversa.

1.4. Problemas sin solución: En esta sección se detalla cualquier problema que no se ha podido solucionar en el trabajo.

1.5. Plan de Actividades realizadas por estudiante: Este es un planeamiento de las actividades que se realizarán para completar la tarea, este debe incluir descripción de la tarea, tiempo estimado de completitud, fecha de entrega y responsable a cargo.

|  |  |
| --- | --- |
| Roles Asignados | Participante |
| Creación de condiciones de verificación y movimientos del cubo | Axel |
| Creación de Utilidades y movimientos del cubo | Jan |
| Creación de GUI y movimientos del cubo | Pablo |

|  |  |  |
| --- | --- | --- |
| Fecha | Actividades | Participantes |
| Semana1 – 2 dias | Hacer reunión e investigar sobre la parte lógica y grafica de Racket. | Axel – Pablo - Jan |
| Semana1 – 2 dias | Empezar a crear funciones de verificar y el GUI | Axel - Pablo |
| Semana 1 – 3 dias | Crear un archivo de función útiles que nos facilitan la tarea | Pablo - Jan |
| Semana 2 - 2 dias | Crear Gui que pinte líneas, cubos y caras de un cubo rubik | Pablo |
| Semana 2 – 2 dias | Condiciones y movimientos de cubo Rubik completado con GUI | Axel – Pablo - Jan |
| Semana 2 – 3 dias | Arreglar errores y mejorar condiciones | Axel – Pablo - Jan |

1.6. Problemas encontrados: descripción detallada, intentos de solución sin éxito, soluciones encontradas con su descripción detallada, recomendaciones, conclusiones y bibliografía consultada para este problema específico.

1.7. Conclusiones y Recomendaciones del proyecto.

1.8. Bibliografía consultada en todo el proyecto