1) Requerimientos Funcionales

|  |  |
| --- | --- |
| **Nombre:** | R. #1. Iniciar juego |
| **Resumen:** | El programa debe permitir leer un m y un n que serás las dimensiones de la cuadricula y además recibir un k para la cantidad de espejos, este valor no puede ser >= mxn y por último un nickname |
| Entradas: | Nickname, filas, columnas y espejos |
|  |  |
| Resultados: | Iniciar juego con cuadricula mxn, cantidad de espejos y un jugador |
|  |  |

|  |  |
| --- | --- |
| **Nombre:** | R. #2. Establecer nomenclatura para la cuadricula |
| **Resumen:** | El programa debe permitir interpretar la nomenclatura para indicar la celda que se necesita |
| Entradas: | Ninguna |
|  |  |
| Resultados: | Nomenclatura cuadricula |
|  |  |

|  |  |
| --- | --- |
| **Nombre:** | R. #3. Colocar inclinación y ubicación de los espejos |
| **Resumen:** | El programa permite colocar aleatoriamente la inclinación y la ubicación de los espejos “/”, “\” |
| Entradas: | Ninguna |
|  |  |
| Resultados: | Inclinación de los espejos en la cuadricula aleatoriamente |
|  |  |

|  |  |
| --- | --- |
| **Nombre:** | R. #4. Disparar rayo laser |
| **Resumen:** | El programa permite disparar un rayo de forma horizontal o vertical si el usuario lo desea, indicando además la celda por donde inicia. |
| Entradas: | Horizontal o vertical, celda |
|  |  |
| Resultados: | Disparar rayo laser |
|  |  |

|  |  |
| --- | --- |
| **Nombre:** | R. #5. Indicar la celda por donde sale el rayo |
| **Resumen:** | El programa indica la celda por donde sale el rayo después de hacer el recorrido con ayuda de los espejos |
| Entradas: | Ninguna |
|  |  |
| Resultados: | Celda por donde salió |
|  |  |

|  |  |
| --- | --- |
| **Nombre:** | R. #6. Visualizar inicio y final |
| **Resumen:** | El programa debe indicar con una S el inicio de la celda por donde ingresa el rayo y una E por donde sale el rayo |
| Entradas: | Ninguna |
|  |  |
| Resultados: | S y E en la cuadricula |
|  |  |

|  |  |
| --- | --- |
| **Nombre:** | R. #7. Encontrar espejo |
| **Resumen:** | El programa debe permitir al usuario escoger donde se encuentra el espejo |
| Entradas: | Ninguna |
|  |  |
| Resultados: | SI: Se visualiza el espejo |
|  | No: Se muestra una X en la celda escogida |

|  |  |
| --- | --- |
| **Nombre:** | R. #8. Mostrar espejos faltantes |
| **Resumen:** | El programa debe permitir mostrar en la parte superior izquierda la cantidad faltante de espejos a encontrar. |
| Entradas: | Ninguna |
|  |  |
| Resultados: | Espejos faltantes |
|  |  |

|  |  |
| --- | --- |
| **Nombre:** | R. #9. Almacenar puntaje |
| **Resumen:** | El programa debe permitir almacenar en un árbol binario de búsqueda |
| Entradas: | Ninguna |
|  |  |
| Resultados: | Puntaje almacenado |
|  |  |

|  |  |
| --- | --- |
| **Nombre:** | R. #10. Mostrar puntaje |
| **Resumen:** | El programa debe permitir recorrer el árbol binario de búsqueda en inorder para encontrar el resultado esperado |
| Entradas: | Ninguna |
|  |  |
| Resultados: | Puntaje |
|  |  |

|  |  |
| --- | --- |
| **Nombre:** | R. #11. Permitir terminar programa |
| **Resumen:** | Permitir finalizar el programa cuando el usuario lo indique |
| Entradas: | Ninguna |
|  |  |
| Resultados: | Programa finalizado |
|  |  |

2) Diagrama de clases Diagrama, Esquemático

Descripción generada automáticamente