**Reporte Sprint #3**

Implementen todas las características que permiten a un jugador humano jugar un juego SOS simple o general contra un oponente humano y **refactoricen su código existente si es necesario**. Las características mínimas incluyen elegir el modo de juego (simple o general), elegir el tamaño del tablero, configurar un nuevo juego, hacer un movimiento (en un juego simple o general) y determinar si un juego simple o general ha terminado. El siguiente es un diseño de GUI de muestra.

Se requiere el uso de una jerarquía de clases para hacer frente a los requisitos comunes del juego simple y general. Si tu código para Sprint 2 no ha considerado la jerarquía de clases, es hora de refactorizar su código.

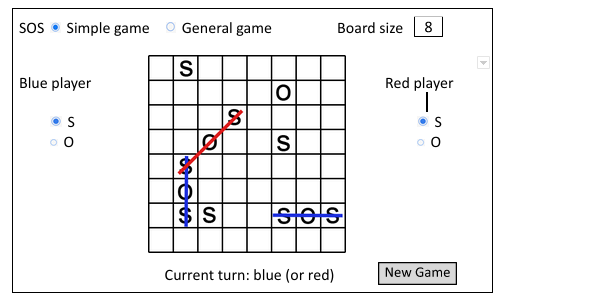


Figura 1. Diseño de GUI de muestra del programa en el Sprint 3

**Entregables: expande y mejora tu entrega para el sprint 2.**

1. **Demostración (6 puntos)**

Envíen un video de no más de cinco minutos, que demuestre claramente las siguientes características.

1. Un juego simple en el que el jugador azul es el ganador.
2. Un juego simple empatado con el mismo tamaño de tablero que es 1
3. Un juego general en el que el jugador rojo es el ganador y el tamaño del tablero es diferente de 1
4. Un juego general empatado con el mismo tamaño de tablero que es 1
5. Algunas pruebas unitarias automatizadas para el modo de juego simple
6. Algunas pruebas unitarias automatizadas para el modo de juego general

En el video, debes explicar lo que se está demostrando.

1. **Resumen del código fuente (2 puntos)**

| Nombre del archivo de código fuente | ¿Código de producción o de prueba? | # lineas de código |
| --- | --- | --- |
| Tablero | código de producción | 400 |
| Tableroconsola | código de producción | 29 |
| GUI | código de producción |  |
| testAsignacionTurno | código de prueba | 75 |
| testMovimientos | código de prueba | 106 |
| testTableroVacio | código de prueba | 18 |
| testVictoria | código de prueba | 203 |
| Total | |  |

**Deben enviar todo el código fuente para obtener más puntos por esta tarea.**

1. **Código de producción vs Historias de usuario/Criterio de aceptación (4 puntos)**

Resuman cómo se implementa cada uno de los siguientes criterios de aceptación/historia de usuario en tu código de producción (nombre de clase y nombre de método, etc.)

| **ID de historia de usuario** | **Nombre de historia de usuario** |
| --- | --- |
| 1 | Selección del tamaño del tablero |
| 2 | Elección del modo de juego |
| 3 | Inicia el juego nuevo con el tamaño del tablero elegido |
| 4 | Hacer un movimiento en un juego simple |
| 5 | Asignación de turno en el juego simple |
| 6 | Un juego simple ha terminado |
| 8 | Elección de las letras “S” o “O” |
| 9 | Formación de la palabra “SOS” |
| 10 | Conteo de las palabras “SOS” |
| 11 | Hacer un movimiento en el juego general |
| 12 | Asignación de turnos en un juego general |
| 13 | Un juego general ha terminado |
| 14 | Asignación de un ganador en un juego general |

| **Nombre y ID de la historia usuario** | **AC ID** | **Nombre clase(s)** | **Nombre Método(s)** | **Estatus (completo o no)** | **Notas (opcional)** |
| --- | --- | --- | --- | --- | --- |
| 1 Selección del tamaño del tablero | 1.1 | Tablero | setNumFilas, setNunColumnas | Completo |  |
| 1.2 | Tablero | Tablero, initTablero, setNumFilas, setNunColumnas | Completo |  |
| 2 Elección del modo de juego | 2.1 | Tablero | getModo | Completo |  |
| 2.2 | Tablero | getModo | Completo |  |
| 3 Inicia el juego nuevo con el tamaño del tablero elegido | 3.1 | Tablero | Tablero, initTablero, setNumFilas, setNunColumnas | Completo |  |
| 3.2 | Tablero | Tablero, initTablero, setNumFilas, setNunColumnas | Completo |  |
| 4 Hacer un movimiento en un juego simple | 4.1 | Tablero | Gana, CeldaValida | Completo |  |
| 4.2 | Tablero | Gana, CeldaValida | Completo |  |
| 5 Asignación de turno en el juego simple | 5.1 | Tablero | getTurno, Gana | Completo |  |
| 6 Un juego simple ha terminado | 6.1 | Tablero | ActualizarEstadoDeJuego, Gana | Completo |  |
| 6.2 | Tablero | ActualizarEstadoDeJuego, Gana | Completo |  |
| 8 Elección de las letras “S” o “O” | 8.1 | Tablero | setSeleccion | Completo |  |
| 8.2 | Tablero | setSeleccion | Completo |  |
| 9 Formación de la palabra “SOS” | 9.1 | Tablero | Gana | Completo |  |
| 9.2 | Tablero | Gana | En progreso |  |
| 10 Conteo de las palabras “SOS” | 10.1 | Tablero | Gana | Completo |  |
| 10.2 | Tablero | Gana | En progreso |  |
| 11 Hacer un movimiento en el juego general | 11.1 | Tablero | setSeleccion, setModo, CeldaValida | Completo |  |
| 11.2 | Tablero | setSeleccion, setModo, CeldaValida | Completo |  |
| 12 Asignación de turnos en un juego general | 12.1 | Tablero | getTurno, Gana | Completo |  |
| 12.2 | Tablero | getTurno, Gana | Completo |  |
| 13 Un juego general ha terminado | 13.1 | Tablero | ActualizarEstadoDeJuego, Gana | En progreso |  |
| 14 Asignación de un ganador en un juego general | 14.1 | Tablero | ActualizarEstadoDeJuego, Gana | En progreso |  |

1. **Pruebas vs Historias de usuario/Criterio de aceptación (4 puntos)**

Resuman cómo cada uno de los criterios de aceptación/historia de usuario es probado por su código de prueba (nombre de clase y nombre de método) o pruebas realizadas manualmente.

| **User Story ID** | **User Story Name** |
| --- | --- |
| 1 | Selección del tamaño del tablero |
| 2 | Elección del modo de juego |
| 3 | Inicia el juego nuevo con el tamaño del tablero elegido |
| 4 | Hacer un movimiento en un juego simple |
| 5 | Asignación de turno en el juego simple |
| 6 | Un juego simple ha terminado |
| 8 | Elección de las letras “S” o “O” |
| 9 | Formación de la palabra “SOS” |
| 10 | Conteo de las palabras “SOS” |
| 11 | Hacer un movimiento en el juego general |
| 12 | Asignación de turnos en un juego general |
| 13 | Un juego general ha terminado |
| 14 | Asignación de un ganador en un juego general |

4.1 Pruebas automatizadas que corresponden directamente a los criterios de aceptación de las historias de usuario anteriores

| **Nombre y ID de la historia usuario** | **AC**  **ID** | **Nombre Clase (s) del código de prueba** | **Nombre método(s) del código Prueba** | **Descripción de los casos de prueba (entrada & salida esperada)** |
| --- | --- | --- | --- | --- |
| 1 Selección del tamaño del tablero | 1.1 | testTableroVacio | testSizeTableroValido | Contiene dos assertEquals. El tamaño esperado para ambos es 4, si el método getNumFilas y getNumColumnas del objeto tablero retorna el mismo valor del tamaño esperado, entonces las pruebas pasarán (estarán en color verde) |
| 1.2 | testTableroVacio | testSizeTableroInvalido | Contiene dos assertEquals. El tamaño esperado para ambos es 1, si el método getNumFilas y getNumColumnas del objeto tablero retorna el mismo valor del tamaño esperado, entonces las pruebas pasarán (estarán en color verde) |
| 2 Elección del modo de juego | 2.1 | testModoDeJuego | testModoDeJuegoSimple | Contiene un assertEqual. El valor esperado es un caracter S, si el método getSeleccion retorna el mismo valor esperado, entonces la prueba pasará (estará en color verde) |
| 2.2 | testModoDeJuego | testModoDeJuegoGeneral | Contiene un assertEqual. El valor esperado es un caracter G, si el método getSeleccion retorna el mismo valor esperado, entonces la prueba pasará (estará en color verde) |
| 3 Inicia el juego nuevo con el tamaño del tablero elegido | 3.1 | testTableroVacio | testTableroVacio | Contiene un assertEqual dentro de dos for anidados. Se espera que todos los casilleros estén vacíos, de ser así se inicia un juego con el tamaño elegido en el modo Simple |
| 3.2 | testTableroVacio | testTableroVacio | Contiene un assertEqual dentro de dos for anidados. Se espera que todos los casilleros estén vacíos, de ser así se inicia un juego con el tamaño elegido en el modo General |
| 4 Hacer un movimiento en un juego simple | 4.1 | testMovimientos | testMovimientoSimpleValidoFilaColumna | Contiene dos assertEquals. El valor esperado es true, si el método CeldaValida del objeto tablero (se les pasó como argumentos a los métodos una fila y columna dentro del rango permitido (0,0) y (2,2), el tamaño del tablero es de 4x4) retorna el mismo valor esperado, entonces pasarán las pruebas (estará en color verde) |
| 4.2 | testMovimientos | testMovimientoSimpleInvalidoFila, testMovimientoSimpleInvalidoColumna | Cada método contiene dos assertEquals. El valor esperado es false, si el método CeldaValida del objeto tablero (se les pasó como argumentos a los métodos una fila y columna fuera del rango permitido (-1,3), (7,3), (3, -1) y (3, 9), el tamaño del tablero es de 4x4) retorna el mismo valor esperado, entonces pasarán las pruebas (estará en color verde) |
| 5 Asignación de turno en el juego simple | 5.1 | testAsignacionTurno | testAsignacionDeTurnoEnJuegoSimple | Por defecto el jugador Azul inicia, este método asigna el modo simple, selecciona el caracter S y realiza un movimiento a la fila 1, columna 2, en el assertEqual el valor esperado es el caracter R, si el getTurno del objeto tablero retorna el mismo valor esperado, entonces pasará la prueba |
| 6 Un juego simple ha terminado | 6.1 | testVictoria | testVictoriaJuegoSimpleRojo, testVictoriaJuegoSimpleAzul | Ambos métodos preparan el escenario donde el jugador Rojo y Azul formarán una cadena SOS. El valor esperado es ROJO\_GANA, AZUL\_GANA, respectivamente, si el método getEstadoActual del objeto retorna el mismo valor esperado, entonces las pruebas pasarán |
| 6.2 | testVictoria | testEmpateJuegoSimple | Se prepara el escenario para generar el empate, el valor esperado del assertEqual es DRAW, si el método getEstadoActual retorna el mismo valor esperado, entonces la prueba pasará |
| 11 Hacer un movimiento en el juego general | 11.1 | testMovimientos | testMovimientoGeneralValido | Se prepara el escenario donde la casilla (2,3) contenga AZUL\_O, el assetEqual tiene el valor esperado AZUL\_O, si el método getContenidoCeldas(2,3) retorna el mismo valor esperado, entonces la prueba pasará |
| 11.2 | testMovimientos | testMovimientoGeneralInvalido | Se prepara el escenario donde la casilla (2,3) contenga AZUL\_O, el primer assetEqual tiene el valor esperado AZUL\_O, si el método getContenidoCeldas(2,3) retorna el mismo valor esperado, entonces la prueba pasará, luego el segundo assertEqual tiene valor esperado false, si el método CeldaValida(2,3) retorna el mismo valor esperado, entonces la prueba pasará |
| 12 Asignación de turnos en un juego general | 12.1 | testAsiganacionTurno | testAsignacionDeTurnoEnJuegoGeneral | El método prepara el escenario donde el jugador AZUL colocó S en la casilla (1,2), El valor esperado del assertEqual es el caracter R, si el método getTurno retorna el mismo valor esperado, entonces la prueba pasará |
| 12.2 | testAsiganacionTurno | testRepeticionDeTurnoEnJuegoGeneral | El método prepara el escenario donde el jugador AZUL forma una cadena SOS, luego el assertEqual verificará que el turno del jugador AZUL continue |

4.2 Pruebas manuales que corresponden directamente a los criterios de aceptación de las historias de usuario anteriores

| **Nombre y ID de la historia usuario** | **AC**  **ID** | **Entrada de caso de prueba** | **Salida esperada** | **Notas** |
| --- | --- | --- | --- | --- |
| 1 Selección del tamaño del tablero | 1.1 | int 4 | int 4 |  |
| 1.2 | int 1 | int 1 |  |
| 2 Elección del modo de juego | 2.1 | char S | char S |  |
| 2.2 | char G | char G |  |
| 3 Inicia el juego nuevo con el tamaño del tablero elegido | 3.1 | VACIO | VACIO |  |
| 3.2 | VACIO | VACIO |  |
| 4 Hacer un movimiento en un juego simple | 4.1 | VACIO  char A | VACIO  char A |  |
| 4.2 | SIMBOLO\_AZUL\_S  char R | SIMBOLO\_AZUL\_S  char R |  |
| 5 Asignación de turno en el juego simple | 5.1 | char R | char R |  |
| 6 Un juego simple ha terminado | 6.1 | ROJO\_GANA  AZUL\_GANA | ROJO\_GANA  AZUL\_GANA |  |
| 6.2 | DRAW | DRAW |  |
| 11 Hacer un movimiento en el juego general | 11.1 | AZUL\_O | AZUL\_O |  |
| 11.2 | AZUL\_O  false | AZUL\_O  false |  |
| 12 Asignación de turnos en un juego general | 12.1 | char R | char R |  |
| 12.2 | char R | char R |  |

4.3 Otras pruebas automatizadas o manuales que no corresponden a los criterios de aceptación de las historias de usuario anteriores

| **Número** | **Entrada prueba** | **Resultado esperado** | **Nombre de clase del código de prueba** | **Nombre del método del código de prueba** |
| --- | --- | --- | --- | --- |
| 1 |  |  | testGUI | testTableroVacio |

5. Describe cómo la jerarquía de clases en tu diseño trata con los requisitos comunes y diferentes del juego simple y el juego general. (4 puntos)