

Reporte Sprint #2

Implementen las siguientes características del juego SOS: (1) los componentes básicos para las opciones del juego (tamaño del tablero y modo de juego) y el juego inicial, y (2) la ubicación del S/O para jugadores humanos sin verificar la formación de SOS o determinar el ganador. La siguiente es una interfaz de muestra. Se recomienda enfáticamente la implementación de una GUI. Deben practicar la programación orientada a objetos, haciendo que su código sea fácil de extender. Es importante separar el código de la interfaz de usuario y el código de la lógica del juego en diferentes clases (consulta el ejemplo de TicTacToe). Se requieren pruebas de xUnit.

SOS ☒ Simple game ☐ General game

Board size

Blue player

☒ S
☐ O

O							
		S	O	S			
				S			
							S

Red player

☒ S
☐ O

Current turn: blue (or red)

Figura 1. Diseño de GUI de muestra del programa en el Sprint 2

Entregables:

1. Demostración (8 puntos)

Envíen un video de no más de tres minutos, donde demuestran claramente que implementaron las funciones requeridas y escribieron algunas pruebas unitarias automatizadas. En el video, deben explicar lo que se está demostrando.

	Característica	Detalles
1	Escoge el tamaño del tablero	El jugador tendrá un text file para poder ingresar el tamaño del tablero en el que desee jugar.
2	Escoge el modo del juego	En la interfaz de usuario se muestran botones que sirven para especificar qué modo de juego se desea jugar.
3	Juego inicial del tamaño de tablero elegido y modo de juego	Se mostrará en la parte inferior de la ventana del juego las barras que forman el tablero de juego con el tamaño del tablero elegido.

4	Movimientos “S”	El usuario para poder realizar el movimiento “S” tendrá que hacer uso del botón con el mismo nombre del movimiento, esto solo se le permitirá hacer en su turno.
5	Movimientos “O”	El usuario para poder realizar el movimiento “O” tendrá que hacer uso del botón con el mismo nombre del movimiento, esto solo se le permitirá hacer en su turno.
6	Turno de cada jugador	En la parte superior derecha de la ventana del juego se indicará el turno de cada jugador; a su vez, se inhabilitarán los botones del jugador cuyo turno no esté activo.

2. Resumen del código fuente (2 puntos)

Nombre del archivo de código fuente	¿Código de producción o de prueba?	# líneas de código
GUI	código de producción	232
testGUI	código de prueba	26
Tablero	código de producción	126
testTableroVacio	código de prueba	30
testMovimiento	código de prueba	36
testModoDeJuego	código de prueba	22
Total		

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

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

Actualicen sus historias de usuario y los criterios de aceptación de la asignación anterior y asegúrense de que capturan adecuadamente los requisitos. 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	Escoge el tamaño del tablero
2	Escoge el modo de juego de un tablero escogido
3	Comienza un nuevo juego del tamaño de tablero y del modo de juego elegidos
4	Hacer un movimiento en un juego simple
5	Asignación de turno en el juego simple
6	Un juego simple ha terminado

Nombre y ID de la historia usuario	AC ID	Nombre clase(s)	Nombre Método(s)	Estatus (completo o no)	Notas (opcional)
1 Escoge el tamaño del tablero	1.1	Tablero	Tablero	Completo	

2testSizeTableroValido

	1.2	Tablero	getContenidoDeLasCeldasDelTablero	Completo	
2 Escoge el modo de juego de un tablero escogido	2.1	Tablero	getSelccion	Completo	
	2.2	Tablero	getSelccion	Completo	
3 Comienza un nuevo juego del tamaño de tablero y del modo de juego elegidos	3.1			No iniciado	
	3.2			No iniciado	
4 Hacer un movimiento en un juego simple	4.1			No iniciado	
	4.2			No iniciado	
5 Asignación de turno en el juego simple	5.1			No iniciado	
6 Un juego simple ha terminado	6.1			No iniciado	
	6.2			No iniciado	

4. Pruebas vs Historias de usuario/Criterio de aceptación (6 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	Escoge el tamaño del tablero
2	Escoge el modo de juego de un tablero escogido
3	Comienza un nuevo juego del tamaño de tablero y del modo de juego elegidos
4	Hacer un movimiento en un juego simple
6	Hacer un movimiento 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 Escoge el tamaño del tablero	1.1	testTableroVacio		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 tablero1 retorna el mismo valor del tamaño esperado, entonces las pruebas pasarán (estarán en color verde)
2 Escoge el modo de juego de un tablero escogido	2.1	testModoDeJuego	testModoDeJuegoSimple	Contiene un assertEquals. El valor esperado es un carácter S, si el método getSeleccion retorna el mismo valor esperado, entonces la prueba pasará (estará en color verde)
	2.2	testModoDeJuego	testModoDeJuegoGeneral	Contiene un assertEquals. El valor esperado es un carácter G, si el método getSeleccion retorna el mismo valor esperado, entonces la prueba pasará (estará en color verde)
4 Hacer un movimiento en un juego simple	4.1	testMovimiento	testMovimientoValidoModoSimple	Contiene dos assertEquals. El primero verifica si la casilla está vacía haciendo una comparación del valor esperado y el valor actual y el segundo verifica si el turno le corresponde al jugador A (Azul)
	4.2	testMovimiento	testMovimientoInvalidoModoSimple	Primero se asigna el caracter S a la variable seleccion, luego hace un movimiento a la posición fila 0, columna 0. Contiene dos assertEquals, el primero verifica si

4testSizeTableroValido

				valor esperado (simbolo azul S) es igual al valor actual y el segundo verifica si el turno era del jugador R (rojo)
--	--	--	--	--

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 Escoge el tamaño del tablero	1.1	int 4	int 4	
	1.2	int 1	int 1	
2 Escoge el modo de juego de un tablero escogido	2.1	char S	char S	
	2.2	char G	char G	
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	

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