

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		
1	Escoge el tamaño del tablero		
2	Escoge el modo del juego		
3	Juego inicial del tamaño de tablero elegido y modo de juego		
4	Movimientos "S"		
5	Movimientos "O"		
6	Jugador 1 gana juego simple.		
7	Pruebas unitarias		

2. Resumen del código fuente (2 puntos)

Nombre del archivo de código fuente	¿Código de producción o de prueba?	# líneas de código
Board	Producción	146
Console	Producción	76
GUI	Producción	365
TestBoard	Prueba	63
TestBoardConsole	Prueba	28
Total		678

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	Ingrese el tamaño del tablero GUI
3	Escoge el modo de juego de un tablero escogido
	Comienza un nuevo juego con el tamaño y modo de juego elegidos
4	Hacer un movimiento en un juego.
7	Formar SOS.
8	Juego 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 de tablero	AC 1.1<Tablero de tamaño NxN> Dado el usuario antes de iniciar en juego Cuando inicie el programa de juego Entonces se le debe pedir que ingrese el tamaño deseado del tablero Y el tablero debe ser inicializado con ese tamaño	Board	Board(int squaresPerSide)	Si	El tamaño del tablero es pasado al constructor de la clase Board como argumento.
2. Ingrese el tamaño del tablero GUI	AC 1.1<Tablero de tamaño NxN> Dado visualizado el contenido del juego Cuando el jugador quiera cambiar el	GameContent GameBoard	addContentTop() drawBoard(int rows, int columns) replace()	Si	El tamaño del tablero por defecto es 3 pero al ingresar un nuevo tamaño y dar enter se crea un

	tamaño del tablero por defecto Entonces debe ingresar el tamaño Y el tablero debe ser visualizarse con ese tamaño.				nuevo tablero vacío del tamaño ingresado
2.Escoge el modo de juego de un tablero escogido	AC 2.1<Elegir modo de juego GENERAL/SIMPLE> Dado el usuario antes de iniciar en juego Cuando inicie el programa de juego Entonces se le debe pedir que ingrese el modo deseado de juego Y la condición de victoria será ajustada acordeamente.	Console	Console.play()	Si	El modo de juego es determinado por la variable isGameSimple en el método Console.play() . El modo de juego solamente afecta la condición de victoria.
2.Escoge el modo de juego de un tablero escogido GUI	AC 2.1<Elegir modo de juego GENERAL/SIMPLE> Dado el usuario al querer realizar un movimiento Cuando el juego está iniciado con un tablero por defecto Entonces se le pedirá que elija el modo de juego.	GameContent	addContentTop()	Si	Al elegir el modo de juego ya no se podrá cambiar hasta que inicie otro juego nuevo
3.Comienza un nuevo juego con el tamaño y modo de juego elegidos	AC 3.1<Iniciar juego en la consola> Dado un usuario que acaba de ingresar el tamaño de tablero deseado Cuando ingrese el modo de juego que desea Entonces la consola debe imprimir el tablero con el tamaño deseado e iniciar el juego con el turno del jugador 1.	Console	Console.play()	Si	En la función play() el programa inicializa las variables necesarias para luego entrar al loop del juego en sí.
	AC 3.2<Iniciar juego en la GUI> Dado un usuario que acaba de ingresar el tamaño de tablero deseado	GUI	GUI.main()	Si	

	<p>Cuando ingrese el modo de juego que desea</p> <p>Entonces GUI debe redimensionar el tablero de juego a lo indicado e iniciar el juego en el modo indicado.</p>				
4.Hacer un movimiento	<p>AC 4.1<El jugador 1 / 2 hace un movimiento></p> <p>Dado un juego en curso</p> <p>Cuando jugador 1 / 2 ingrese la fila, columna y símbolo deseado</p> <p>Entonces dibujará el símbolo elegido en la posición indicada</p> <p>Y se cambiará el turno si es que no ha formado un SOS</p>	Board	makePlay(int row, int column, char chosen)	Si	El método makePlay() conoce el turno actual, pues el turno es un parámetro de la clase Board, de la cual makePlay() es un método.
5.Formar SOS	<p>AC 5.1<Un jugador forma un SOS en modo GENERAL></p> <p>Dado un juego curso con la posibilidad de formar un SOS</p> <p>Cuando un jugador completa el(los) SOS con el ingreso de una 'S' u 'O'</p> <p>Entonces el juego contabiliza la cantidad de SOSs que acaba de formar</p> <p>Y se agregan al puntaje del jugador.</p>	Board	hoyManySOS()	Si	El método howManySOS() recibe como argumentos los parámetros de la última jugada e inspecciona cuadrículas adyacentes para contar los SOS formados. El método devuelve un variable tipo INT points , que se usará para contabilizar el score de cada jugador.
	<p>AC 5.2<Un jugador forma un SOS en modo SIMPLE></p> <p>Dado un juego curso con la posibilidad de formar un SOS</p> <p>Cuando un jugador completa el(los) SOS con el ingreso de una 'S' u 'O'</p> <p>Entonces el juego termina y se designa como ganador al jugador activo.</p>	Board	howManySOS()	Si	Si el modo de juego está en SIMPLE, y además howManySOS() devuelve un número mayor que 0, entonces el juego termina.
6. Juego terminado	<p>AC 6.1<Un juego general termina></p>	Board	isGameFinished()	Si	isGameFinished() revisa si alguna celda del tablero aún está vacía, si

	Dado un juego curso con solo una casilla libre. Cuando el jugador activo llena la última casilla con 'S' u 'O'. Entonces el juego termina y se designa como ganador al que haya formado más SOSs				no hay ninguna entonces devuelve true ; caso contrario, devuelve false .
	AC 6.2<Un juego simple termina> Dado un juego curso con la posibilidad de formar un SOS Cuando un jugador completa el(los) SOS con el ingreso de una 'S' u 'O' Entonces el juego termina y se designa como ganador al jugador activo.				

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

4.2 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 de tablero	AC 1.1<Tablero de tamaño NxN>	Una fila y columna válidos	'.'	
2. Escoge el modo de juego de un tablero escogido	AC 2.1<Elegir modo de juego GENERAL/ SIMPLE>			
	...			
3. Comienza un nuevo juego con el tamaño y	AC 3.1<Iniciar juego en la consola>			

modo de juego elegidos				
4.Hacer un movimiento	AC 4.1<El jugador 1 / 2 hace un movimiento>	testMakePlay() Una fila y columna válida, además un carácter 'S' u 'O'	'S'	
5.Formar SOS	AC 5.1<Un jugador forma un SOS en modo GENERAL>	testHowManySOS() Una fila y columna válida, además un carácter 'S' u 'O' que completen un SOS	1	
	AC 5.2<Un jugador forma un SOS en modo SIMPLE>			
6. Juego terminado	AC 6.1<Un juego general termina>	testIsGameFinishedTrue() La entrada es un tablero lleno.	true	
	AC 6.2<Un juego simple termina>			