

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.

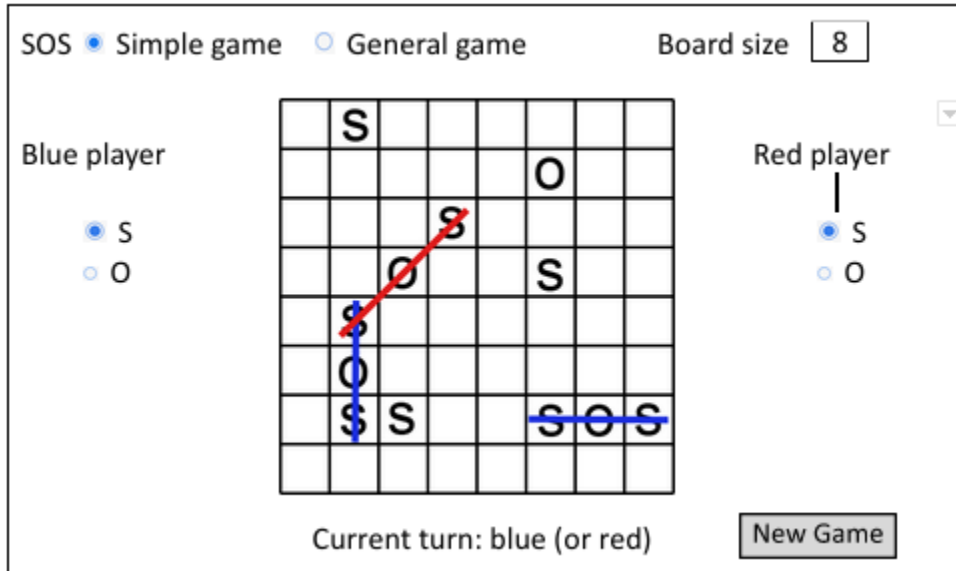


Figura 1. Diseño de GUI de muestra del programa en el Sprint 3

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.

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

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

2. Resumen del código fuente (2 puntos)

Nombre del archivo de código fuente	¿Código de producción o de prueba?	# líneas de código
Proyecto/Pruebas/TestGameMode	Código de prueba	37
Proyecto/Pruebas/TestNewGame	Código de prueba	30
Proyecto/Pruebas/TestSizeBoard	Código de prueba	50
Proyecto/Pruebas/TestFinishSimpleGame	Código de prueba	89
Proyecto/Pruebas/TestFinishGeneralGame	Código de prueba	22
Proyecto/Pruebas/TestMoveGeneralGame	Código de prueba	52
Proyecto/Pruebas/TestMoveSimpleGame	Código de prueba	47
Proyecto/Board	Código de producción	17
Proyecto/GuiSOS	Código de producción	453
Total		797

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)

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	Realizar un nuevo juego del tamaño de tablero y del modo de juego elegidos
4	Hacer un movimiento en un juego simple
5	Un juego simple a terminado
6	Hacer un movimiento en un juego general
7	Un juego general a terminado

Nombre y ID de la historia usuario	AC ID	Nombre clase(s)	Nombre Método(s)	Estatus (completo o no)	Notas (opcional)
1	1.1	GuiSOS	SizeN	completado	Retorna n
1	1.2	GuiSOS	SizeN	completado	Retorna n+1
1	1.2	GuiSOS	SizeN	completado	Retorna error
2	2.1	GuiSOS	ButtonGame, NuevoTableroSimple	Completado	Selecciona el modo juego simple
2	2.1	GuiSOS	ButtonGame, NuevoTableroGeneral	completado	Selecciona el modo de juego general
3	3.1	GuiSOS	init_Tablero	completado	Realiza esta acción después de apretar en New Game y luego Start
4	4.1	GuiSOS	ButtonGame, SOS, NuevoTableroSimple, Game, fullScreen, bluePlayer, redPlayer	completado	Si un jugador aun no forma SOS y el tablero sigue vacío, cambia de jugador
4	4.2	GuiSOS	init_Tablero,Game, bluePlayer, redPlayer	completado	No se selecciona S u O entonces no pasa nada hasta seleccionar S u O

5	5.1	GuiSOS	ButtonGame, SOS, NuevoTableroSimple, Game, setGameFinal, bluePlayer, redPlayer	completado	SOS lee si se formo un SOS, dentro de Game condicional si se forma SOS entonces hay un ganador
5	5.2	GuiSOS	ButtonGame, SOS, NuevoTableroSimple, Game, fullScreen, setGameFinal, bluePlayer, redPlayer	completado	SOS lee si se formó un SOS, dentro de Game condicional si no se forma SOS y el tablero esta lleno (fullScreen) empate
6	6.1	GuiSOS	ButtonGame, SOS, NuevoTableroSimple, Game, bluePlayer, redPlayer,fullScreen	completado	Si todavía queda espacios vacíos(fullScreen), y cuando se forma un SOS (SOS) el jugador de turno realice un segundo movimiento (Game) hasta dejar de realizar SOS
6	6.2	GuiSOS	ButtonGame, SOS, NuevoTableroSimple, Game, bluePlayer, redPlayer,fullScreen	completado	Si todavía queda espacios vacíos(fullScreen), y si no realiza SOS (SOS) entonces el otro jugador sigue realizando su turno (Game)
7	7.1	GuiSOS	ButtonGame, SOS, NuevoTableroSimple, Game, bluePlayer, redPlayer,fullScreen	completado	Si el tablero se encuentra lleno (fullScreen) se pasa a hacer la comparación de los auxiliares acumulados (número dado por SOS a lo largo del juego) en Game y se escoge al ganador con más puntos
7	7.2	GuiSOS	ButtonGame, SOS, NuevoTableroSimple, Game, bluePlayer, redPlayer,fullScreen	completado	Si el tablero se encuentra lleno (fullScreen) se pasa a hacer la comparación de los auxiliares acumulados (número dado por SOS a lo largo del juego) en Game y si son iguales resulta en un empate

4. 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.

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	Realizar un nuevo juego del tamaño de tablero y del modo de juego elegidos
4	Hacer un movimiento en un juego simple
5	Un juego simple a terminado
6	Hacer un movimiento en un juego general
7	Un juego general a terminado

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

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	1.1	TestSizeBoard	setup, CorrectN	Una prueba para dato correcto. Entrada 3 (n) y salida 3 (n)
1	1.2	TestSizeBoard	setup, WrongN1	Una prueba para dato incorrecto. Entrada 2 (n) y salida 3 (n+1)
2	2.1	TestGameMode	setup, teardown, SOS_GameModeSimpleGame, NuevoTableroSimple, GetModeGame	Escojo un NuevoTableroSimple Tengo como salida esperada "Simple Game"
2	2.1	TestGameMode	setup, teardown, SOS_GameModeGeneralGame, NuevoTableroGeneral, GetModeGame	Escojo un NuevoTableroGeneral Tengo como salida esperada "General Game"
3	3.1	TestNewGame	setup,teardown,ButtonNewGame, init_Tablero, getIniGame	Como se llama a la función de inicializar tablero, posteriormente iniGame cambia a TRUE (Lo que se espera)

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	Nombre Clase (s) del código de prueba	Entrada de caso de prueba	Salida esperada	Notas
------------------------------------	-------	---------------------------------------	---------------------------	-----------------	-------

4	4.1	TestMoveSimpleGame	Se realiza de manera manual el ingreso de datos SOS para un n=3	Red turn	Se evalúa a quien pertenece el turno actual
5	5.1	TestFinishSimpleGame	Se realiza de manera manual el ingreso de datos SOS para un n=3	BLUE IS THE WINNER	Con la condición de que gane azul
5	5.1	TestFinishSimpleGame	Se realiza de manera manual el ingreso de datos SOS para un n=3	RED IS THE WINNER	Con la condición de que gane rojo
5	5.2	TestFinishSimpleGame	Se realiza de manera manual el ingreso de datos SOS para un n=3	DRAW	Se rellena todo el tablero con O para simular un juego con empate
6	6.1	TestMoveGeneralGame	Se realiza de manera manual el ingreso de datos SOS	Blue turn	Se evalúa a quien pertenece el turno actual
7	7.1	TestFinishGeneralGame	no	no	no

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)

Para el desarrollo de este proyecto no se ha utilizado a la jerarquía de clases, ya que todo se desarrolla dentro de la clase GuiSOS. Dicho esto, lo que se ha tratado de realizar es separar de manera independiente las funciones con respecto a un juego simple y un juego general. Para esto, se ha utilizado opciones de selección para diferenciarlos. Sin embargo, aquellos modos vuelven a encontrarse en modo de condicionales en el método Game. El cual dependiendo de qué modo de juego se ha seleccionado en el GUI, evaluará dichas formas de jugar.