

## Reporte Sprint #4

Implementa todas las características que permiten a un jugador (humano o computadora) jugar un juego SOS simple o general contra otro jugador (humano o computadora). Las características mínimas incluyen elegir humano o computadora para jugadores rojos y/o azules, 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 componente informático debe ser capaz de jugar juegos simples y generales completos. Se te anima a considerar estrategias básicas para ganar juegos simples o generales. No se requiere un juego óptimo. El siguiente es un diseño de GUI de muestra. Debes usar una jerarquía de clases para lidiar con los requisitos del oponente de la computadora. Si tu código actual aún no ha considerado la jerarquía de clases, es hora de refactorizar tu código.

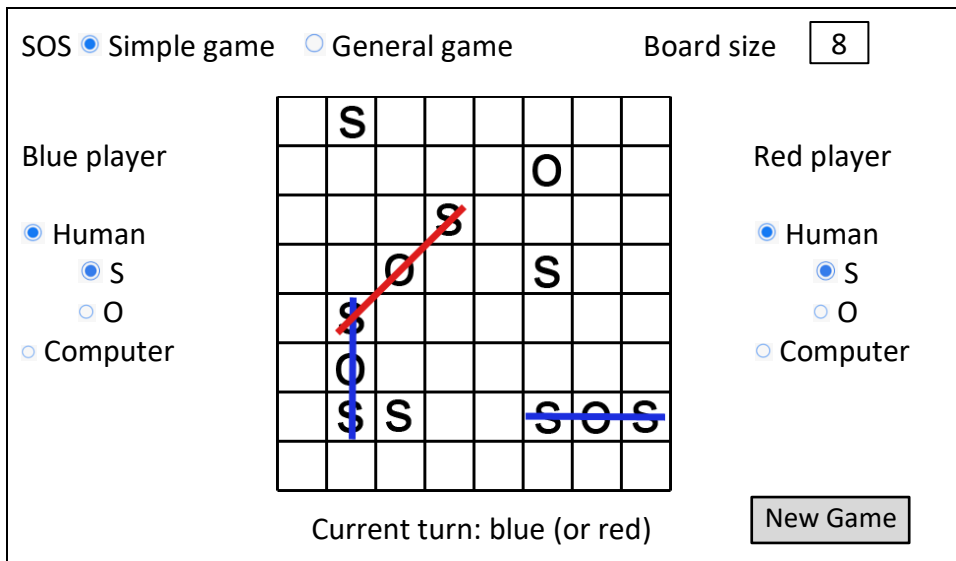


Figura 1. Diseño de GUI de muestra del programa de trabajo para Sprint 3

### Puntos totales: 20

#### 1. Demostración (10 puntos)

Envía un video de no más de cinco minutos, que demuestre claramente que implementaste las funciones del oponente de la computadora y escribiste algunas pruebas de unidad automatizadas.

- 1) Un juego completamente simple donde el jugador azul es un humano, el jugador rojo es la computadora y hay un ganador.
- 2) Un juego general completo donde el jugador azul es la computadora, el jugador rojo es un humano y hay un ganador.
- 3) Un juego simple completo donde ambos lados son jugados por la computadora
- 4) Un juego general completo donde ambos lados son jugados por la computadora.
- 5) Algunas pruebas unitarias automatizadas para el oponente de la computadora.

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

## 2. Historias de usuario para los requisitos del oponente de la computadora (1 punto)

**Plantilla de historia de usuario:** Como <rol>, quiero <objetivo> [tal que <beneficio>]

ID	Nombre de historia de usuario	Descripción de historia de usuario	Prioridad	Esfuerzo estimado (horas)
8	Computadora realiza un movimiento	La computadora realiza un movimiento cuando el usuario primer lugar se realiza clic en los botones S u O y luego en alguna casilla del tablero. Escoger entre S u O no es importante, pero se debe marcar una de ellas, ya que marcando cualquiera de las dos opciones y al realizar el clic en el tablero se escoge una casilla aleatoria y pone un S u O aleatorio.	5	7
9	Elegir tipo de jugador	Se selecciona, antes de empezar new game el tipo de jugador a seleccionar, humano o computador. Para así poder jugar ya sea humano vs humano, humano vs computador o computador vs computador.	4	5
10	Realizar un juego humano vs computador	Cuando se realiza un juego humano vs computador o computador vs humano. El jugador “humano” puede seleccionar el S u O y la casilla en la cual colocar dicha letra, mientras que el jugador “computador” realiza un movimiento aleatorio apoyado por el usuario es decir realizando el procedimiento del ID 8.	2	5
11	Realizar un juego computador vs computador	Cuando se realiza un juego computador vs computados se realiza un juego aleatorio apoyado con el ID8.	2	5

## 3. Criterio de aceptación (AC) para los requisitos del oponente de la computadora (1 punto)

ID y nombre de la historia de usuario	AC ID	Descripción del criterio de aceptación	Estado (completado, por hacer, en progreso)
8. Computadora realiza un movimiento	8.1	AC 8.1 <Movimiento valido de computadora> Given se realiza un movimiento computadora When se selecciona una opción S u O del tablero Y se escoge una casilla del tablero Then la computadora juega de modo aleatorio	8
	8.2	AC 8.2 <Movimiento invalido de computadora> Given se desea realiza un movimiento computadora When cuando no se elige una opción S u O del tablero Y se da clic en una casilla del tablero Then no se realiza movimiento hasta que haga el movimiento valido	5
9. Elegir tipo de jugador	9.1	AC 9.1 <Elección entre computador o humano> Given que se desea escoger un tipo de jugador When se realiza clic en la sección human o jugador del color blue o red Y seleccionado una cantidad de tablero n Y el modo de juego Y se realiza clic en el botón star Then se establece el tipo de jugador elegido	5

10. Realizar un juego humano vs computador	10.1	AC 10.1 <Se realiza un juego humano vs computador> Given se escoge estos jugadores When es turno de jugador humano, se realiza un movimiento humano And si es turno de jugador computadora, se realiza un movimiento computadora Then se realiza el juego con normalidad según el tipo de juego	3
11. Realizar un juego computador vs computador	11.1	AC 11.1 <Se realiza un juego computador vs computador> Given se escoge estos jugadores When es turno de jugador computador 1, se realiza un movimiento computadora And si es turno de jugador computador 2, se realiza un movimiento computadora Then se realiza el juego con normalidad según el tipo de juego	3

#### 4. Resumen de todo el código fuente (1 punto)

Nombre del archivo de código fuente	Código de producción o de prueba	# líneas de código
Total		

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

#### 5. Código de producción vs Nuevas historias de usuario/Criterios de aceptación (2 puntos)

Resume cómo se implementa cada una de las nuevas historias de usuario/criterios de aceptación en tu código de producción (nombre de clase y nombre de método, etc.)

ID y nombre de la historia de usuario	AC ID	Nombre(s) clases	Nombres(s) métodos	Estado (completo o no)	Notas (opcional)
8	8.1	GuiSOS	Game	Completado	Se necesita apoyo humano. Pero de manera interna se escoge el S u O aleatoriamente y una casilla aleatoriamente.
8	8.2	GuiSOS,	Game	Completado	Es necesario escoger un S u O para realizar un movimiento computador, esta elección en el tablero no afecta a la aleatoriedad.
9	9.1	GuiSOS, ClassPlayer	init_Tablero, cambiarturno, getisHuman, getPlayer	Completado	Se designa true para humano y false para computadora
10	10.1	GuiSOS, ClassPlayer, ClassScore, Cl	init_Tablero, cambiarturno, getCurrentPlayer,	Completado	El juego cambia de turno independientemente

		assGeneralGame, ClassSimpleGame	Game, getisHuman,getPlayer,		e de que tipo de jugador se haya elegido
11	11.1	GuiSOS, ClassPlayer,ClassScore,Cl assGeneralGame, ClassSimpleGame	init_Tablero, cambiarturno, getCurrentPlayer, Game, getisHuman,getPlayer	Completado	El juego cambia de turno independientement e de que tipo de jugador se haya elegido

## 6. Pruebas vs Nuevas historias de usuario/Criterio de aceptación (2 puntos)

Resumen cómo tu código de prueba (nombre de clase y nombre de método) o las pruebas realizadas manualmente prueban cada una de las nuevas historias de usuario/criterios de aceptación.

Nombre y ID de la historia usuario	AC ID	Nombre Clase (s) del código de prueba	Entrada de caso de prueba	Salida esperada	Notas
8	8.1	TestComputerMoveG eneralGame	Se realiza de manera manual ingreso de datos SOS para un n=3 player1 = ("BLUE", false); player2 = ("RED", false);	BLUE	Si bien no se puede introducir una variable aleatoria para probar, se realiza la simulación del juego entre computadoras en un juego general
8	8.1	TestComputerMoveS impleGame	Se realiza de manera manual ingreso de datos SOS para un n=3 player1 = ("BLUE", false); player2 = ("RED", false);	RED	Si bien no se puede introducir una variable aleatoria para probar, se realiza la simulación del juego entre computadoras en un juego simple
9	9.1	TestWhoIam	Player1=new ClassPlayer("BLUE",f alse) Entrada: player1.getisHuman()	false	Se identifica si el jugador escogido es una computadora
9	9.1	TestWhoIam	Player2=new ClassPlayer("RED",tru e) Entrada: player2.getisHuman()	true	Se identifica si el jugador escogido es un humano
9	9.1	TestWhoIam	Player1=new ClassPlayer("BLUE",f alse) Entrada: player1.getPlayer ()	BLUE	Se identifica si el jugador escogido es blue
9	9.1	TestWhoIam	Player2=new ClassPlayer("RED",tru e) Entrada: player2.getPlayer ()	RED	Se identifica si el jugador escogido es red
10	10.1	TestHumanAndComp FinishGeneralGame,	Ingreso de datos SOS modo general para un n=3 player1 = ("BLUE", false);	BLUE IS THE WINNER, RED IS THE WINNER, DRAW	Se simula un juego general para humano y computador como jugadores. Donde debe ganar el jugador azul, el rojo y otro test en queda en empate

			player2 = ("RED", true);		
10	10.1	TestHumanAndCompFinishSimpleGame,	Ingreso de datos SOS modo simple para un n=3 player1 = ("BLUE", false); player2 = ("RED", true);	BLUE IS THE WINNER, RED IS THE WINNER, DRAW	Se simula un juego simple para humano y computador como jugadores. Donde debe ganar el jugador azul, el rojo y otro test en queda en empate
10	10.1	TestHumanAndCompMoveGeneralGame	Ingreso de datos SOS modo general para un n=3 player1 = ("BLUE", false); player2 = ("RED", true);	BLUE	Se simula los movimientos para los jugadores humano y computador en SOS general. Donde el turno siguiente es el azul
10	10.1	TestHumanAndCompMoveSimpleGame	Se realiza de manera manual ingreso de datos SOS para un n=3 player1 = ("BLUE", false); player2 = ("RED", true);	RED	Se simula los movimientos para los jugadores humano y computador en SOS simple. Donde el turno siguiente es el rojo
11	11.1	TestComputerFinishGeneralGame	Se realiza de manera manual ingreso de datos SOS para un n=3 player1 = ("BLUE", false); player2 = ("RED", false);	BLUE IS THE WINNER, RED IS THE WINNER, DRAW	Se simula un juego SOS en modo general de computador vs computador donde se dan los resultados de azul gana, rojo gana y empate
11	11.1	TestComputerFinishSimpleGame	Se realiza de manera manual ingreso de datos SOS para un n=3 player1 = ("BLUE", false); player2 = ("RED", false);	BLUE IS THE WINNER, RED IS THE WINNER, DRAW	Se simula un juego SOS en modo simple de computador vs computador donde se dan los resultados de azul gana, rojo gana y empate

#### 6.1 Pruebas automatizadas que corresponden directamente a algunos criterios de aceptación.

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)
9	9.1	TestWhoIam, ClassPlayer	Choose_Human,	Player1=new ClassPlayer("BLUE",false) Entrada: player1.getisHuman() Salida: false
9	9.1	TestWhoIam, ClassPlayer	Choose_Computer	Player2=new ClassPlayer("RED",true) Entrada: player2.getisHuman() Salida:true
9	9.1	TestWhoIam, ClassPlayer	Choose_Red	Player1=new ClassPlayer("BLUE",false) Entrada: player1.getPlayer () Salida:"BLUE"
9	9.1	TestWhoIam, ClassPlayer	Choose_Blue	Player1=new ClassPlayer("RED",false) Entrada: player2.getPlayer () Salida:"RED"

#### 6.2 Pruebas manuales que corresponden directamente a algunos criterios de aceptación.

Nombre y ID de la historia usuario	AC ID	Nombre Clase (s) del código de prueba	Entrada de caso de prueba	Salida esperada	Notas
8	8.1	TestComputerMoveGeneralGame	Se realiza de manera manual ingreso de datos SOS para un n=3 player1 = ("BLUE", false); player2 = ("RED", false);	BLUE	Si bien no se puede introducir una variable aleatoria para probar, se realiza la simulación del juego entre computadoras en un juego general
8	8.1	TestComputerMoveSimpleGame	Se realiza de manera manual ingreso de datos SOS para un n=3 player1 = ("BLUE", false); player2 = ("RED", false);	RED	Si bien no se puede introducir una variable aleatoria para probar, se realiza la simulación del juego entre computadoras en un juego simple
10	10.1	TestHumanAndCompFinishGeneralGame,	Ingreso de datos SOS modo general para un n=3 player1 = ("BLUE", false); player2 = ("RED", true);	BLUE IS THE WINNER, RED IS THE WINNER, DRAW	Se simula un juego general para humano y computador como jugadores. Donde debe ganar el jugador azul, el rojo y otro test en queda en empate
10	10.1	TestHumanAndCompFinishSimpleGame,	Ingreso de datos SOS modo simple para un n=3 player1 = ("BLUE", false); player2 = ("RED", true);	BLUE IS THE WINNER, RED IS THE WINNER, DRAW	Se simula un juego simple para humano y computador como jugadores. Donde debe ganar el jugador azul, el rojo y otro test en queda en empate
10	10.1	TestHumanAndCompMoveGeneralGame	Ingreso de datos SOS modo general para un n=3 player1 = ("BLUE", false); player2 = ("RED", true);	BLUE	Se simula los movimientos para los jugadores humano y computador en SOS general. Donde el turno siguiente es el azul
10	10.1	TestHumanAndCompMoveSimpleGame	Se realiza de manera manual ingreso de datos SOS para un n=3 player1 = ("BLUE", false); player2 = ("RED", true);	RED	Se simula los movimientos para los jugadores humano y computador en SOS simple. Donde el turno siguiente es el rojo
11	11.1	TestComputerFinishGeneralGame	Se realiza de manera manual ingreso de datos SOS para un n=3 player1 = ("BLUE", false); player2 = ("RED", false);	BLUE IS THE WINNER, RED IS THE WINNER, DRAW	Se simula un juego SOS en modo general de computador vs computador donde se dan los resultados de azul gana, rojo gana y empate
11	11.1	TestComputerFinishSimpleGame	Se realiza de manera manual ingreso de datos SOS para un n=3 player1 = ("BLUE", false); player2 = ("RED", false);	BLUE IS THE WINNER, RED IS THE WINNER, DRAW	Se simula un juego SOS en modo simple de computador vs computador donde se dan los resultados de azul gana, rojo gana y empate

### 6.3 Otras pruebas automatizadas o manuales que no correspondan a los criterios de aceptación

Número	AC ID	Entrada de prueba	Salida esperada	Nombre de clase del código de prueba	Nombre del método del código de prueba

7. ¿Presenta el diagrama de clases de tu código de producción y describe cómo la jerarquía de clases en su diseño trata con los requisitos del oponente de la computadora (3 puntos)?