Reporte del Sprint #5

Las principales tareas de esta asignación son:

(1) Agrega la función de grabar (record) un juego en un archivo de texto. Se requiere la historia de usuario y los criterios de aceptación tanto de grabación como de reproducción
(2) Realización de un ejercicio de revisión de código.
(3) Resumir las lecciones aprendidas del Sprint 0 al Sprint 5.

El siguiente es un diseño de GUI de muestra del producto final, donde "Replay" es opcional.

El trabajo es de caracter individual.

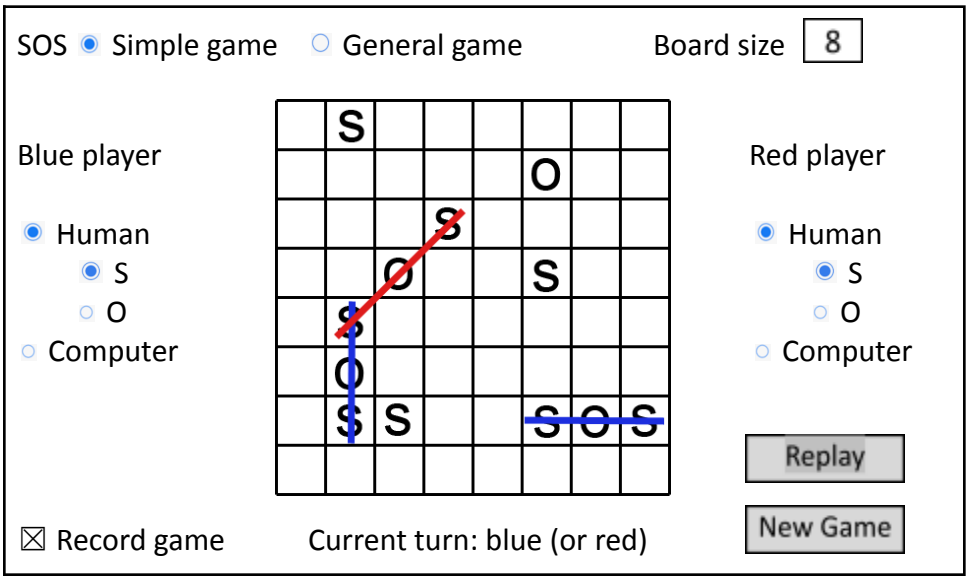


Figura 1. Sample GUI layout of the final product Diseño de GUI del producto final

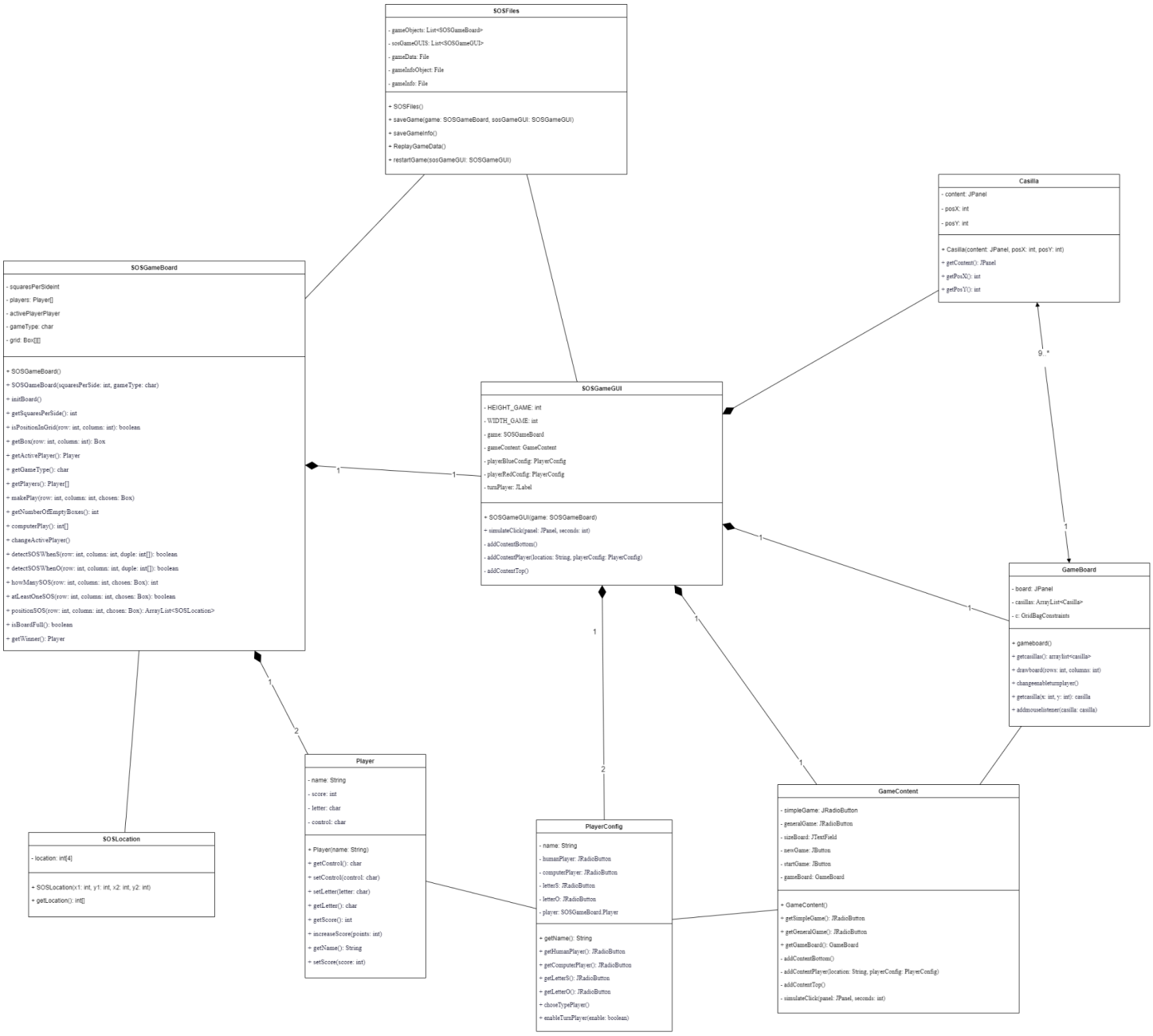
Puntos totales

1. Demostración (10 puntos)

Envía un video de no más de 15 minutos, demostrando claramente que has implementado todas las funciones en la siguiente tabla. En el video, debes explicar lo que se está demostrando. **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.**

	Feature
1	Se graba un juego simple completo de dos jugadores humanos.
2	Se graba un juego general completo de dos jugadores humanos
3	Se graba un juego simple completo de jugadores humano-computadora
4	Se graba un juego general completo de jugadores humano-computadora
5	Se graba un juego simple completo de jugadores computadora-computadora
6	Se graba un juego general completo de jugadores computadora-computadora

Si has implementado la función de "replay" para obtener crédito adicional, debes incluir tu demostración en el video.



2. Historias de usuario y criterios de aceptación para los requisitos para los requerimientos Record/Replay (1 punto)

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

Agrega o elimina filas si es necesario

ID	Nombre de historia de usuario	Descripción de historia de usuario	Prioridad	Esfuerzo estimado (horas)
9	Guardar partida en transcurso o finalizado	Como usuario quiero poder guardar la partida tal que luego pueda retomar el juego o ver cómo finalizó el juego.	4	3
10	Iniciar juego guardado	Como usuario quiero retomar o ver los juegos guardados.	4	4

ID y nombre de la historia de usuario	AC ID	Descripción del criterio de aceptación	Estado (completado, por hacer, en progreso)
Historia 9	20.1	AC 20.1 <Juego guardado antes de finalizar> **Dado** iniciado el juego **Cuando** el usuario grabe el juego en transcurso **Entonces** la partida se guardará en una carpeta para luego retomar el juego.	completado
	20.2	AC 20.2 <Juego guardado después de finalizar> **Dado** finalizado el juego **Cuando** el usuario grabe el juego **Entonces** los datos de la partida se guardarán en una carpeta.	completado
Historia 10	21.1	AC 21.1 <Retomar juego guardado> **Dado** seleccionado un juego guardado **Cuando** se inicie el juego a retomar **Entonces** se mostrará el juego en el estado que se guardó(finalizado o en transcurso).	completado

3. Revisión de código (4 puntos)

Aplica la revisión del código fuente a una o dos de las clases más importantes (y a otras clases si el tiempo te permite) e informa de los resultados. Además de buscar errores, la revisión debe verificar: (1) si todo el proyecto ha seguido el estándar de codificación de manera consistente, (2) si el proyecto ha seguido los principios de diseño presentados en clase y (3) si hay olores de código que indican la necesidad de refactorización.

Las siguientes listas de verificación proporcionan pautas básicas. Puedes agregar nuevos elementos a cada una de las listas de verificación. Asegúrate de que tus respuestas sean el resultado del ejercicio de revisión del código. Si no hay hallazgos para una entrada, debes proporcionar una explicación. Por ejemplo, si tu respuesta a "¿Se violan las convenciones de nomenclatura?" es no, debes describir una convención de nomenclatura y presentar un ejemplo. No recibirás puntaje por si tus respuestas son simplemente sí o no sin información adicional.
Clases que han sido revisadas: SOSGameBoard, SOSGameGUI
Fecha/hora de duración del ejercicio de revisión del código: 3h

Checklist	Items Checklist	Conclusiones	
Estándares de codificación	Convenciones de nombres	Camel case, nombres descriptivos, snake case	
	Convención de ordenación de argumentos de método	Orden basado en propósito	
	Comentarios significativos y válidos.	no	
	Estilo consistente de bloques de código	identacion, uso de llaves, longitud de línea	
	Indentación consistente	Alineación vertical, 4 espacios para cada nivel	
	…		
Principio de diseño	Clase o método no bien modularizado	addMouseListener(Casilla casilla)	
	Visibilidad adecuada de cada variable, método y clase.	variables privados, clases anidadas y privados, métodos públicos y privados	
	Alguna clase con pobre abstracción	SOSLocation	
	Diseño por contrato (pre/postcondiciones)		
	¿Se viola el Principio Abierto-Cerrado?	No, dado que para cada sprint se han agregado nuevos métodos pero no modificado el código fuente y no se ignoran los principios de herencia y polimorfismo.	
	¿Se viola el Principio de Responsabilidad Única [1]?	No, las clases no tienen múltiples funcionalidades por lo que no se han modificado frecuentemente.	
Smells código	Números mágicos	Las variables del tipo de juego ‘S“ y ‘G“ no explican bien la referencia de simple y general game.	
	Variable global /clase innecesaria	no	
	Código duplicado	no	
	Métodos largos	addMouseListener(Casilla casilla) dado que se muestra el ganador y la posibilidad de los movimientos	
	Larga lista de parámetros	no	
	Expresión demasiado compleja	no	
	Switch o if-then-else que necesita ser reemplazado con polimorfismo	addMouseListener(Casilla casilla) debería presentar clases de los tipos de juego para facilitar el uso.	
	Nombre de método o variable cuya intención no está clara	simulateClick(JPanel casilla, int seconds)	
	¿Algún método similar en otras clases?	no	
	…		
Errores	**Fragmento de código con errores**	**¿Cuál es el error?**	**¿Por qué es un error?**

4. Resumen de todo el código (1 points)

Nombre del archivo de código fuente	Código de producción o prueba?	# lineas de código
SOSFIles	producción	217
SOSGameBoard	producción	221
SOSGameGUI	producción	750
	Total de líneas de código	

No recibirás puntaje por esta tarea a menos que envíes tu código fuente completo.

[1] Revisa: Violation solution for single responsibility principle

5. Resume las lecciones aprendidas de todo el proyecto respondiendo las siguientes preguntas desde la perspectiva de los procesos de desarrollo, codificación, diseño, refactorización y prueba (**4 puntos**):

- ¿Qué ganaste personalmente con el proyecto?

  Cómo planificar las funcionalidades que voy a necesitar antes de codificar, luego de tener la codificación de las funcionalidades ver si es necesario refactorizar de este modo tener el código del software mejor estructurado y para luego pasar a las pruebas donde también veré si falta refactorizar el código nuevamente o agregar implementaciones necesarias.

- ¿Qué hace bien tu proyecto y qué podría hacer mejor tu proyecto?

  Presenta todas las funcionalidades que se requerían pero falta refactorizar algunas clases dado que hay métodos donde se hacen funcionalidades donde no deberían estar en la clase donde se implementa.

- ¿Cómo podrías mejorar tu proceso de desarrollo si desarrollas un juego similar desde cero?

  Dado que no tenía mucho conocimiento de la librería Swing y de las pruebas, podría hacer un mejor uso de estas, y clasificar mejor las clases y la estructura de cómo se implementan estas mediante la refactorización y patrones de diseño.

Requisito mínimo para (5): Una página completa a espacio simple, tamaño de fuente no mayor a 12 puntos.