Informe Sprint #1

Instrucciones

Lee atentamente las instrucciones. Todos los miembros del equipo deben discutir las instrucciones juntos para asegurarse de que todos estén en la misma página.

Objetivos

1. Especifica los requisitos (es decir, todas las historias de usuario y los criterios de aceptación) del software objetivo que permite que un jugador humano juegue contra un oponente humano. Las características mínimas **incluyen elegir el turno del jugador, configurar un juego, colocar una pieza, mover una pieza, volar una pieza, quitar una pieza del oponente, determinar si el juego ha terminado.**
2. Implemente las funciones primitivas, incluido el objeto del tablero y la visualización, la colocación de piezas para ambos jugadores. La interfaz también debe proporcionar información sobre las piezas y el turno de cada jugador.
3. Cada equipo debe reunirse al menos una vez a la semana. Una reunión puede servir para múltiples propósitos en el proceso Scrum. Cada miembro debe contribuir a la codificación.

Entregables y Política de Calificación

Comprima el informe del proyecto, el video de demostración y todo el código fuente en un archivo .zip antes de enviarlo. Por favor, no cargues archivos rar y no envíes enlaces.

1. Informe del proyecto

   El informe del proyecto debe incluir las siguientes secciones:

   I. Historias de usuarios usando la plantilla discutida en clase:
   Proporciona una lista completa de historias de usuarios y esfuerzos estimados para el software objetivo que permite que un jugador humano juegue contra un jugador humano. Todo el equipo debe realizar el enfoque de poker de planificación para la estimación del esfuerzo. Las historias de usuario deben cubrir las siguientes características por separado:
   - Elegir el turno del jugador
   - Configurar un juego
   - Colocar una pieza
   - Mover una pieza
   - Volar [1] una pieza
   - Quitar la pieza de un oponente
   - Determinar si el juego ha terminado.

   II. Criterios de aceptación utilizando la plantilla Dado-Cuándo-Entonces.
   Proporciona criterios de aceptación completos para cada una de las historias de usuario anteriores.

   III. Tareas de implementación
   Describa el código de producción, el código de prueba automatizado o el caso de prueba manual para cada historia de usuario y el criterio de aceptación relacionado con la implementación de las funciones primitivas, es decir, objeto del tablero, visualización y colocación de piezas para ambos jugadores. Para cada criterio de aceptación de cada historia de usuario para las funciones

[1] fly

primitivas, debe implementar al menos una prueba (ya sea código de prueba o caso de prueba manual).

IV. Resumen del código fuente
Proporciona un archivo zip de todo el código fuente y resume la contribución de cada miembro. No recibirás puntaje si no se envía el código fuente. Asegúrate de que el informe de tu proyecto sea coherente con el código fuente.

V. Actas de todas las reuniones, incluidas, entre otras, reuniones de planificación de proyectos/sprints, reuniones de trabajo, backlog grooming, reuniones retrospectivas y sesiones de programación (o desarrollo) en pares.

VI. Una tabla de calificaciones de amigos. Los miembros individuales pueden enviar sus calificaciones de compañeros por correo electrónico al profesor del curso.

Cada equipo solo necesita presentar un informe. Para que un miembro individual reciba puntaje por esta parte del proyecto, el informe del proyecto del equipo debe incluir evidencia explícita de tu contribución.

2. Demostración

Entregar un vídeo 10- 15 minutos, claramente demostrando que:

a) Tu proyecto ha implementado el software de trabajo para las funciones primitivas, es decir, el objeto y la visualización del tablero, y la ubicación de las piezas para ambos jugadores.

Característica	Puntuación
Visualización del tablero	
Colocación de piezas válidas e inválidas	

b) Para cada criterio de aceptación de cada historia de usuario para las funciones primitivas, tu proyecto implementó un método de prueba automatizado o realizó una prueba de aceptación manualmente.

Característica	Puntuación
Intersecciones/puntos válidos e inválidos	
Colocación de piezas válidas e inválidas	

c) Tu proyecto tiene algunas características o mejoras únicas (opcional).

La calificación de la demostración se basa en la finalización de las funciones requeridas y la presentación general utilizando la siguiente rúbrica de evaluación:

	Pobre	Justo	Bueno	Muy bueno	Excelente
¿Estaba la demostración lógicamente organizada?					
¿Se formularon los puntos de forma clara y concisa?					

Informe Sprint #1

Plantilla del informe

Nombre del equipo: Los galácticos

Información dada por el equipo del Proyecto		A ser usado por el profesor	
NombreEstudiante	**Contribuciones específicas para este Sprint**	**Puntaje Equipo**	**Puntaje Individual**
Ruben Anthony Ricapa Corrales	Contribución con las historias de usuario y criterios de aceptación Implementación de la clase tabla, ficha y game		
Juan José Camarena Zamalloa	Implementación de los tests y contribución a implementar la historia de usuario a código. Corrección de lógica.		
Gerardo Michael Carpio Usquiano	Contribución del diseño de la tabla, piezas y turnos de la ficha, corrección de errores y revisión de estilo pep8		

Un estudiante sin ninguna participación no recibirá puntaje.

I. Historias de usuarios

ID	Nombre Historia de Usuario	Descripción Historia Usuario	Prioridad	Esfuerzo estimado(horas)	Esfuerzo real (si se completó)	Status (completado, por Hacer, en Progreso)	Nombre desarrollador
1	Colocar una pieza	Como jugador necesito colocar una de mis piezas en el tablero para poder hacer un movimiento	3	2 horas	4 horas	completado	Ruben
2	Mover una pieza	Como jugador necesito mover una pieza hacia otra posición para formar un molino	4	3 horas		en Progreso	Juan
3	Volar una pieza	Como jugador necesito volar una pieza para formar un molino	6	0.5 horas		por Hacer	Juan
4	Elegir turno del jugador	Como jugador necesito un turno para poder jugar.	2	1 hora	2 horas	completado	Michael
5	Configurar un juego	Como jugador necesito un tablero vacío para poder comenzar el juego.	1	4 hora	4 horas	completado	Ruben
6	Determinar si el juego ha terminado	Como jugador necesito saber si el juego ha terminado después de cada movimiento	7	0.5 horas		por Hacer	Michael
7	Quitar pieza de un oponente	Como jugador necesito quitar una pieza de mi	5	3 horas		en Progreso	Ruben

		oponente cuando forme un molino para avanzar en el juego					

II. Criterios de aceptación (AC)

vID Historia de Usuario y Nombre	AC ID	Descripción de los criterios de aceptación	Status (completado, por Hacer, en Progreso)	Nombre desarrollador
1 Colocar una pieza	1.1	AC 1.1 La casilla está ocupada Dado el comienzo de un juego de 2 personas Cuando quiera colocar una pieza en una casilla Y esta casilla se encuentre ocupada Entonces el movimiento será invalido	completado	Ruben
	1.2	AC 1.2 La casilla está vacía Dado el comienzo de un juego de 2 personas Cuando quiera colocar una pieza en una casilla Y esta se encuentre vacía Entonces el movimiento será válido	completado	Ruben
2 Mover una pieza	2.1	AC 2.1 La casilla está ocupada Dado un juego de 2 personas después de haber colocado todas las piezas Cuando quiero mover una pieza a una casilla Y la casilla se encuentra ocupada Entonces el movimiento será invalidado Y se le volverá a preguntar en qué casilla desea colocar la pieza	en Progreso	Juan
	2.2	AC 2.2 La casilla está vacía pero no es adyacente Dado un juego de 2 personas después de haber colocado todas las piezas Cuando quiero mover una pieza a una casilla Y la casilla se encuentra vacía Y no es adyacente Entonces el movimiento será invalidado Y se le volverá a preguntar en qué casilla desea colocar la pieza	en Progreso	Juan
	2.3	AC 2.3 La casilla está vacía y es adyacente, pero no se forma un molino Dado un juego de 2 personas después de haber colocado todas las piezas Cuando quiero mover una pieza a una casilla Y la casilla se encuentra vacía Y es adyacente Y no se forma un molino Entonces el movimiento se confirmará Y se mueve la pieza al lugar Y se cambia de turno	en Progreso	Juan
3 Volar una pieza	3.1	AC 3.1 La casilla está ocupada Dado un juego de 2 personas y a un jugador le quedan solo 3 piezas Cuando quiero volar una pieza a una casilla Y la casilla se encuentra ocupada Entonces el movimiento será invalidado Y se le volverá a preguntar en qué casilla desea colocar la pieza	por Hacer	Juan
	3.2	AC 2.2 La casilla está vacía y es adyacente Dado un juego de 2 personas y a un jugador le quedan solo 3 piezas Cuando quiero volar una pieza a una casilla Y la casilla se encuentra vacía	por Hacer	Juan

		Entonces el movimiento será válido Y se mueve la pieza al lugar		
4 Elegir turno del jugador	4. 1	AC 4.1 Turno del jugador “blanco” Cuando el turno del jugador “negro” finalice Y no haya ganado Entonces se da el turno al jugador “blanco” Y si la posición es válida, se realiza la jugada	completado	Michael
	4.2	AC 4.2 Turno del jugador “negro” Cuando el turno del jugador “blanco” finalice Y no haya ganado Entonces se da el turno al jugador “negro” Y si la posición es válida, se realiza la jugada	completado	Michael
5 Configurar un juego	5. 1	AC 5.1 Tablero vacío Dado un juego de 2 personas Cuando el juego comience Entonces el tablero debe estar vacío Y el usuario empezará a llenarlo	completado	Ruben
	5.2	AC 5.2 Colocar una pieza fuera del tablero Cuando el juego comienza con el tablero vacío Y el usuario coloque una pieza fuera del tablero Entonces el movimiento no hará efecto	completado	Ruben
6 Determinar si el juego ha terminado	6. 1	AC 6.1 Victoria del jugador “negro”, jugador “blanco” sin fichas suficientes Dado el movimiento del jugador “negro” Y realice una jugada válida Cuando se realice la jugada Y el jugador “blanco” se quede con dos fichas Entonces ganará el jugador “negro” Y el juego termina	por Hacer	Michael
	6.2	AC 6.2 Victoria del jugador “blanco”, jugador “negro” sin fichas suficientes Dado el movimiento del jugador “blanco” Y realice una jugada válida Cuando se realice la jugada Y el jugador “negro” se quede con dos fichas Entonces ganará el jugador “blanco” Y el juego termina	por Hacer	Michael
	6.3	AC 6.3 Victoria del jugador “negro”, jugador “blanco” sin movimientos Dado un movimiento válido del jugador “negro” Cuando sea el turno del jugador “blanco” Y no tenga jugadas válidas Entonces el jugador “negro” gana Y el juego termina	por Hacer	Michael
	6.4	AC 6.4 Victoria del jugador “blanco”, jugador “negro” sin movimientos Dado un movimiento válido del jugador “blanco” Cuando sea el turno del jugador “negro” Y no tenga jugadas válidas Entonces el jugador “blanco” gana Y el juego termina	por Hacer	Michael
7. Quitar pieza de un oponente	7.1	7.1 Se forma un molino Dado un juego de 2 jugadores Cuando un jugador forme un molino Entonces el jugador podrá quitar una pieza	en Progreso	Ruben
	7.2	AC 7.2 La pieza es nuestra Dado un juego donde formó un molino Cuando quiera quitar una pieza del tablero Y esta pieza sea mía Entonces el juego no permitiría quitarla Y no cambia de turno hasta que quite la pieza	por Hacer	Ruben

	7.3	AC 7.3 La pieza es del rival Dado un juego donde formó un molino Cuando quiera quitar una pieza del tablero Y esta pieza sea del rival Y no forme parte de un molino Entonces la pieza del rival desaparece del tablero Y cambia el turno	en Progreso	Ruben

III. Tareas de implementación

Resumen del código de producción.

ID Historia de Usuario y Nombre	AC ID	Nombre clase(s)	Nombre método(s)	Nombre desarrollador(es)	Status	Notas (opcional)
1 Colocar una pieza	1.1	Game() Table()	colocar_ficha() check_empty()	Ruben Juan Michael	Completado	
	1.2	Table()	check_empty()	Juan Ruben Michael	Completado	
2 Mover una pieza	2.1				por Hacer	
3 Volar una pieza	3.1				por Hacer	
4 Elegir turno del jugador	4.1	Game()	cambiar_turno()	Michael	completado	
	4.2	Game()	cambiar_turno()	Michael	completado	
5 Configurar un juego	5.1	Game()	__init__() update()	Ruben	Completado	
	5.2	Game()	colocar_ficha()	Ruben	Completado	
6 Determinar si el juego ha terminado	6.1				por Hacer	

Resumen del código de prueba automatizado (correspondiente directamente a algunos criterios de aceptación)

ID Historia de Usuario y Nombre	AC ID	Nombre de clase (s) del código de prueba	Nombre del método(s) del código de prueba	Descripción del caso de prueba (entrada y salida esperada)	Status	Nombre desarrollador(es)
1 Colocar ficha	1.1	TestGame()	Test_colocar_ficha()	Verificación de que la ficha es colocada y el espacio es ocupado, salida esperada True y False respectivamente	Completado	Juan
2 Mover pieza	2.1	TestFicha()	Test_move()	Verificiación de colocación correcta al mover	En proceso	Juan

4 Cambiar Turno	4.1	TestGame()	Test_cambiar_turno() Test_cambiar_turno2()	Verificación de que cambie el turno, entrada el color en juego y salida que se confirme respectivamente el color.	Completado	Juan
7 Quitar Pieza de Oponente	7.1	TestTable()	Test_molino_True() Test_molino_False() Test_molino_True2() Test_molino_False2()	Verificación de que el código confirme correctamente la formación de un molino, salida esperada True y False respectivamente para cada prueba	Completado	Juan
	…					

Resumen de casos de prueba manual (correspondientes directamente a algunos criterios de aceptación)

ID Historia de Usuario y Nombre	**Criterio Aceptación ID**	**Entrada de casos de prueba**	**Prueba Oráculo (salida esperada)**	**Status**	**Notas**	**Nombre desarrollador(es)**
1.Colocar una pieza	1.1	Coordenadas del plano(pantalla)	Su aproximación a una matriz de 7x7	completado	Hay una función en game que sirve para transformar de x,y(posicion del mouse) a filas y columnas	Ruben

Resumen de otras pruebas automatizadas o manuales (no correspondientes a los criterios de aceptación)

Número	**Entrada de prueba**	**Resultado esperado**	**Nombre de clase del código de prueba**	**Nombre del método código de prueba**	**Status**	**Nombre desarrollador(es)**

IV. Resumen del código fuente

Código de producción o prueba ?	Nombre del archivo de código fuente	# lineas código	Nombre desarrolladores y contribuciones (% de código fuente)
Produccion	table.py	107	Ruben Anthony Ricapa Corrales 80% Gerardo Michael Carpio Usquiano 20%
Produccion	constants.py	26	Ruben Anthony Ricapa Corrales %85 Camarena Zamalloa Juan José%15
Produccion	ficha.py	32	Ruben Anthony Ricapa Corrales 100 %
Produccion	game.py	75	Ruben Anthony Ricapa Corrales %60 Gerardo Michael Carpio Usquiano 15% Camarena Zamalloa Juan José%15
Produccion	main.py	23	Ruben Anthony Ricapa Corrales %100

Prueba	test_table.py	48	Camarena Zamalloa Juan José%100
Prueba	test_ficha.py	18	Camarena Zamalloa Juan José%100
Prueba	test_game.py	34	Camarena Zamalloa Juan José%100
	Total	363	

V. Acta de reuniones

Reporta las actas de todas las reuniones, incluidas, entre otras: reunión de planificación de proyecto/sprint, reunión de trabajo, backlog grooming , reunión retrospectiva y sesiones de programación en pares.

Fecha	Tiempo y Depuración	Lugar	Nombre Participantes	Propósito de la reunión	Elementos de acciones específicos
3/05	1 hora	Meet - hie-kddt-tts (google.com)	-Ruben Anthony Ricapa Corrales -Juan José Camarena Zamalloa -Gerardo Michael Carpio Usquiano	- Analizar el sprint 1	-Avance de las historias de usuario
4/05	1 hora	Meet - saq-poeg-exw (google.com)	-Ruben Anthony Ricapa Corrales -Juan José Camarena Zamalloa -Gerardo Michael Carpio Usquiano	- Avanzar las historias de usuario	-Repartimos el trabajo de las historias de usuario y los criterios de aceptación
5/05	1 hora	Meet - fyd-qdco-fsh (google.com)	-Juan José Camarena Zamalloa -Ruben Anthony Ricapa Corrales -Gerardo Michael Carpio Usquiano	- Ver los avances de criterios de aceptación	-Acordamos terminar los criterios de aceptación
9/05	2 horas	Meet (google.com)	-Ruben Anthony Ricapa Corrales -Juan José Camarena Zamalloa -Gerardo	- Discutir la implementación de los criterios de aceptación	- Se completo las clases tablero, ficha y game
11/05	2 horas	Meet (google.com)	-Juan José Camarena Zamalloa	- Implementar los criterios de aceptación en python	-

			-Gerardo Michael Carpio Usquiano -Ruben Anthony Ricapa Corrales		
12/05		Meet - mdh-xtj e-zyn (google .com)	-Juan José Camarena Zamalloa -Gerardo Michael Carpio Usquiano -Ruben Anthony Ricapa Corrales	-Grabar el video del sprint 1	-Retoques en el documento

VI. Calificación de amigos

Si no te sientes cómodo al incluir tus calificaciones en este informe, puedes enviarlas por correo electrónico al profesor.

Calificación receptor

	Ruben Anthony Ricapa Corrales	Juan José Camarena Zamalloa	Gerardo Michael Carpio Usquiano
Ruben Anthony Ricapa Corrales		20/20	20/20
Juan José Camarena Zamalloa	20/20		20/20
Gerardo Michael Carpio Usquiano	20/20	20/20	
Promedio	20/20	20/20	20/20