Universidad Nacional del Sur
Departamento de Ciencias e Ingeniería de la Computación

Tecnología de Programación (TdP)

Segundo Cuatrimestre 2024

Planilla de Corrección Proyecto 1 ***Comisión 18***

Nota final: B -	**Aprobado**

Análisis preliminar del proyecto: *descarga y compilación del código fuente; uso de la aplicación en ejecución.*	
¿Todos los alumnos mostraron actividad en el uso del repositorio y las defensas?	**SI** - Se observa una menor injerencia en la contribución de código del alumno Sosa Tomas.
¿El código se compila?	**NO** - El código se compila correctamente luego de unas modificaciones de nombres y classpath.
¿El juego muestra tiempo, puntaje y vidas en todo momento? ¿El juego implementa dos modos de juego diferentes? ¿El juego opera el ranking de forma consistente?	**SI**
¿El juego se comporta como se espera (estado inicial, condición de finalización, transición de como mínimo 3 niveles)?	**SI**
¿El juego se comporta como se espera ante la interacción con los power-ups (cambio de estado, cambio de sprites)?	**SI**
¿El juego se comporta como se espera ante la interacción con los enemigos (condición para muerte, estados intermedios de enemigos tales como tortuga)?	**SI**
¿El juego se comporta como se espera respecto del lanzamiento de bolas de fuego?	**SI**
Virtudes de la GUI y usabilidad	- Sin observaciones.
Debilidades de la GUI y usabilidad	**SI** - El tamaño de la pantalla es excesivamente pequeño. - El juego se cierra al finalizar la partida, tanto al ganar como al perder.

	- Varias veces, sobre todo al estar desplazándose el jugador horizontalmente, los inputs para saltar no se registran, perjudicando la jugabilidad del juego.
Errores y/o características que deberían modificarse	- Sin observaciones.

(sigue debajo)

Análisis preliminar del proyecto: *diagramas reducidos, extendidos y de secuencia.*	
¿Se encuentran accesibles los *diagramas reducidos* y *extendidos* de clases de cada una de las defensas?	**SI**
¿Se encuentran accesibles los *diagramas de secuencia* solicitados para la defensa correspondiente?	**SI**
¿Se encuentran accesibles los *diagramas reducido y completo de clases* de la entrega final?	**SI** - Los nombres de las clases no son consistentes con mayúsculas, minúsculas y underscores, y no están actualizados con los nombres de las clases en el código. - El patrón State para el estado de mario está duplicado entre MarioState y MarioBase.

(sigue debajo)

Universidad Nacional del Sur
Departamento de Ciencias e Ingeniería de la Computación

Tecnología de Programación (TdP)

Segundo Cuatrimestre 2024

Análisis del código fuente: *inspección del código fuente entregado.*	
¿El código fuente se encuentra adecuadamente organizado en paquetes?	**SI**
¿El código exhibe de forma consistente y adecuada las prácticas propuestas por Clean Code?	**NO** - Existen println**s** en el código. - Hay inconsistencias de nombres. - Existen infinidad de líneas de código comentadas. - Se observan comentarios deliberados que solo ofuscan el código. Recuerde que, el mejor comentario posible es el que nunca debería existir.
¿Existe una adecuada división de responsabilidades entre las clases que implementan cuestiones de lógica/gráfica?	**NO** - ControladorVista no debería ser responsable por controlar las colisiones, debería estar separado en otro módulo, aunque esto fue señalado por la comisión así que son conscientes de ello.
¿Se implementa de forma consistente la detección de colisiones?	**SI**
¿Se implementa de forma consistente la resolución de colisiones?	**SI**
¿Se implementa de forma consistente los diferentes estados de Mario?	**NO** - Hay duplicación y mezclas de responsabilidades entre las clases, MarioState y MarioBase, y el hecho de que Mario implementa MarioState resulta confuso.
Virtudes	- Sin observaciones.
Debilidades	**SI** - No resulta claro cuál es la ventaja de manejar la cantidad de clases y operaciones de forma estática que proponen. Esto debería reconsiderarse, y mantener asociaciones uni o bidireccionales entre las clases que deban relacionarse, según corresponda.
Errores y/o características que deberían modificarse	- Existen infinidad de líneas de código comentadas, cuestión que atenta contra la legibilidad del código.