



Universidad Nacional del Sur
Departamento de Ciencias e Ingeniería de la Computación
Tecnología de Programación (TdP)
Segundo Cuatrimestre 2022



Planilla de Corrección Proyecto 2
Comisión 05

<u>Link repositorio</u>	<u>Integrantes</u>
https://github.com/sgraziabile/tdp-proyecto-2	<ul style="list-style-type: none">- GRAZIABILE, STÉFANO- GALLARDO, JULIÁN- ARAVENA, JOAQUÍN NICOLÁS- AGUILAR, EZEQUIEL MAXIMILIANO

Entrega:

<u>Nota final:</u> C	Aprobado.
----------------------	------------------

Análisis preliminar del proyecto: *descarga y compilación del código fuente; uso de la aplicación en ejecución.*

¿Todos los alumnos realizaron al menos un commit?	SI
¿El código se compila?	SI
¿El juego muestra puntaje, tiempo y considera al menos 3 tipos de power-up y al menos 5 tipos de alimentos?	SI
¿El juego se comporta como se espera (estado inicial de criatura, condición de finalización, transición de niveles)?	SI
¿El juego se comporta como se espera (cambio dirección, movimiento, colisión, crecimiento y cambio de estado)?	SI
Virtudes de la GUI y usabilidad	<ul style="list-style-type: none">- Se muestra un menú principal que funciona correctamente y además existe una pantalla de transición entre niveles.- La ventana tiene nombre e icono.- Se permite reiniciar el juego sin tener que reiniciar la aplicación y además hay un botón de salir.- Los niveles incrementan la dificultad del juego al avanzar
Debilidades de la GUI y usabilidad	<ul style="list-style-type: none">- A pesar de que se modifica la dirección de la criatura, la imagen de su cabeza siempre mira en la misma dirección.- Se podría mejorar la visibilidad utilizando un color más neutro en el fondo.
Errores y/o características que deberían modificarse	<ul style="list-style-type: none">- El ranking no funciona correctamente. Todas las posiciones muestran el nombre y el puntaje del último jugador ingresado.



Universidad Nacional del Sur
Departamento de Ciencias e Ingeniería de la Computación
Tecnología de Programación (TdP)
Segundo Cuatrimestre 2022



	<ul style="list-style-type: none">- Al pasar de nivel, si ya se jugó anteriormente no se muestra el nombre del nivel correctamente. Por ejemplo, si se juega hasta el nivel dos, al reiniciar el juego al pasar del nivel 1 al 2 se muestra una transición que dice nivel 3.- Al chocar una pared se muestran mensajes por consola, lo cual no debería suceder.
--	--

(sigue debajo)



Análisis del código fuente: *inspección del código fuente entregado.*

¿El proyecto contiene el diagrama de clases actualizado a la versión actual? Evaluación y comentarios del modelo.

NO

- El diagrama reducido no se corresponde con la implementación.
- El diagrama se podría haber reorganizado para mejorar su lectura, por ejemplo, se podrían haber posicionado las clases que heredan de Entidad debajo de la clase Entidad y no alrededor en la parte superior del diagrama.
- La decisión de modelar el cuerpo de la SerpienteLogica es cuestionable, y no hacía realmente falta: la serpiente se constituía como una agregación de bloques.
- Las clases con las que se modeló e implementó el patrón de diseño visitor no corresponden: ¿no es la serpiente la que visita bloques que pueden o no tener alimentos y power-ups? En consecuencia, ¿no es la serpiente quien debía entonces ser el visitor heredando su interface? ¿por qué crea una entidad colisionVisitor que ni siquiera se corresponde con el problema modelado?

¿El código fuente se encuentra adecuadamente organizado en paquetes?

SI.

¿Existe una adecuada división de responsabilidades entre las clases que implementan cuestiones de lógica/gráfica?

SI

- No debería ser responsabilidad de la clase Juego hacer visible a una entidad Grafica.

¿Existe un modelado adecuado de la criatura?

SI.

¿Existe una adecuada resolución de las colisiones? ¿Y respecto del afectar la criatura con power-ups y alimentos?

SI.

Virtudes

- Los nombres de los atributos son consistentes entre clases, lo cual facilita la lectura del código.

Debilidades

- Existen dos clases con comportamiento similar llamadas Leer. No es correcto tener dos clases con el mismo nombre, ni tener dos clases con funcionalidades similares, ni nombrar las clases utilizando verbos.
- La clase Ventana se encarga de modelar la GUI y de lanzar la aplicación. Se podría haber creado una clase Launcher

Errores y/o características que deberían modificarse

- Se deberían eliminar las instrucciones `System.out.println(" ")` en los distintos métodos.