



Universidad Nacional del Sur
Departamento de Ciencias e Ingeniería de la Computación
Tecnología de Programación (TdP)
Segundo Cuatrimestre 2021



Planilla de Corrección Proyecto 2
Comisión 17

<u>Link repositorio</u>	<u>Integrantes (nota)</u>
http://github.com/daro-all/tdp-proyecto-2	- PRADA, MARCOS DAMIAN - LEAL, DARIO ADRIÁN - SCHMIT, LUCAS GABRIEL - GONZALEZ DUPOY, VALENTÍN

Etapas 2:

1. ¿Todos los alumnos se presentaron a horario a la exposición?	SI
Sin observaciones.	
2. ¿El repositorio posee el modelo de clases subido y accesible en algún formato legible (imagen, pdf)?	SI
El png presentado está muy desordenado.	
3. ¿El proyecto contempla el conjunto de entidades (Lógica, GUI, Celda, Tetrimino) para modelar correctamente el problema?	SI
Sin observaciones.	
4. ¿El proyecto relaciona de forma adecuada las diferentes clases consideradas? Por ejemplo, no se debería hacer abuso de asociaciones, cuando corresponden agregaciones y/o composiciones. Por otro lado, no deberían existir relaciones entre todas las clases con todas las clases.	NO
El diagrama no tiene agregaciones ni composiciones. Abusan de las asociaciones, está relacionado todo con todo.	
5. ¿Se modelan adecuadamente los diferentes Tetriminos, en relación a una jerarquía de herencia entre clases y a la vez, como una agregación de celdas?	NO
No, plantearon un algoritmo que se basa en el alto y ancho. No hay agregación.	
6. Traza funcionalidad 1: mover Tetrimino a izquierda. ¿El diagrama modela adecuadamente todas las relaciones y operaciones necesarias para resolver esta funcionalidad? ¿Se encuentra correctamente resuelto la actualización de la GUI a partir de la modificación de la grilla de bloques?	SI
Sin observaciones.	
7. Traza funcionalidad 2: mover Tetrimino abajo. ¿El diagrama modela adecuadamente todas las relaciones y operaciones necesarias para resolver esta funcionalidad? ¿Se encuentra correctamente resuelto la actualización del puntaje y el control de si el juego contiene líneas completas?	SI
Sin observaciones.	
8. ¿Las respuestas de los alumnos a las consultas realizadas fueron adecuadas? Especifique los casos puntuales de falencias, en caso de corresponder.	NO
Las respuestas hacían mayor referencia a la implementación y no tanto al modelo. En varios casos las respuestas fueron insuficientes. Hubo un caso en el que se preguntó una cosa y se explicó otra totalmente diferente.	



Universidad Nacional del Sur
Departamento de Ciencias e Ingeniería de la Computación
Tecnología de Programación (TdP)
Segundo Cuatrimestre 2021



Entrega:

Nota final: D

Desaprobado

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? ¿Warnings?

SI

¿El juego muestra puntaje, tiempo, próximo tetrimino y considera los 7 tipos de tetriminos posibles?

SI

¿El juego se comporta como se espera (rotación, colisión, control de líneas, puntaje, condición de finalización, reinicio de juego)?

NO

Probando el juego, se encontraron casos en los cuales las piezas que colisionan se fusionan en vez de apilarse. Para ser más específico, sucede cuando la primera y la segunda pieza son del mismo color.

Virtudes de la GUI y usabilidad

- Se implementó la posibilidad de acelerar la caída del tetrimino.
- Se utilizó un tipo especial de representación para el puntaje que asemeja a un juego tetris original.

Debilidades de la GUI y usabilidad

- Al iniciar el juego el tetrimino que se visualiza como siguiente es en realidad el actual.

Errores y/o características que deberían modificarse

- Se debe corregir la razón por la cual algunos bloques se fusionan con otros.

(sigue debajo)



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

¿El proyecto contiene el diagrama de clases actualizado a la versión actual?	SI
¿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
¿Existe una adecuada jerarquía de clases para los diferentes <i>tetriminos</i> (resolución mover/rotar)?	NO Se implementaron los tetriminos mediante un arreglo de posiciones. No existen clases específicas para cada tetrimino.
Virtudes	-
Debilidades	- El proyecto contiene <code>System.out.println</code> que deberían haberse eliminado antes de la entrega del mismo.
Errores y/o características que deberían modificarse	<ul style="list-style-type: none">- El modelo entregado no tiene las correcciones realizadas durante la defensa, esto no favorece la reusabilidad ni la extensibilidad del juego.- Para cumplir con lo indicado en el modelo, un tetrimino es una agregación de cuatro bloques, en la implementación la clase "Pieza" debería referenciar a cuatro bloques de la grilla.- La resolución de la rotación mediante una matriz no es la forma más eficiente para el desarrollo del juego.