



Proyecto Unidad 2 (Extensión) Google's Dino Run

Objetivos

- Trabajar en parejas para implementar soluciones a problemas complejos
- Aplicar conocimientos de diseño de software y patrones de diseño para diseñar una solución a un caso particular.
- Implementar la solución al caso particular a través de un programa con funcionalidad limitada.

Instrucciones

- **Extensión para trabajar en grupos de 3 personas.**
- Leer el enunciado del caso y realizar las actividades especificadas en la sección final.
- Exportar el informe en **formato PDF**. *Asegúrese que los diagramas se encuentren en buena calidad (ideal SVG) y se puedan ver fácilmente al hacer zoom.*
- Empaquetar el informe, el código y cualquier otro archivo que usted desee agregar en un archivo **formato ZIP**.
- **Realizar la entrega del archivo ZIP a través de Educandus**, en el enlace correspondiente. Un archivo por grupo.
- Incluir los nombres de todos los integrantes como comentario en la entrega.
- La **fecha límite** de entrega es el Martes 12 de Noviembre, hasta las 23:55 hrs. *Se aceptan entregas posteriores con nota máxima 4.0*

Nota: Todas las entregas serán revisadas en [Copleaks](#) y [MOSS](#). En caso de que cualquiera de las dos herramientas detecte copia o plagio; todas las entregas involucradas serán evaluadas con nota mínima sin posibilidad de recuperación.

Google's Dino Run



Chrome popularizó un juego en el cual, cuando no tienes conexión, puedes entretenerte jugando con un pequeño dinosaurio que corre infinitamente en un plano 2D hacia la derecha mientras esquivaba obstáculos.

El objetivo del juego es ayudar al dinosaurio a avanzar lo más lejos posible, esquivando obstáculos como cactus y pterosaurios. Para hacerlo, el dinosaurio puede saltar (usando la barra espaciadora o la flecha hacia arriba) y agacharse (usando la flecha hacia abajo). A medida que el tiempo pasa, la velocidad del juego aumenta, los obstáculos se vuelven más frecuentes y aparecen patrones más complejos, lo que incrementa la dificultad.

El puntaje se incrementa conforme avanza y se muestra en la esquina superior derecha de la pantalla. El juego guarda automáticamente tu puntaje máximo para que puedas intentar superarlo en partidas futuras. Para iniciar el juego, simplemente presiona la tecla de salto la primera vez. Al finalizar, tienes la opción de reintentarlo presionando nuevamente la tecla de salto.

Puede probar el juego en el siguiente enlace: <https://trex-runner.com>

Actividades

Utilizando un lenguaje orientado a objetos y la librería de su preferencia (Java y JavaFX *canvas* por ejemplo); realice la implementación de una réplica del juego con las siguientes reglas:

- Para manejar el estado del juego y los puntajes máximo/actual, utilice una clase que implemente el patrón **singleton**.
- Para crear los distintos obstáculos utilice un **abstract factory**.
- Instancie los obstáculos a través del factory una única vez y cree copias utilizando el patrón **prototype**.

El juego además tendrá unas **variaciones respecto del original**:

- Los obstáculos se podrán comportar de dos formas. **Tradicionales y dinámicos**. Los tradicionales siguen la regla del juego normal, mientras que los dinámicos se transforman al azar entre obstáculos de piso (para saltar) y obstáculos de cielo (para agacharse). Utilice una **strategy** para manejar este comportamiento
- El juego deberá tener **3 “temáticas” distintas**, las cuales se deben seleccionar antes de iniciar el juego. Todas tendrán la misma lógica y obstáculos. Las temáticas serán: Una tradicional de desierto, otra temática de bosque y otra a su elección. Utilice distintas **“factories”** del abstract factory para manejar los distintos escenarios.

Se espera que al abrir el juego se permita seleccionar la “temática” deseada. Luego de eso el juego comenzará cuando el usuario presione la tecla salto. Una vez el jugador pierda, se mostrará el puntaje obtenido y la opción de reiniciar el juego. En los intentos siguientes siempre se deberá mostrar el puntaje mayor obtenido desde que se abrió el juego.

Extensión para 3 personas.

En adición a jugabilidad y reglas básicas del juego; será necesario agregar 2 funcionalidades extra:

1.- **Monedas:** En el escenario deben aparecer monedas similares a las que vemos en juegos como Super Mario Bros. Las monedas deben estar ubicadas de forma en que el dinosaurio deba saltar para recogerlas.

Cada moneda deberá agregar una cantidad extra al puntaje actual. Al igual que los obstáculos las monedas deberán comportarse de dos formas: **Tradicional y dinámica**. Las tradicionales estarán estáticas en una posición, mientras que las dinámicas deberán moverse de arriba hacia abajo.

Al igual que los otros componentes del escenario deberá utilizar un factory para instanciarla, un prototype para clonarla y un strategy para manejar las dos formas de comportamiento.



2.- **Disparo:** En vez de un dinosaurio utilizarán un dragón, el cual podrá disparar bolas de fuego utilizando la tecla que ustedes definan. Los disparos siempre son hacia adelante en la posición en la que está el dinosaurio en el momento en que se presiona la tecla.

En conjunto a los disparos deberá agregar un **nuevo tipo de obstáculo**: Un bloque rompible. El bloque rompible aparecerá al igual que los otros obstáculos, pero en vez de saltar, el jugador deberá dispararle con la bola de fuego para poder romperlo y no chocar contra él.

Para el bloque rompible deberá seguir la misma regla de utilizar un Factory para instanciarlo y un prototype para duplicarlos.