

ESCUELA COLOMBIANA DE INGENIERÍA PROGRAMACIÓN ORIENTADA A OBJETO POOBuilding 2016-02

Propuesta: Juan José Andrade - María Camila Gómez

repara-FÉLIX JR

Repara-FÉLIX JR es un videojuego de plataforma arcade inspirado en la película de Disney Wreck-It Ralph. Este es un juego clásico donde lucha un villano llamado Ralph con un héroe llamado Félix. .

El entorno en el que se desarrolla el juego es un edificio residencial donde Ralph rompe los vidrios del edificio; los residentes piden ayuda a Félix y empieza una lucha entre las habilidades de construcción de Félix, que cuenta con un martillo mágico, y la habilidad de lanzamiento de ladrillos de Ralph. El objetivo principal de Félix es arreglar todos los vidrios del edificio y el de Ralph destruirlo con golpes de ladrillo.

La historia transcurre en una ciudad donde hay patos silvestres volando que pueden tumbar a Félix del edificio. Por ultimo siempre existe un elemento sorpresa y son los pasteles que algunas abuelitas brindan a Félix. Como siempre un buen pastel hace invulnerable al héroe contra ladrillos y permite que construya rápido los vidrios del edificio.

<http://juegos.disney.es/rompe-ralph/repara-felix-jr>



POOBuilding

Para reparar las ventanas de edificio **POOBuilding** más rápidamente, Félix ha decidido invitar a uno de sus amigos para que le ayuden.

Los dos trabajan en la reparación de manera independiente y, como son muy competitivos, pueden simular tropezar con su compañero para hacerlo caer por los bordes del edificio.

POOBuilding tiene diferentes tipos de ventanas que se diferencian por el número de vidrios. En los pisos pueden existir obstáculos horizontales o verticales que impiden que los personajes los atraviesen.

Un héroe gana cuando, al repararse todas las ventanas, logra más puntos que su compañero.



Heroes

Los héroes solamente pueden moverse (derecha, izquierda, arriba, abajo) y golpear un vidrio con su martillo mágico para repararlo. Los dos movimientos les restan energía, pero el que más la consume es la reparación. Adicionalmente, los héroes pierden energía al recibir un golpe de ladrillo o caerse del edificio; en el primer caso, la pérdida es inversamente proporcional y la cercanía de Ralph y en segundo caso, directamente proporcional a la distancia al piso.

De cada uno de los héroes interesa conocer:

- **Vidas:** El número de vidas que le quedan
- **Energía:** El nivel de energía, de su vida actual, puede ser de 0 .. 100%. Si la energía cae a 0 el héroe muere; si es menor a 25%, no puede dar dos golpes seguidos; y si es menor a 50%, es más lento en sus movimientos.
- **Puntos:** Es el valor acumulado por las ventanas reparadas. El valor de una ventana depende de su número de vidrios.
- **Protección.** El detalle de los poderes que tiene en el momento.

Los héroes pueden ser controlados por el usuario o por la máquina. En este momento existen dos tipos de héroes máquina:

- **Calhoun:** Los movimientos de héroe no siguen ninguna estrategia.
- **Candy:** Su prioridad es lograr que Ralph acabe con su compañero bloqueando su movimiento.

Sorpresas

Los héroes pueden recibir elementos sorpresas que les ofrecen los habitantes del edificio para ayudarlos en su trabajo.

Actualmente, se tienen tres tipos de elementos.

- **Pastel:** Hace que puedan reparar cualquier ventana con un único golpe del martillo reparador. Su efecto termina cuando cambian de piso.
- **Bebida energizante:** Incrementa el nivel de energía de los héroes. Los héroes la pueden tomar cuando su nivel de energía es menor del 50% y su efecto es aumentar al doble el nivel de energía actual.
- **Kriptonita:** Lo hace invulnerable a cualquier ataque. El efecto dura poco tiempo.

Animales

El vecindario está poblado de animales que pueden perjudicar o ayudar a los héroes en su trabajo. En esta versión del juego sólo se tienen dos animales:

- **Patos:** Que al pasar volando pueden tumbar a los héroes del edificio.
- **Cigüeñas:** Que toman a los héroes en sus picos y los llevan al piso inferior de dónde está Ralph.

REQUISITOS FUNCIONALES

La aplicación debe:

- ➔ Permitir generar el edificio de juego
- ➔ Permitir importar la descripción de los elementos del juego
- ➔ Permitir abrir y salvar el estado de un juego
- ➔ Permitir seleccionar el nombre del jugador usuario o el nombre del jugador máquina
- ➔ Presentar permanentemente el estado del juego
- ➔ Permitir consultar el estado de cada jugador
- ➔ Permitir que el jugador usuario realice los movimientos.
- ➔ Presentar el movimiento del jugador computador de manera "inmediata"
- ➔ Decidir cuando termina el juego y comunicar la causa.
- ➔ Permitir terminar el juego en cualquier momento indicando el ganador hasta el momento.

REQUISITOS DE DISEÑO

De visualización

- ➔ El espacio de juego debe tener una representación gráfica detallada que permita conocer el estado momento a momento.
- ➔ Se debe explicar el movimiento realizado en una zona de texto siempre visible.
- ➔ Se deben asignar íconos que permitan diferenciar los diferentes elementos de la batalla.

De extensión

- ➔ Permitir generar nuevas versiones de la aplicación que incluyan otros tipos de criaturas (por ejemplo, las arañas), de sorpresas (por ejemplo, una cerveza) y otros perfiles de héroe computador (por ejemplo, en la competencia, el nuevo perfil con la estrategia)

De manejo de excepciones

- ➔ Deben definir mínimo una nueva clase excepción para manejar las excepciones propias.
- ➔ Los métodos correspondientes a las acciones deben lanzar una excepción si la acción solicitada no es válida explicando claramente su causa.
- ➔ Cuando ocurra una excepción no esperada o una propia grave se debe escribir esta información en el log de errores para los programadores y terminar la ejecución del mismo.

REQUISITOS DE ENTREGA

Funcionalidades

- ➔ Diagrama de casos de uso

Capa de presentación

- ➔ Boceto de interfaz gráfica
- ➔ Diagrama de clases (sólo componentes públicos)
- ➔ Código de programa
- ➔ Pruebas de aceptación

Capa de aplicación

- ➔ Diagrama de clases (completo)
- ➔ Diagramas de secuencia (desde la fachada)
- ➔ Código del programa
- ➔ Código de pruebas

Retrospectiva

- ➔ Estado actual del proyecto. Detalle por casos de uso.
- ➔ Tiempo invertido en la entrega.
- ➔ Problemas y logros tecnológicos.
- ➔ Problemas y logros como equipo. Compromisos.

DETALLES ENTREGAS

Versión uno Presentación - aplicación	Diagrama de casos de uso Boceto de interfaz gráfica Diagrama de clases de presentación (sólo componentes públicos) Diagrama de clases de aplicación (completo)	A par S14
Versión dos Aplicación- persistencia	Generar campo de juego Salvar guardar estado de juego Usuario vs. usuario Byakko: Mitrax Recompensa: Coin	A par S16
Versión tres	Funcionamiento completo	A evaluador Inicial Miercoles 7 de diciembre Final Martes 13 de diciembre
Versión cuatro Competencia	Es requisito para participar en la competencia que el equipo se haya presentado a todas las revisiones de pares y que todas las entregas del proyecto hayan sido aprobadas. El equipo ganador tiene 5.0 en la nota del tercer tercio.	Viernes 16 de diciembre