



Trabajo Práctico Final

Ing. De Software

“Arkanoid”

Profesor: Miceli, Martín.

Alumnos: BARRERA, Cesar.
GARBIGLIA, Diego.
REMONDA, Aurelio.
TISSOT, Esteban.

Introducción

El proyecto se basa en realizar el juego de Arkanoid de manera simplificada, para el cual solo se tendrá en cuenta la movilidad de la raqueta, el uso de dos bolas y la reproducción de un sonido al tocar las bolas contra la raqueta o las paredes.

En el informe se pretende analizar el ejemplo del Libro Head First Design Patterns en la página 526 a 548 (DJView), con el objetivo de extenderlo implementando una nueva funcionalidad del modelo realizado en el mismo.

Manejo de las Configuraciones

Dirección y Forma de Acceso

Se utiliza github como repositorio remoto https://github.com/estebantissot/arkanoid_MVC y git (en Ubuntu) como herramienta de control de versiones.

Esquema de Direcciones

El repositorio consta de un directorio /bin donde se encuentran los archivos class, un directorio /src donde se guardan los archivos java y los archivos .classpath y .project propios del proyecto creado en Eclipse.

Normas de Etiquetado y Nombramiento

Las versiones 1.* son versiones del juego Arkanoid funcionando solo. En las versiones 2.* están sumados al paquete Arkanoid la implementación del resto de los paquetes (Beat, Heart, etc.).

Plan de Esquema de Ramas

Solamente se usó la rama principal (master).

Se utilizaron tags para etiquetar código entregable (releases versionados).

Forma de entrega de los Releases

Los entregables se pueden descargar desde el repositorio en formato zip con su correspondiente proyecto para poder abrirse en Eclipse. El último release será un archivo ejecutable para la entrega al cliente, que no necesita de ningún proceso de instalación para ser ejecutado, pero sí deberá contar con un intérprete java (JRE Java Runtime Environment), por lo general vienen incluidos por defecto en la mayoría de los sistemas operativos.

Integrantes del Equipo

Integrantes	Forma de Contacto	Roles en la CCB
Barrera, Cesar	<ul style="list-style-type: none">- Redes Sociales: Facebook- Contacto Telefónico: Celular- Reuniones Periódicas- Correo Electrónico	<ul style="list-style-type: none">- Colaborador General- Master- Arquitecto Analista- El Diseñador (gráfico)- Informe
Garbiglia, Diego	<ul style="list-style-type: none">- Redes Sociales: Facebook- Contacto Telefónico: Celular- Reuniones Periódicas	<ul style="list-style-type: none">- Colaborador General- Master- Calidad (Tester)- Analista Arquitecto- Programador

	- Correo Electrónico	- El Diseñador (gráfico)
Remonda, Aurelio	<ul style="list-style-type: none"> - Redes Sociales: Facebook - Contacto Telefónico: Celular - Reuniones Periódicas - Correo Electrónico 	<ul style="list-style-type: none"> - Colaborador General - Master - Arquitecto Analista - Gestor de Proyectos - Programador
Tissot, Esteban	<ul style="list-style-type: none"> - Redes Sociales: Facebook - Contacto Telefónico: Celular - Reuniones Periódicas - Correo Electrónico 	<ul style="list-style-type: none"> - Colaborador General - Master - Arquitecto Analista - Gestor de Proyectos - Programador

Requerimientos

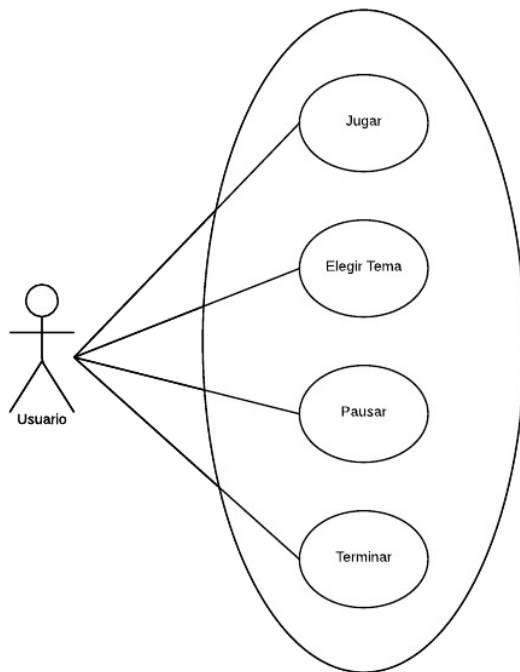
Requerimientos Funcionales

- El juego debe mostrar la puntuación durante la partida.
- El juego debe mostrar el SCORE final al terminar la partida.
- El juego debe tener dos bolas.
- El juego debe terminar cuando una de las bolas se pierda.
- Se debe manejar la barra con el teclado.
- Deberá tener efectos de sonido acordes a los eventos en el juego.
- La ventana debe ser de 400 x 580 píxeles.
- Se debe poder elegir el fondo del juego.
- El juego se debe poder pausar con una tecla.

Requerimientos NO Funcionales

- El juego debe ser portable.
- El software será para todo tipo de usuarios y edades, con una interfaz amigable que no tardará más de 5 minutos en amigarse con el mismo.
- Se debe poder empezar a jugar a los 5 segundos o menos de haber inicializado el programa.
- El desplazamiento de la barra debe ser fluido.
- El software es realizado con fines de entretenimiento.
- El juego debe finalizar al cerrarse la ventana del mismo.

Diagrama de Casos de Uso



Como se observa en la figura, el usuario del juego es el único que interactúa en el sistema, él tendrá la posibilidad de ejecutar la aplicación, jugar al Arkanoid, ponerlo en pausa en el momento deseado o bien terminar el juego, ya sea por Game Over o salir de la aplicación si es necesario.

Diagrama de Actividades

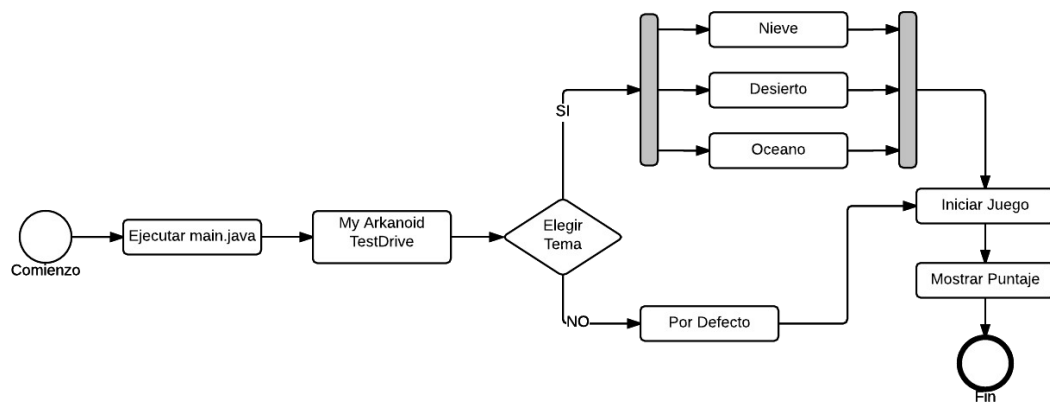
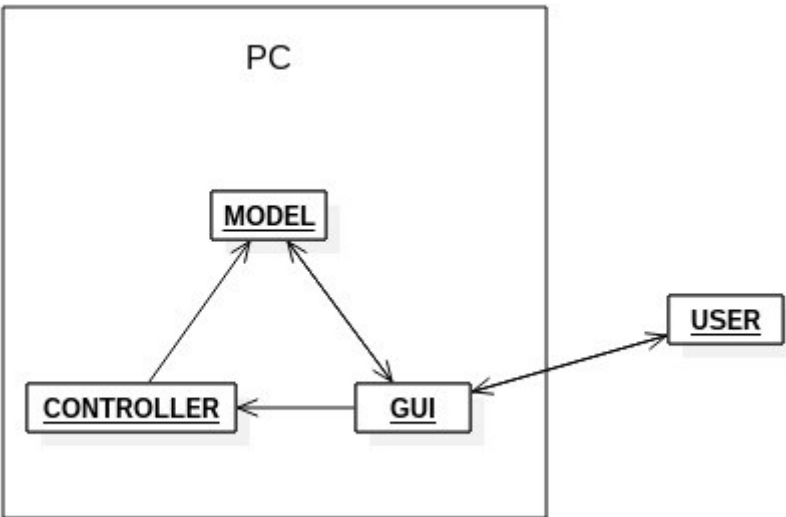


Diagrama de Arquitectura Preliminar



Matriz de Trazabilidad

	Jugar	Elegir Tema	Pausar	Terminar
El juego debe mostrar la puntuación durante la partida	X			
El juego debe mostrar el SCORE final al terminar la partida.				X
El juego debe tener dos bolas.	X			
El juego debe terminar cuando una de las bolas se pierda.				X
Se debe manejar la barra con el teclado.	X			
Deberá tener efectos de sonido acordes a los eventos en el juego.	X			
La ventana debe ser de 400 x 580 píxeles.	X			
Se debe poder elegir el fondo del juego.	X	X		
El juego se debe poder pausar con una tecla.	X		X	
El juego se debe poder pausar con una tecla.	X		X	

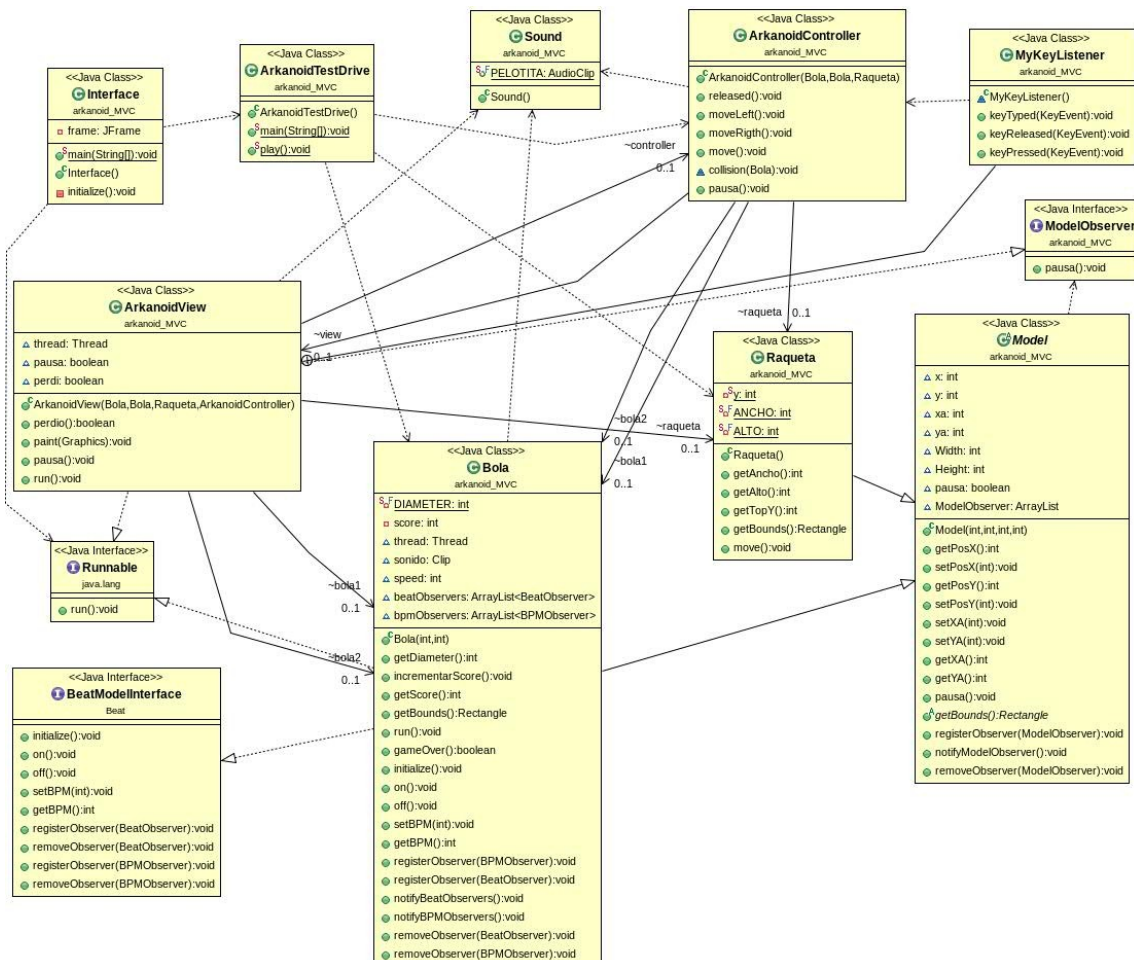
Arquitectura

La arquitectura utilizada es MVC, esto es porque permite una excelente mantenibilidad del código y hasta poder reutilizar partes del mismo, está compuesto por, la View (vista donde está todo lo relacionado con el usuario), el Model (modelo, donde se desarrolla la lógica del Arkanoid), y el Controller (controlador, es lo que modula hace interactuar el modelo con la vista). Si uno quisiera usar un mismo modelo pero con otra vista, solo debería modificar la vista y mantener las interfaces para poder comunicarse con el modelo a través del controlador, similarmente es posible cambiar el modelo y utilizar la misma vista, pudiendo también, en otro caso, tener varias vistas que muestran al mismo modelo de diferentes formas. Al estar separadas en 3 módulos, facilita la corrección de errores y la mejora del producto, como así la organización del software y por ende su calidad.

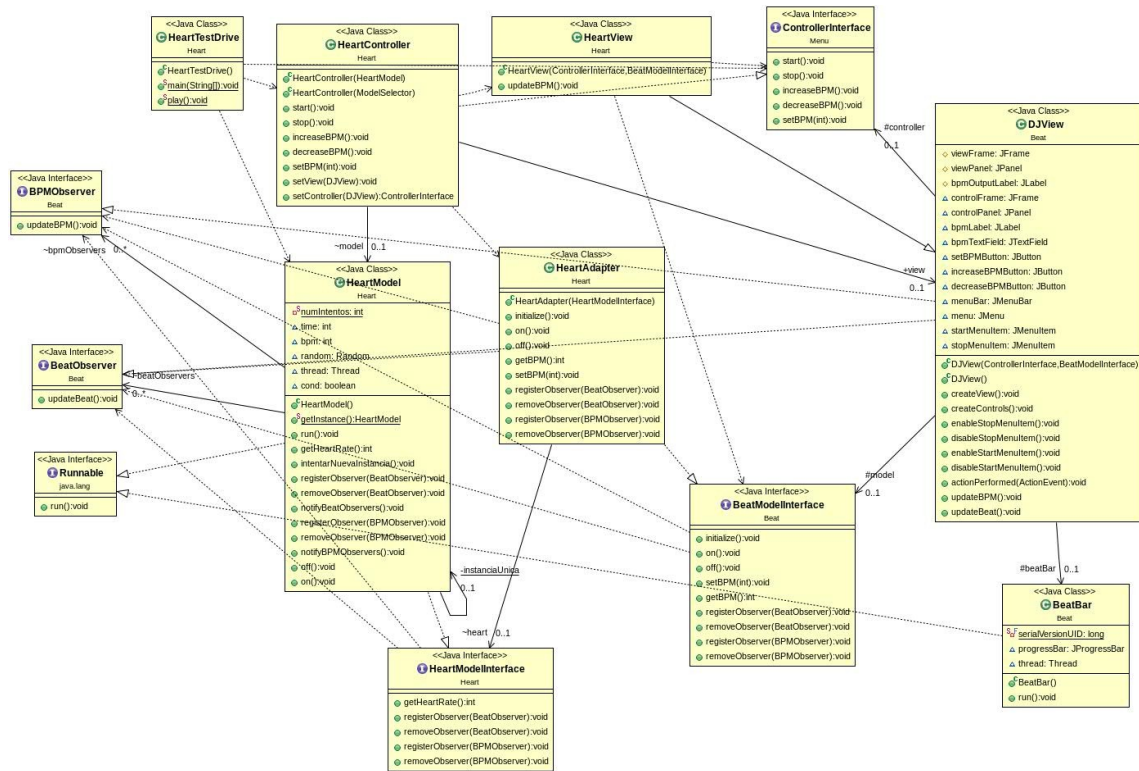
Diseño e Implementación

Diagrama de Clases aplicado a UML

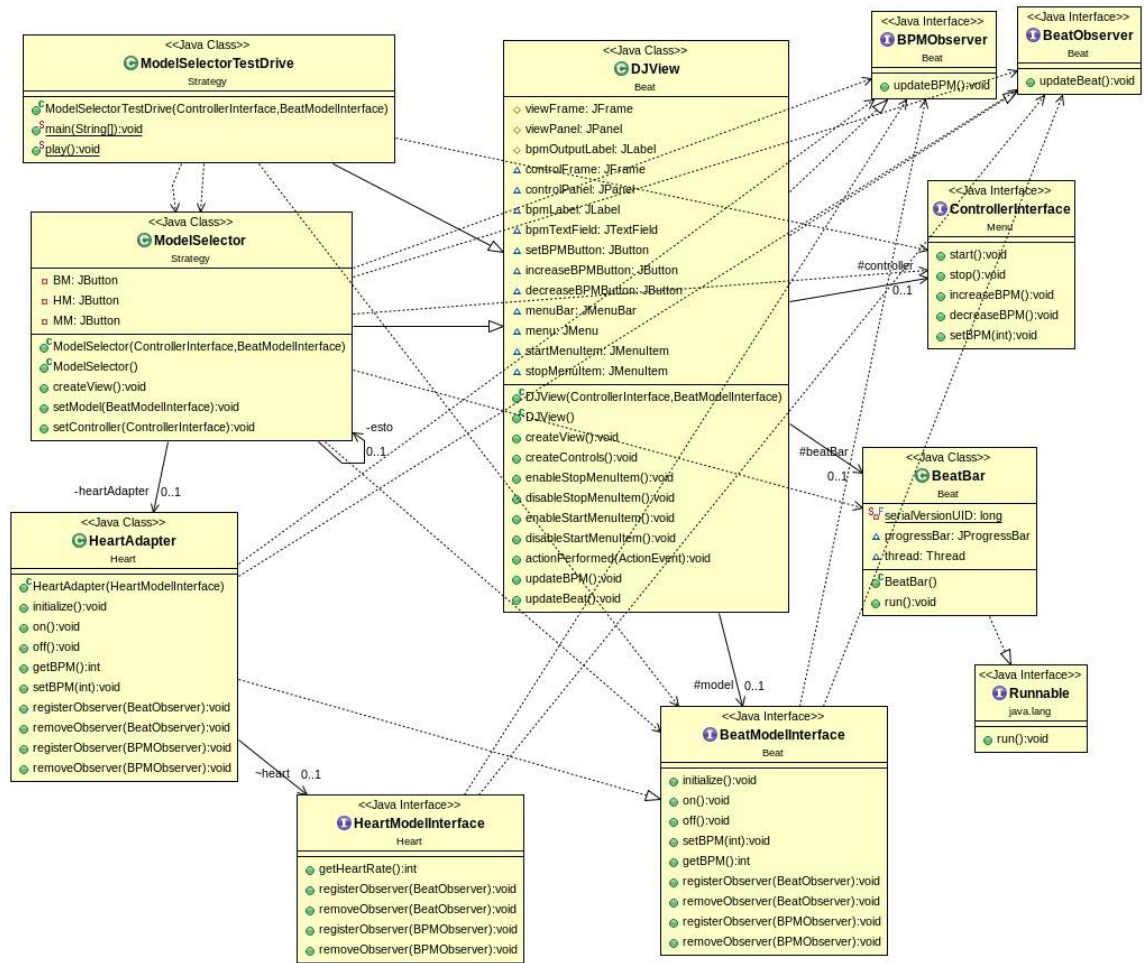
Arkanoid



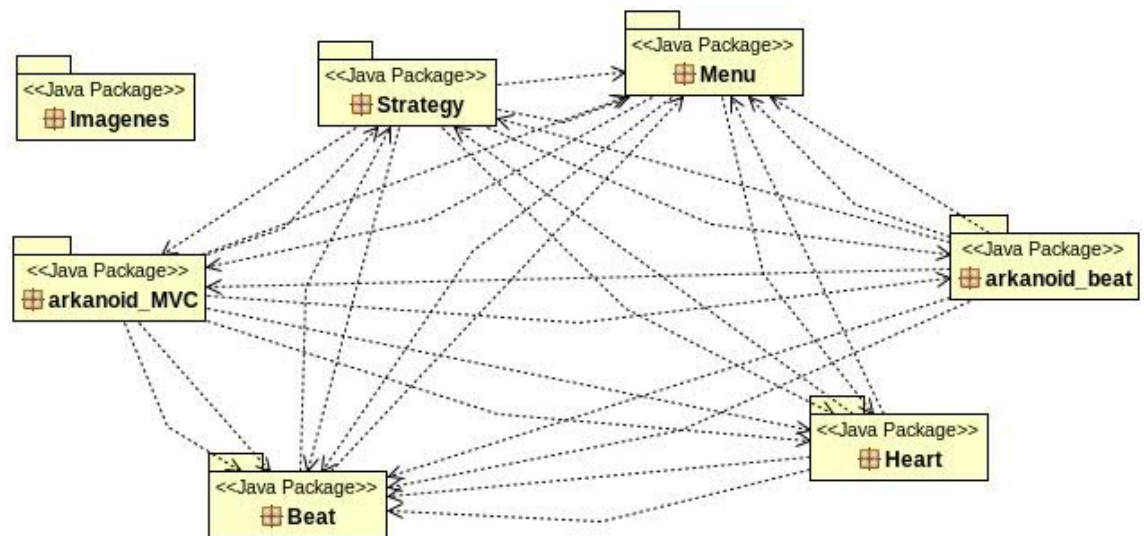
HeartModel (Singleton)



ModelSelector (Strategy)



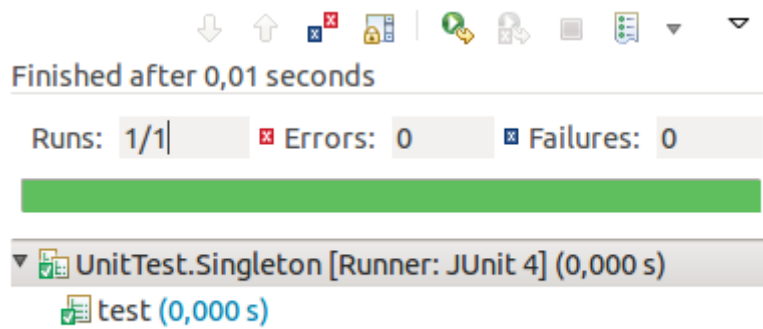
Diagramas de Paquetes



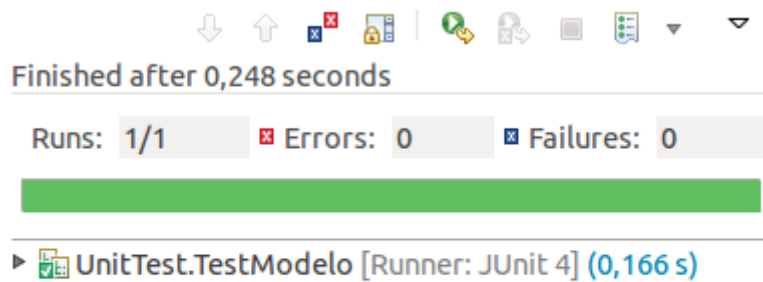
Pruebas Unitarias y del Sistema

Pruebas Unitarias

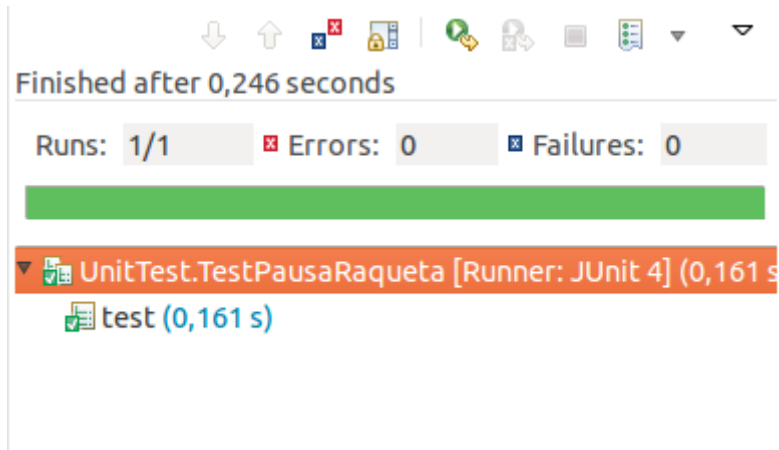
Singleton



Model




Raqueta



Pausa

Finished after 0,285 seconds



Runs: 1/1 Errors: 0 Failures: 0

▶  UnitTest.TestPausaBola [Runner: JUnit 4] (0,178 s)

Bola

Finished after 0,015 seconds



Runs: 1/1 Errors: 0 Failures: 1

▼  UnitTest.UnitTest [Runner: JUnit 4] (0,005 s)
  test (0,005 s)

Collision

Finished after 0,207 seconds

Runs: 1/1 Errors: 0 Failures: 0

▼  UnitTest.TestCollision [Runner: JUnit 4] (0,124 s)
  test (0,124 s)

All

↓

↑

✖

🔒

🔄

🔄

📄

📄

▼

▼

Finished after 0,34 seconds

Runs: 6/6

✖ Errors: 0

✖ Failures: 1

▼

📄

UnitTest.AllTests [Runner: JUnit 4] (0,258 s)

▼

📄

UnitTest.Singleton (0,000 s)

📄

 test (0,000 s)

▼

📄

UnitTest.UnitTest (0,002 s)

📄

 test (0,002 s)

▼

📄

UnitTest.TestPausaRaqueta (0,144 s)

📄

 test (0,144 s)

▼

📄

UnitTest.TestPausaBola (0,090 s)

📄

 test (0,090 s)

▼

📄

UnitTest.TestModelo (0,010 s)

📄

 test (0,010 s)

▼

📄

UnitTest.TestCollision (0,012 s)

📄

 test (0,012 s)

Casos de Pruebas de Sistemas

Descripción	Fin del juego
Ejecución	1 Abrir juego 2 Perder (dejar pasar una pelota)
Resultado Esperado	Al finalizar la partida debe salir un aviso de juego terminado con el score
Pasó/Falló	Pasó
Realizado por	Remonda Aurelio

Descripción	Pausar el juego
Ejecución	1 Abrir el juego 2 Presionar tecla P
Resultado Esperado	Cuando se pausa el juego deben frenar las pelotas y la raqueta. No se muestran ninguna de las anteriores en pantalla.
Pasó/Falló	Pasó
Realizado por	Tissot Esteban

Descripción	Cambiar fondo del juego
Ejecución	1 Abrir juego 2 Seleccionar el tema desde un menú.
Resultado Esperado	Que el fondo del juego y los colores de pelota y raqueta cambian de acuerdo a la elección.
Pasó/Falló	Falló
Realizado por	Garbiglia Diego

Descripción	Colisión raqueta pelota
Ejecución	1 Abrir juego 2 Jugar hasta que haya una colisión entre la raqueta y una pelota
Resultado Esperado	Que la pelota salga despedida en dirección contraria a la que venía (hacia arriba)
Pasó/Falló	Pasó
Realizado por	Barrera Cesar

Descripción	Puntaje
Ejecución	1 Abrir juego 2 Jugar hasta que haya una colisión entre la raqueta y una pelota
Resultado Esperado	Que aumente el puntaje.
Pasó/Falló	Pasó
Realizado por	Tissot Esteban

Descripción	Sonido
Ejecución	1 Abrir juego 2 Jugar hasta que haya una colisión entre la pelota y algún elemento (paredes o raqueta)
Resultado Esperado	Se ejecuta el sonido correspondiente.
Pasó/Falló	Pasó
Realizado por	Remonda Aurelio

Descripción	Recibir feedback del software
Ejecución	1 Hacer jugar a alguien ajeno al proyecto
Resultado Esperado	Que lo pueda jugar intuitivamente, que lo aprenda en poco tiempo
Pasó/Falló	Pasó
Realizado por	Tissot Dana

Pass/Fail

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
▼ arkanoid_MVC				3.430
▼ src				3.430
▶ arkanoid_beat				160
▼ arkanoid_MVC				1.319
▶ ArkanoidController.java				133
▶ ArkanoidTestDrive.java				29
▶ ArkanoidView.java				326
▶ Bola.java				231
▶ Interface.java				413
▶ Model.java				126
▶ Raqueta.java				52
▶ Sound.java				9
▶ Beat				921
▶ Heart				346
▶ Menu				47
▶ Strategy				315
▶ UnitTest				322

Datos Históricos

A continuación se detalla la dedicación de esfuerzo para realizar el trabajo en persona-horas.

Integrantes	Construir la Aplicación	Documentar la Aplicación	Elaborar el informe
Barrera, Cesar	20 Hs.	8 Hs.	20 Hs.
Garbiglia, Diego	18 Hs.	8 Hs.	20 Hs.
Remonda, Aurelio	26 Hs.	8 Hs.	15 Hs.
Tissot, Esteban	30 Hs.	8 Hs.	10 Hs.

Nota de Entrega

Listado de la Funcionalidad

- El juego debe mostrar la puntuación durante la partida. [LISTO]
- El juego debe mostrar el SCORE final al terminar la partida. [LISTO]
- El juego debe tener dos bolas. [LISTO]
- El juego debe terminar cuando una de las bolas se pierda. [LISTO]
- Se debe manejar la barra con el teclado. [LISTO]
- Deberá tener efectos de sonido acordes a los eventos en el juego. [LISTO]
- La ventana debe ser de 400 x 580 píxeles. [LISTO]
- Se debe poder elegir el fondo del juego. [NO IMPLEMENTADO]

Al momento de la entrega no se llegó a implementar la selección del fondo del juego en la interfaz gráfica.

BUGS Conocidos

Los bugs encontrados se fueron corrigiendo a medida que se fue desarrollando el código, sin utilizar una herramienta de administración de bugs.

Al momento del release existen bugs conocidos: Entre ellos se puede observar, que al ejecutar el Arkanoid, las bolas al chocar por primera vez con cualquiera de las paredes, se detienen un instante y luego continúan sin problemas; otro es cuando una bola se va, produciendo el Game Over, el sonido se sigue ejecutando.

Información Adicional

Conclusiones

Se aprendió el uso básico (cometiendo errores en el proceso de aprendizaje) de varias herramientas como:

- Git, para el versionado y para subir las versiones al repositorio remoto.
- La herramienta para generar unit test de Eclipse.
- Objectaid, plugin de Eclipse para generar diagramas UML.
- Framework Junit 4 de Eclipse para pruebas de código.
- WindowBuilder, plugin de Eclipse para crear ventanas.

Hemos visto las ventajas de la programación modular y del uso de patrones de diseño como utilidad a la hora de crear un proyecto o de modificar uno existente acoplando nuevas funcionalidades al mismo, así como nuevas vistas.

Tuvimos la oportunidad de trabajar con interfaces gráficas algo que no es muy estudiado en materias anteriores de la carrera así como la programación en equipo con la herramienta de versionado.