

Entrega GM Avance

Integrantes: Bastian Carrasco, Matias Fuentes y Jan Houter

- 1.1 Realizar una explicación del juego a realizar desde la perspectiva del jugador. Tutorial detallado de qué trata el juego (ej. objetivos, narrativa), elementos, y controles de movimiento/acción utilizando para ello maquetas o pantallazos de las interfaces gráficas**

El proyecto que se va a implementar consistirá en un juego en el que el jugador destruye bloques de tablas de bambú con un shuriken (pelota, en juego original), este al colisionar con un ninja (paddle, en el juego original) que se encuentra en la parte inferior hará que rebote hacia arriba para ir destruyendo los bloques. El ninja solo se moverá en forma horizontal y será controlado por el jugador con las flechas izquierda y derecha del teclado, para poder iniciar el juego se deberá presionar la tecla espaciadora y así darle inicio al lanzamiento del shuriken. El jugador ganara este juego después de destruir todos los bloques de bambú del nivel. Del mismo modo, el jugador pierde si no logra atrapar el shuriken con el ninja en 3 oportunidades.

Controles:



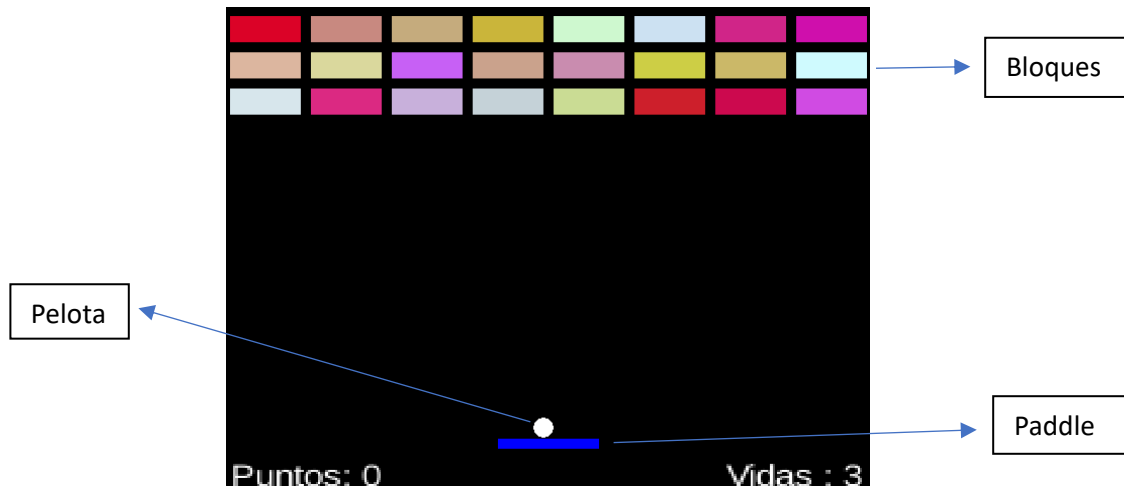
Barra
espaciadora:

Se presiona para
poder iniciar el
juego, se lanza el
shuriken.

Izquierda/Derecha:

Se presiona para poder mover al
ninja según se requiera (hacia la
izquierda o derecha).

En la siguiente imagen se encuentra el juego base de una manera visual.



Las modificaciones visuales que se implementarán al juego serán las siguientes:

- El paddle se quiere reemplazar por la imagen de un ninja.



- La pelota se quiere reemplazar por la imagen de un shuriken.



- Los bloques se quieren reemplazar por dos tipos (en el siguiente punto se explicara a detalle cada bloque)

Bloque de nivel 1:



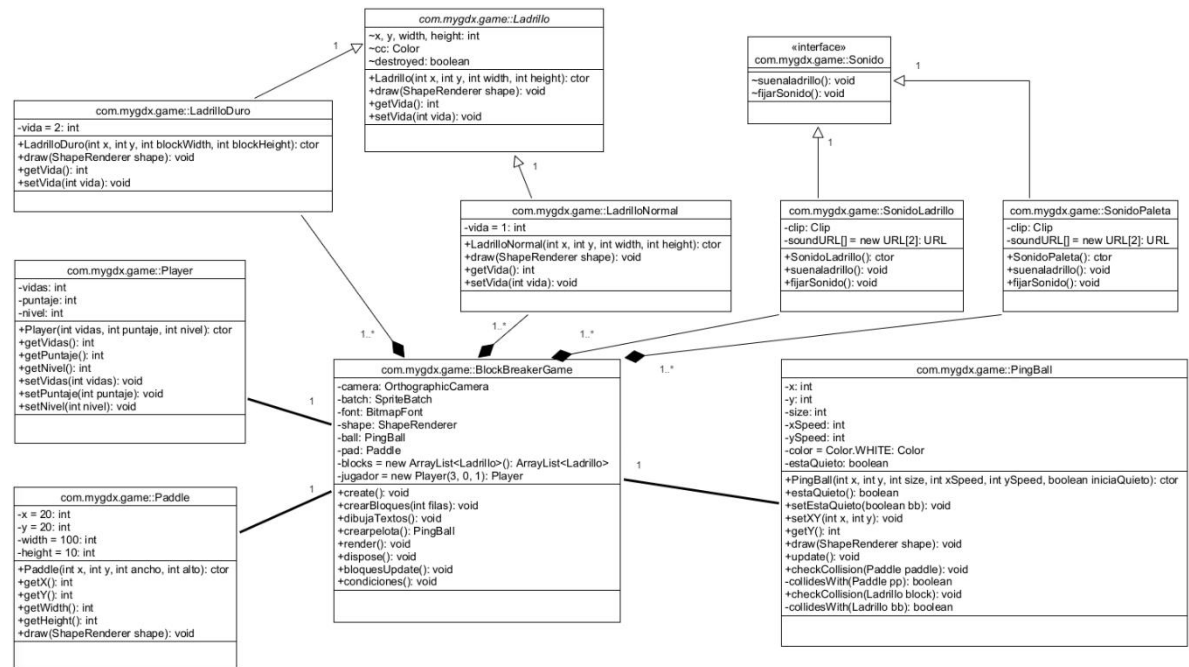
Bloque de nivel 2:



1.2 Análisis del juego a realizar (perspectiva del Ing. de software), especificando el juego a usar de base y las modificaciones que se le pretende realizar en términos de funcionalidad y estructura

Para el proyecto se utilizó de juego base el “BlockBreaker, las modificaciones visuales que se le quieren realizar son las explicadas en el punto 1.1, y si hablamos de funcionalidad y estructura el juego consta de dos niveles, el primero en el cual el jugador deberá romper tres filas de bloques, estos bloques solo tienen una vida, es decir cuando el shuriken colisiona con el bloque este se elimina. Para pasar al segundo nivel, el jugador deberá romper todos los bloques del nivel 1. En el nivel dos nos encontramos con dos tipos de bloques, distribuidos en cuatro filas. El primer tipo es similar a los bloques del nivel 1 y el segundo tipo de bloque cuenta con dos vidas, es decir, el jugador necesita colisionar dos veces con este para poder eliminarlo. Los bloques del tipo 2, se encuentran en las filas par.

1.3 Diseño de diagrama UML con clases del Dominio del juego (clases base LibGDX + propias) y su código en Java

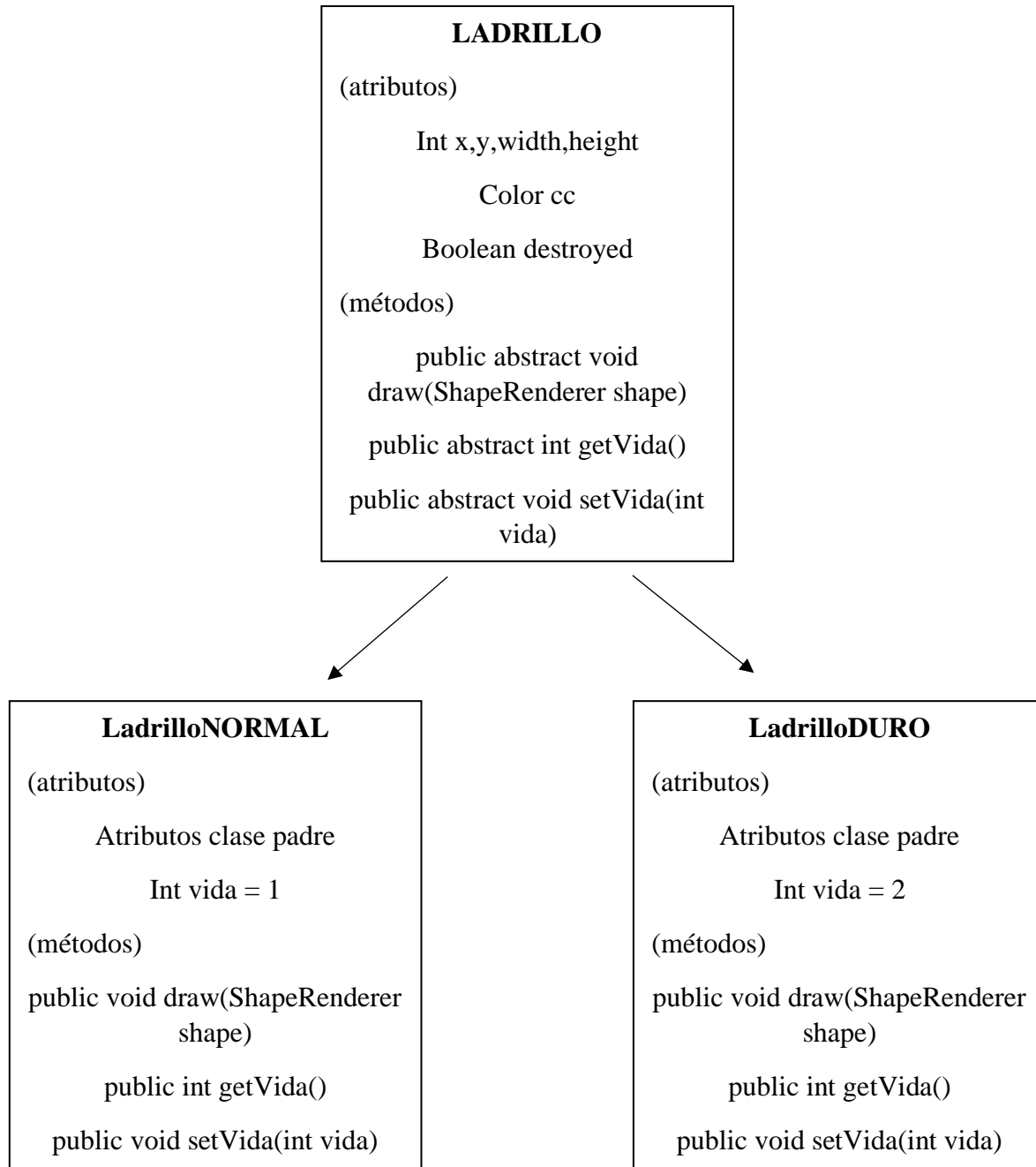


La función que crea los ladrillos usa 1 o más tipos de ladrillos.

La función que crea los sonidos usa 1 o más tipos de sonido.

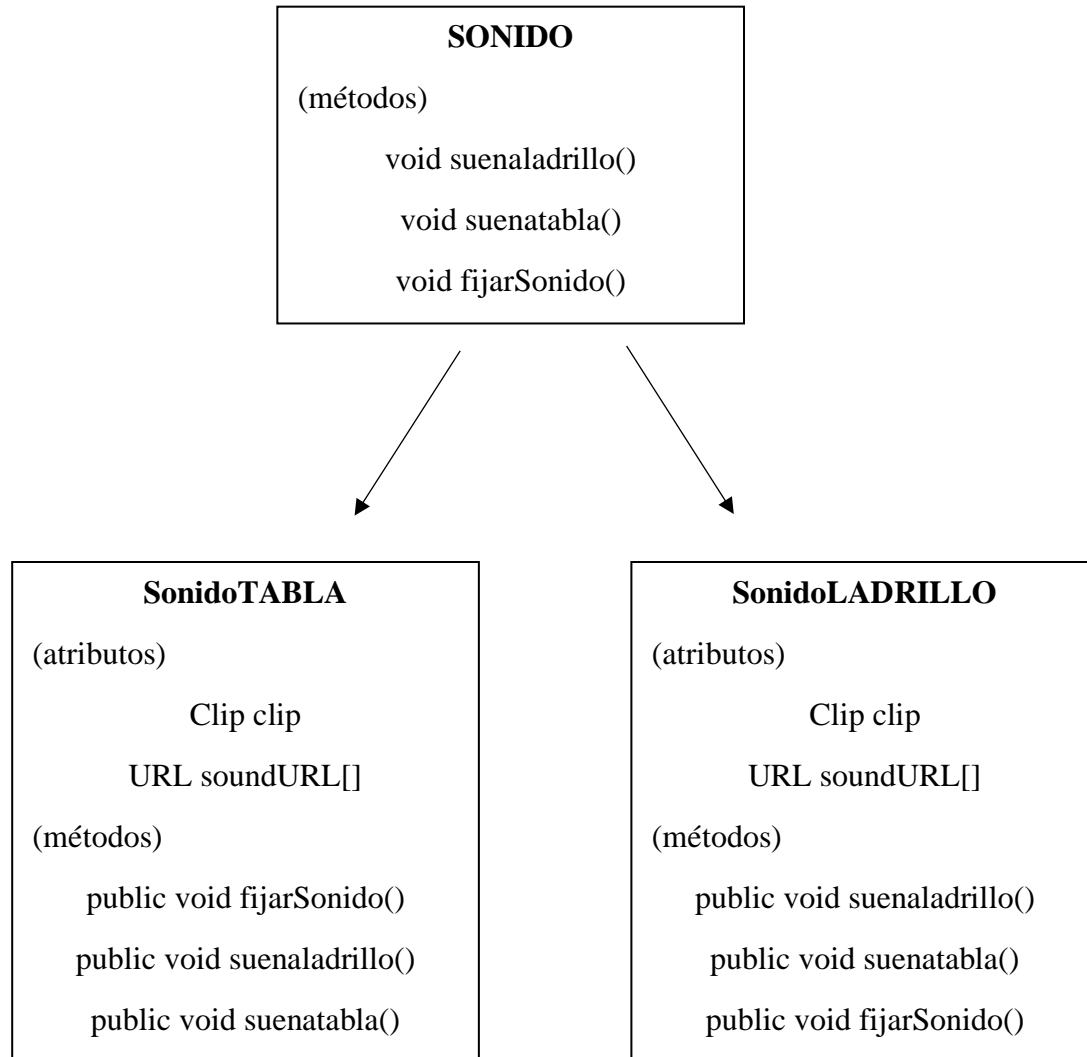
En el juego solo existe un paddle, una pelota.

1.4 Diseño y codificación de 1 (una) clase abstracta que sea padre de al menos 2 (dos) clases. La clase abstracta debe ser utilizada por alguna otra clase (contexto)



Se decidió crear una clase “padre” llamada “Ladrillo”, ya que dentro del juego existen dos tipos de bloques, los de una vida o los de dos vidas. Las clases “hijas” se utilizan en la clase BlockBreakerGame, y se llaman para crear ladrillos de una o dos vidas según el nivel que se encuentre el jugador.

1.5 Diseño y codificación de 1 (una) interfaz que sea implementada por al menos 2 (dos) clases. La interfaz debe ser utilizada por alguna otra clase (contexto)



La interfaz que se decidió implementar. Implementa dos clases, la primera “sonidoTabla” en la cual se le agrega sonido, y este se activa cada vez que rebota el shuriken (pelota) sobre el ninja (paddle). Y la segunda clase “SonidoLadrillo” en la cual se le agrega sonido, y se activa cada vez que el shuriken (pelota) colisiona y destruye un bloque.

Se puede apreciar que existe un arreglo y se implementa porque en la siguiente entrega se quiere tener un sonido para cuando el ladrillo se destruya y otro sonido para avisar que los ladrilloDuros ya fueron golpeados una vez.

1.6 Se debe aplicar encapsulamiento y principios OO

En el código se aplica el encapsulamiento y los principios OO correspondientes.

1.7 Utilización de GitHub (Realización de al menos 3 Commit)

Como se puede visualizar en el historial del repositorio se hicieron los commits correspondientes.