

Pinball Chino SW Specification

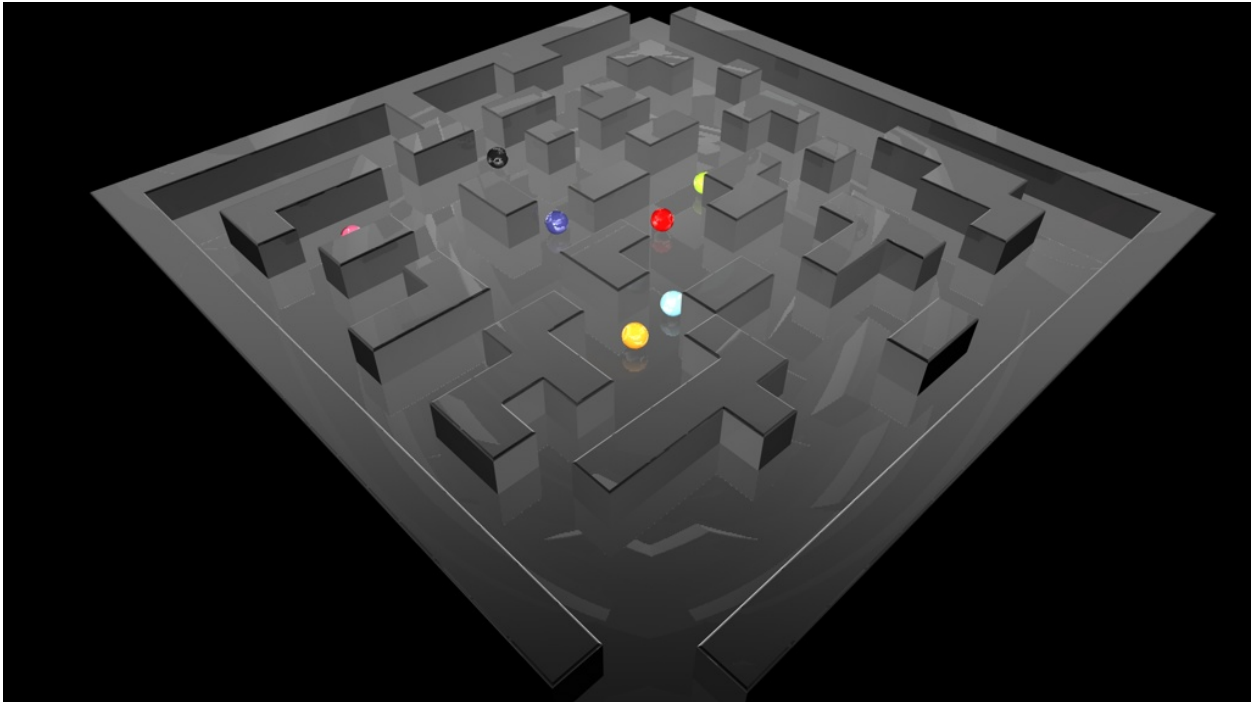


Table of Content

1. Overview

1.1 Game Play (Functional Description)

1.2 Software Architecture

1.3 Addons

1.3.1 ofxBullet

1.3.2 ofxOsc

1.3.3 ofxStateMachine

2. Detailed Description

2.1 CStateManager

2.1.1 Overview

2.2 CTable

1. Overview

1.1 Game Play (Functional Description)

Sesión brainstorm:

- Jugar por relevos;
- Escenario más grande de lo que se ve en pantalla (Sonic style);
- Más de 2 palancas y objetos con los que se puede interactuar;
- Paredes mapeadas de una manera que se cambia el centro de gravedad cuando cambias la superficie en la que estás jugando (mapping de un edificio por ejemplo);

1.2 Software Architecture

un paquete de clases que se va a encargar de gestionar la tabla y los tipos de objetos que están en la tabla, un paquete de clases que gestiona la bola, uno que gestiona los sonidos y envía eventos osc para cada sonido, un state manager que calcula el score decide si has ganado o no, cuando hay que reiniciar etc

También podemos tener clases separadas que se encarga de gestionar las texturas para objetos de tipo tabla o de tipo bola

Un objeto que es la tabla tiene que tener propiedades para marcar el score que lo haces cuando lo chocas y un identificador para el sonido que se activa (o también lo podemos llamar un identificador de evento que también lo vamos a signalizar por OSC)

La clase gameStatus controla el estado del juego, número de puntos, vidas, stage en el que estás y el escenario correspondiente.

UML link:

https://docs.google.com/drawings/d/1YPKONcwK_l8SZdBgoi6M-6fqD5wV9zCRYOWxooUB5D8/edit

1.3 Addons

1.3.1 ofxBullet

<https://github.com/NickHardeman/ofxBullet>

1.3.2 ofxOsc

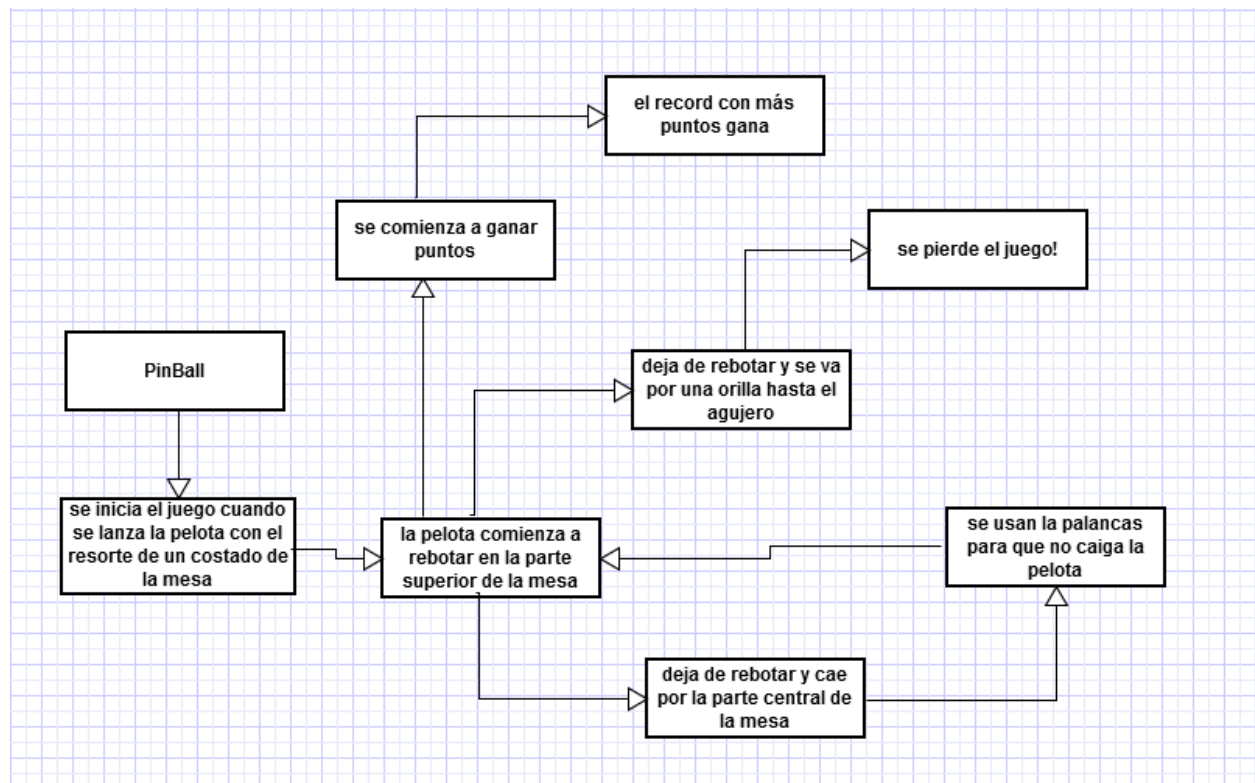
1.3.3 ofxStateMachine

<https://github.com/neilmendoza/ofxStateMachine>

2. Detailed Description

2.1 CStateManager

2.1.1 Overview



<https://github.com/neilmendoza/ofxStateMachine>

2.1.2 Public Interface

Members, methods etc

2.2 CTable

etc

Definicion del pinball Chino:

Idea:Pinball en 3d que se mapea en un espacio fisico. () . Partes reales y partes proyectadas. Ejemplos: como que entra la bola por una ventana y se aparece por otra. Accionar objetos dentro del mapping. Eventos especiales etc.

Brief pinball: pinball básico, se compone: Un escenario (estatico) + Bola + Palancas + objetos del juego + puntacion + maquina de estados

Tipo de escenario

Subdivisiones del pinball:

A) Escenario

Estático por toda la duración de un stage del juego, interactúa con la pelota por el contacto físico con la misma rebotándola cuando se choca. Puede ser modelado por su estructura de primitivas.

B) Objetos del juego:

Propiedades generales: ID , posicion, rotacion, centroide, propiedades físicas del objeto (Friccion , Damping, ...), tipos, Estado de colision (bcolision), Estado de reproduccion (bpl)

Tipos:

- Objetos estáticos que al ser golpeados en uno de sus laterales, envían puntos básicos o especiales (dependiendo del estado del juego).
- Túneles que miden cuánto % ha avanzado la bola en ellos. Envían puntos y + eventos especiales.
- ResorteBasico: objeto estaticos que ademas de sumar puntos, aplican fuerzas específicas sobre la bola.
- Martillo (Hammer) Objeto visible o no , para impulsar determinadamente otro objeto dinamico.
- Glass: Objeto que puede guardar en el la boa durante un tiempo, y que entonces llama a hammer para impulsar la bola de nuevo.

C) Palancas:

Las palancas son un objeto muy básico (tipo Dinamic) que aplica una fuerza específica sobre la bola, que depende del motor de fisicas (las fisicas y de la geometria de los objetos) y controlado por el jugador

D) Animaciones y sonidos:

Las animaciones del juego son imagenes o videos o sonidos, que se reproducen en diferentes estados y de diferente forma:

Propiedades generales de las animaciones: Tipo (sonido, imagen, video), estado de reproduccion (bactive: true, false),

Tipos

- Las mas simples: Se disparan varias a la vez.
- Las de los puntos: Se disparan singularmente y tienen mayor prioridad. Se reproducen por encima de las otras. (Esto es todos los objetos que tienen eventos, deben de tener un sistema de prioridad para activar o omitir las animaciones)

-

E) Maquinas de estados:

Controla el flujo del juego. En el estado inicial la pelota es puesta en juego y se empieza a sumar puntos, de acuerdo con los eventos disparados por el jugador o las interacciones de los objetos (pelota cae y se pierde puntos, empezando el escenario otra vez)