ITESM Campus Toluca
Benjamín Ávila Rosas
A01361843

Virtual Juggs Machine

DESCRIPCIÓN:
Una juggs machine es una maquina lanza-balones de football americano la cual utilizamos diario en la práctica de nuestro deporte para mejor nuestra habilidad de atrapar el balón, esta tiene una fuerza y dirección ajustable según el ejercicio que se busque realizar.

El juego Virtual Juggs Machine consiste en simular que estás utilizando una Juggs Machine verdadera, la máquina te estará lanzando balones con dirección y fuerza aleatoria desde el fondo del escenario y los balones vendrán acercándose hacia ti con la intención de que los “atrapes” haciendo clicks sobre de ellos.

A un balón no atrapado o que lo intentas atrapar y lo sueltas le llamamos “Drop”, el juego consiste en atrapar el mayor número de balones posibles sin llegar a tener 5 Drops, en cuanto alcances la marca de los 5 Drops tu juego se termina.

REGLAS:

- Atrapa la mayor cantidad de balones que puedas haciendo click con tu mouse sobre de ellos cuando estén lo suficientemente cerca de ti.
- Mueve el mouse para mover las manos del receptor hacia el punto deseado
- Al llegar a 5 drops tu juego se termina
- P para poner Pausa
- ESC para salir del juego
- CLICK en cualquier parte de la pantalla para comenzar a jugar

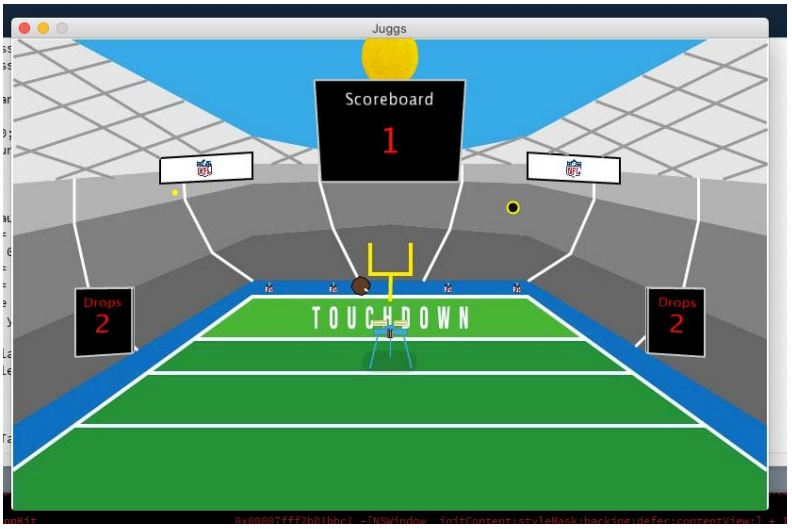
¿Cómo se hizo?

OpenGL

Cómo motor de optimización de gráficos 3D usé el la versión default de OpenGL que maneja processing en su versión 3.5.4

Escenarios

Para diseñar el escenario me apoyé del software de diseño vectorial Adobe Illustrator, con este cree el estadio y prácticamente todos los elementos visuales, a excepción de los marcadores que son cajas 3D que simulan un marcador de estadio.



Texturas

Utilicé una textura de una imagen de arena de mar para darle un color más natural al sol que se encuentra en el espacio sin techo del estadio.

```
sun = loadImage("sun.png"); // Used for texture
```

```
3
4  // Simulates sun with a texture, it turns fully yellow when a catch is performed
5  void sun() {
6    PShape s;
7
8    pushMatrix();
9    translate(width/2,20);
0
1    s = createShape(SPHERE, 30);
2    s.setTexture(sun);
3    shape(s, 0, 0);
4
5    popMatrix();
6  }
7
```

Iluminación

Para los flashes de los aficionados utilicé luces puntuales que se generan en un rango de posiciones aleatorias de forma cíclica para que aparente la sensación de un estadio lleno


```
  fill(255);
  stroke(#FFFF00);
  strokeWeight(2);
  lightFalloff(1.0, 0.001, 0.0); // Lights of flashes from crowd
  pointLight(255, 255, 0, 35, 40, 36);
  ellipse(x, y, 12, 12);
}
```

Audio

De igual forma, para simular la sensación de un estadio lleno, utilicé un audio del estadio U.S. Bank en Minnesota en pleno partido y lo ciclé para siempre tener de fondo ese ruido de tribuna.

```
  // Used for crowd noise
  crowd = new SoundFile(this, "cr.mp3");
  crowd.loop();
}
```

Animaciones

El juego tiene solamente 2 elementos que están animados, las manos del receptor se abren y cierran al atrapar el balón y el balón va rotando conforme va avanzando gracias al torque generado en un lanzamiento.

1º Ley de Newton

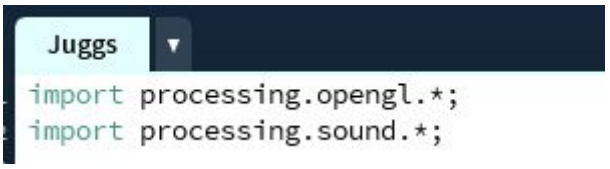
Decidí implementar la Ley de inercia de Newton ya que en esta se declara que V = d / t, por lo tanto la velocidad con la que el balón se aproxima al jugador se debe a la distancia recorrida en cierto tiempo, la distancia se logra obtener gracias a un escalamiento dinámico del objeto y que tiene un tiempo límite para llegar hasta las manos del receptor. Ver uso de variables globales “speed” y “timer”.

```
  translate(equis, ye);

  rotate(angle);
  scale(zoom);

  angle += 0.3; // Rotate ball to simulates throw's torque
  if (zoom <= 0.8) {
    zoom += 0.005 * speed; // Controls ball growth and proximity
  }
}
```

```
if (timer <= maxTime) {
  ball(display);
  timer += speed;
```

Librerías

- processing.audio.*;
- processing.opengl.*;


Juggs
import processing.opengl.*;
import processing.sound.*;