

FÍSICA EN VIDEOJUEGOS 2D

Física en Videojuegos 2D

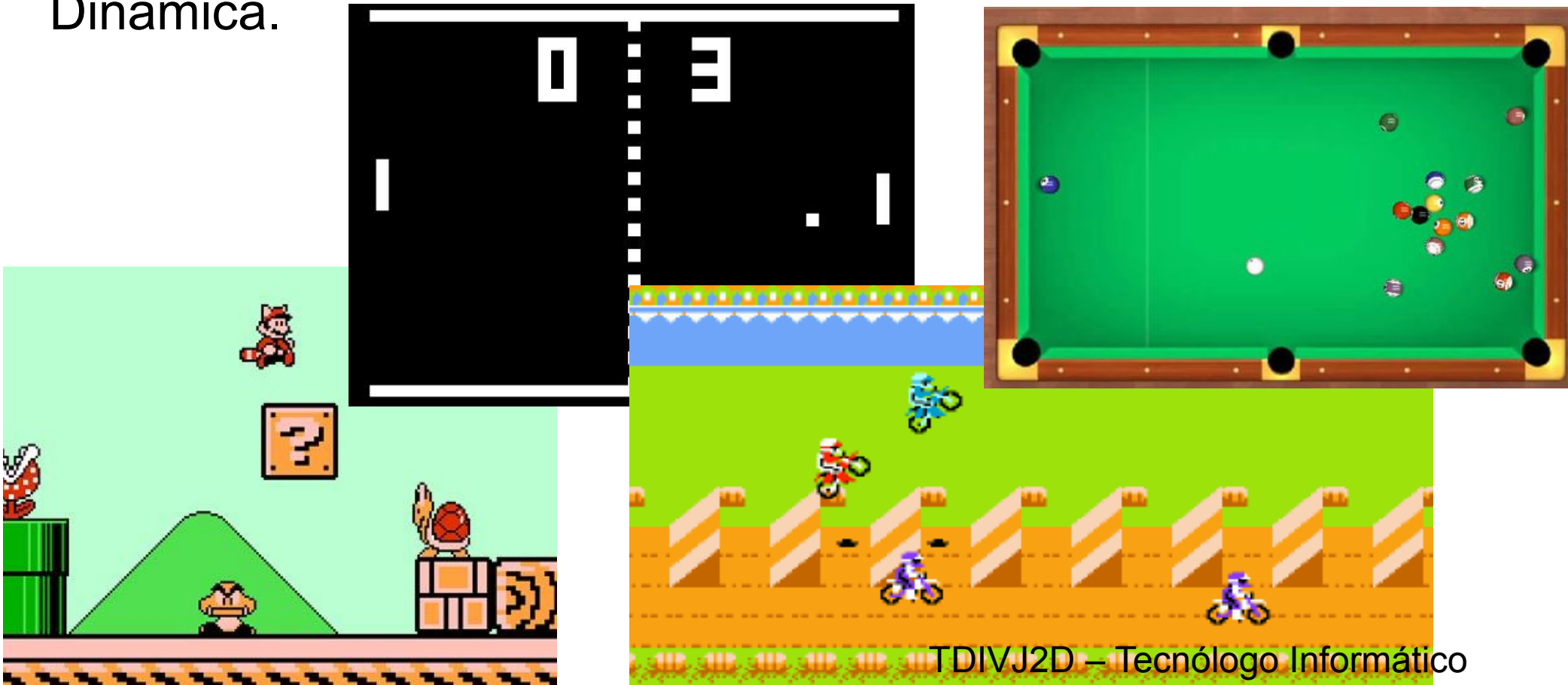
TEMARIO

- 1. Introducción**
2. Movimiento y velocidad
3. Suma de vectores
4. Colisiones
5. Gravedad
6. Masa y rozamiento
7. Otras Fuerzas y conceptos
8. Motores de física

1. Introducción

Para lograr una mejor experiencia los videojuegos introducen simulaciones físicas para alcanzar un nivel más realista.

Las áreas de la física comprendidas son la Cinemática y la Dinámica.



1. Introducción

Cinemática: Estudia las leyes del movimiento de los cuerpos sin tener en cuenta las causas que lo producen.

Dinámica: Estudia el movimiento de los objetos y de su respuesta a las fuerzas.

1. Introducción

Leyes de Newton (dinámica):

- 1 – Ley de Inercia
- 2 – Ley de Fuerza
- 3 – Ley de Acción y Reacción
- 4 – Ley de gravitación universal



Física en Videojuegos 2D

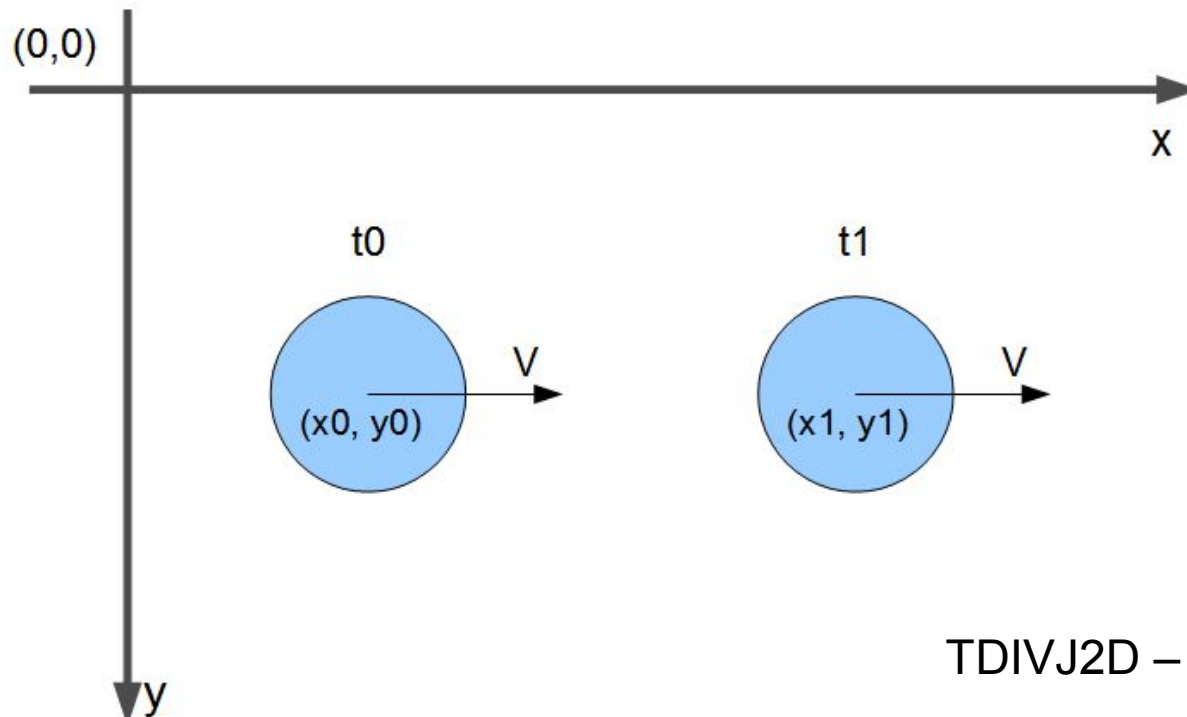
TEMARIO

1. Introducción
- 2. Movimiento y velocidad**
3. Suma de vectores
4. Colisiones
5. Gravedad
6. Masa y rozamiento
7. Otras Fuerzas y conceptos
8. Motores de física

2. Movimiento y velocidad

El movimiento es el cambio de posición de un cuerpo, se da de forma continua con el paso del tiempo.

Para simular el movimiento debemos modelar la dirección y sentido del movimiento y la velocidad del mismo. A partir de esto calcular la nueva posición del cuerpo en el instante de tiempo a dibujar.



2. Movimiento y velocidad

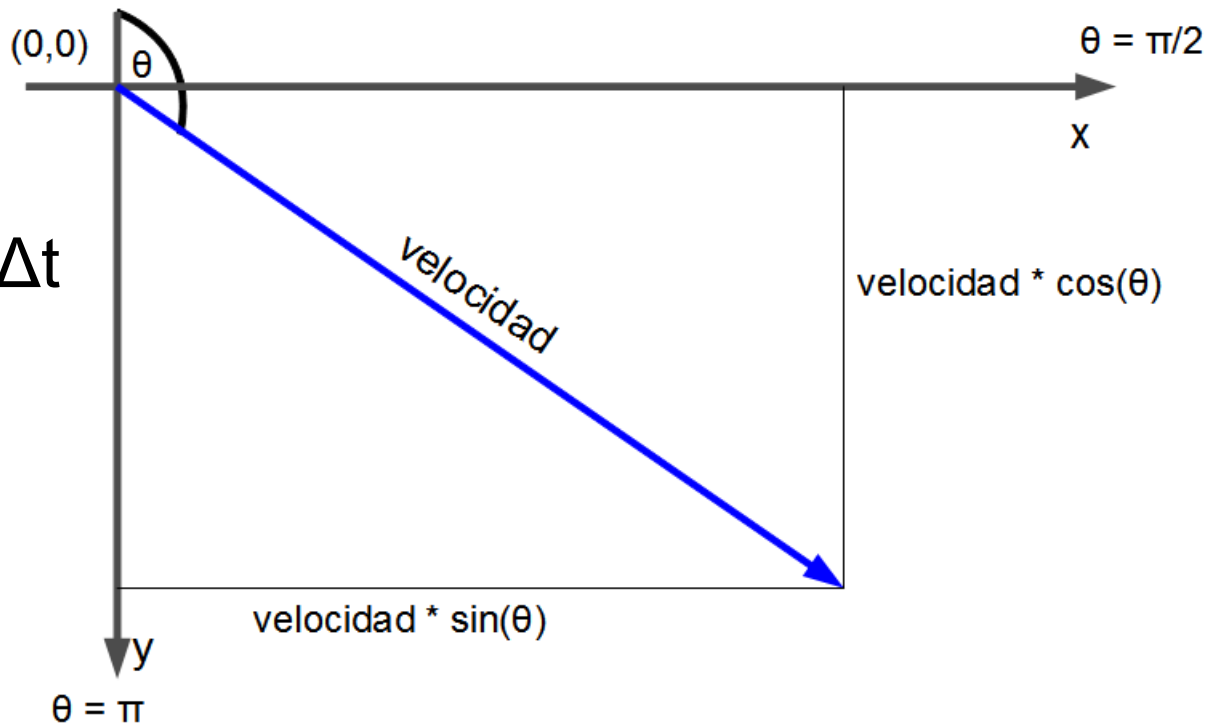
El vector de movimiento puede representarse mediante las coordenadas o mediante un ángulo y una magnitud.

$$V = (v_x, v_y)$$

ó

$$V = (\theta, \text{velocidad})$$

$$\text{velocidad} = \text{distancia} / \Delta t$$



2. Movimiento y velocidad

Calculando la nueva posición de un elemento en movimiento.

Conocidos: t_0 , x_0 , y_0 , V (θ , velocidad)

Incognitas: x_1 , y_1

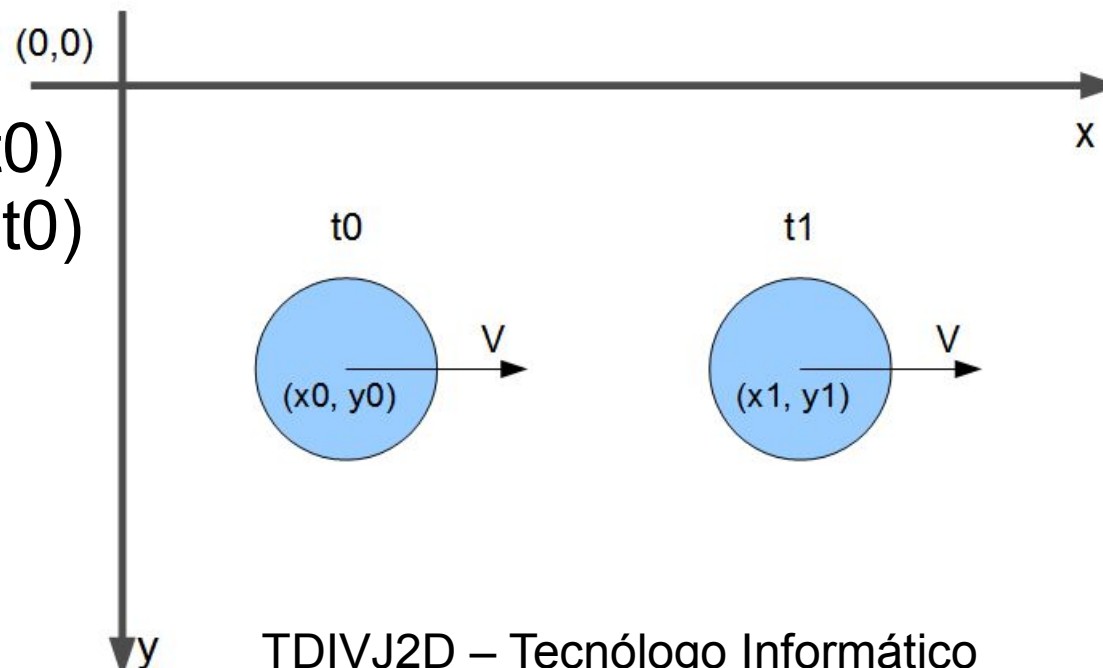
$\text{distancia} = \text{velocidad} * \Delta t$

$$\Delta x = \text{velocidad} * \sin(\theta) * (t_1 - t_0)$$

$$\Delta y = \text{velocidad} * \cos(\theta) * (t_1 - t_0)$$

$$x_1 = x_0 + \Delta x$$

$$y_1 = y_0 + \Delta y$$



2. Movimiento y velocidad

- Se debe tener cuidado con las unidades de tiempo con que se trabaja.
 - Si por ejemplo se asume que el tiempo avanza de a una unidad en cada ciclo del main loop se está ligando el tiempo con el frame rate.
- Los ángulos por lo general se miden en radianes. Si se quiere trabajar con grados debe tenerse en cuenta que $1^\circ = 180/\pi$ radianes.

2.1 Aceleración

- La aceleración de un elemento indica que tan rápido este ganará velocidad.
- Es la consecuencia de la aplicación de una fuerza a un cuerpo.

$a = \text{Fuerza} / \text{masa}$

$$\Delta v = a * \Delta t$$



Física en Videojuegos 2D

TEMARIO

1. Introducción
2. Movimiento y velocidad
- 3. Suma de vectores**
4. Colisiones
5. Gravedad
6. Masa y rozamiento
7. Otras Fuerzas y conceptos
8. Motores de física

3. Suma de vectores

$$V3 = V1 + V2$$

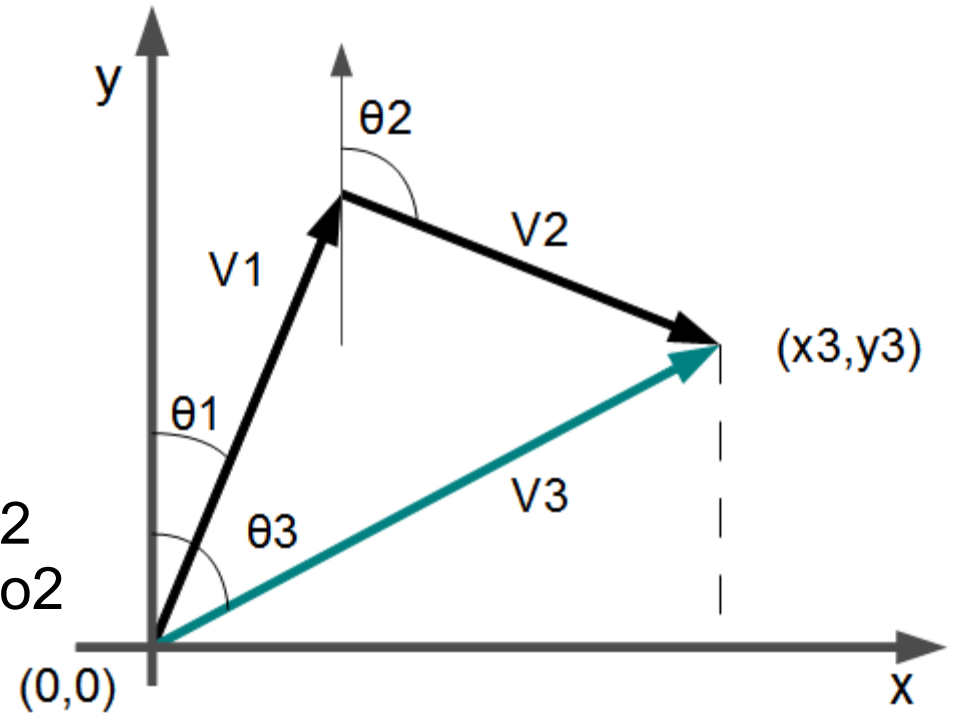
$$V1 = (\theta1, \text{largo1})$$

$$V2 = (\theta2, \text{largo2})$$

$$V3 = (\theta3, \text{largo3})$$

$$x3 = \sin(\theta1) * \text{largo1} + \sin(\theta2) * \text{largo2}$$

$$y3 = \cos(\theta1) * \text{largo1} + \cos(\theta2) * \text{largo2}$$

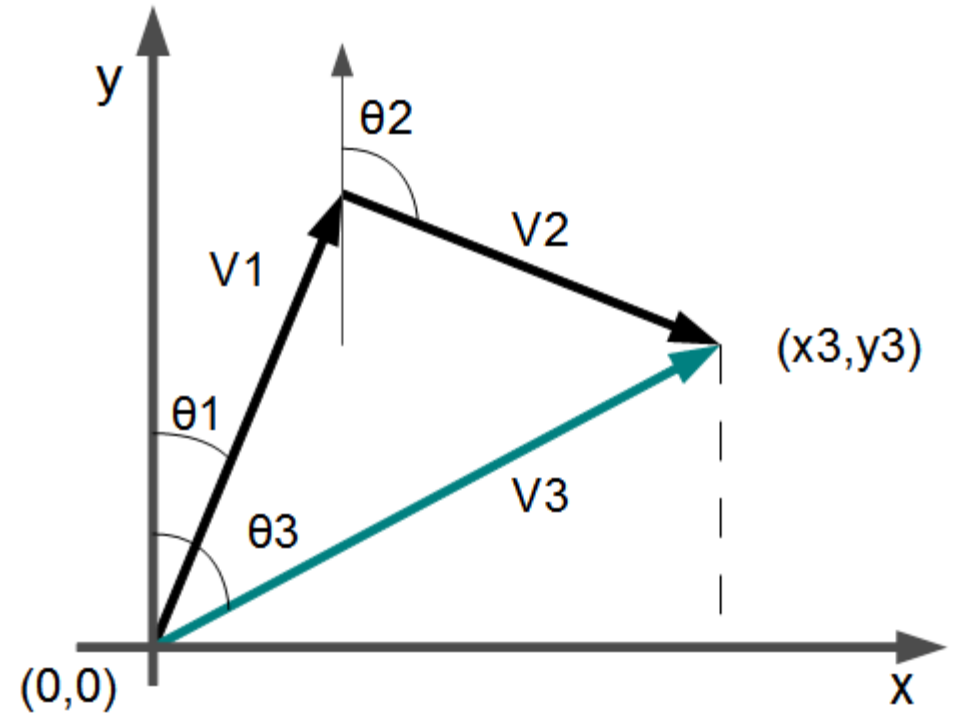


3. Suma de vectores

$V_3 = (x_3, y_3)$ ó $V_3 = (\theta_3, \text{largo}_3)$

$\theta_3 = \pi/2 - \text{arcotangente}(y_3/x_3)$

$\text{largo}_3 = \text{raiz}(x_3^2 + y_3^2)$



Física en Videojuegos 2D

TEMARIO

1. Introducción
2. Movimiento y velocidad
3. Suma de vectores
- 4. Colisiones**
5. Gravedad
6. Masa y rozamiento
7. Otras Fuerzas y conceptos
8. Motores de física

4. Colisiones

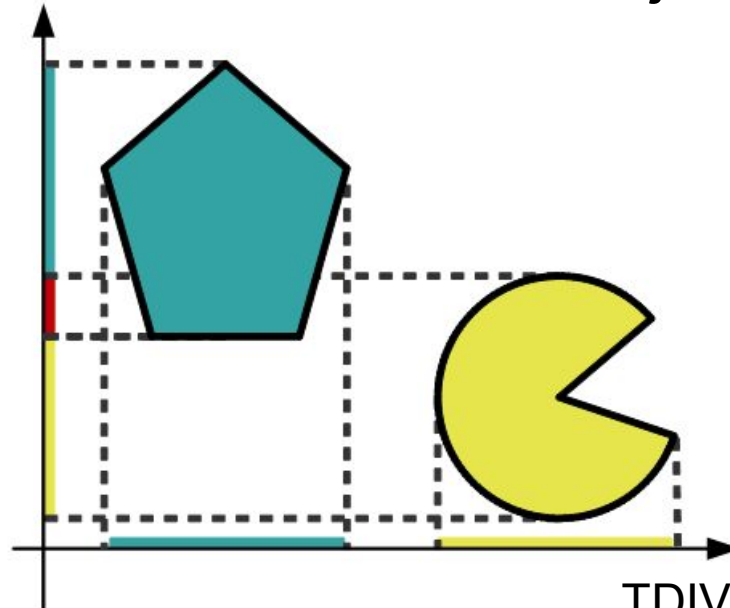
Pasos:

- Detectar colisiones
- Resolución de colisiones
- Evitar solapamientos

4.1 Detección de colisiones

Detección de colisiones

- Por área
- Pixel perfect
- Teorema de ejes separados
 - Si están separados en algún eje entonces no colisionan
 - Si se superponen sobre todos los ejes colisionan



4.1 Detección de colisiones

Collision culling

- Aplicar test de colisiones con fuerza bruta con n objetos implica una cantidad de comparaciones del orden n^2 .
- Las técnicas de collision culling intentan de acotar el costo computacional y el número de comparaciones que debe hacerse para determinar si un objeto colisiona con otro en la escena.
- Las principales técnicas implican explotar la coherencia temporal o geométrica.

4.1 Detección de colisiones

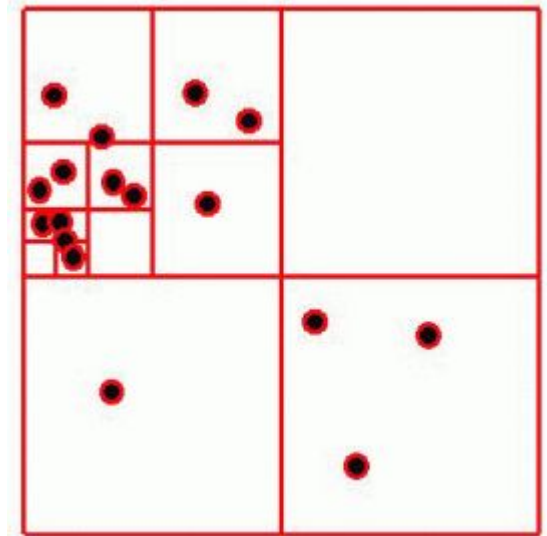
Coherencia temporal

- Supuesto de que se producen pequeños cambios entre frames.
- Esto hace que en cada frame se repitan valores de entrada y cálculos que de guardarse pueden reducir el tiempo de cálculo del próximo frame.

4.1 Detección de colisiones

Coherencia geométrica

- Puede definirse como el grado de separabilidad de los elementos de la escena.
- Las técnicas de cajas envolventes explotan la coherencia geométrica.
- Otras técnicas implican la división del escenario y sus elementos en una estructura jerárquica:
 - Grillas
 - Quadtree
 - Jerarquía de cajas envolventes



4.2 Resolución de colisiones

Resolución de colisiones

- Que hacer cuando se detecta una colisión.
- Información necesaria para resolver que hacer:
 - Punto de contacto.
 - Normal en el punto de contacto.
 - Distancia de penetración.

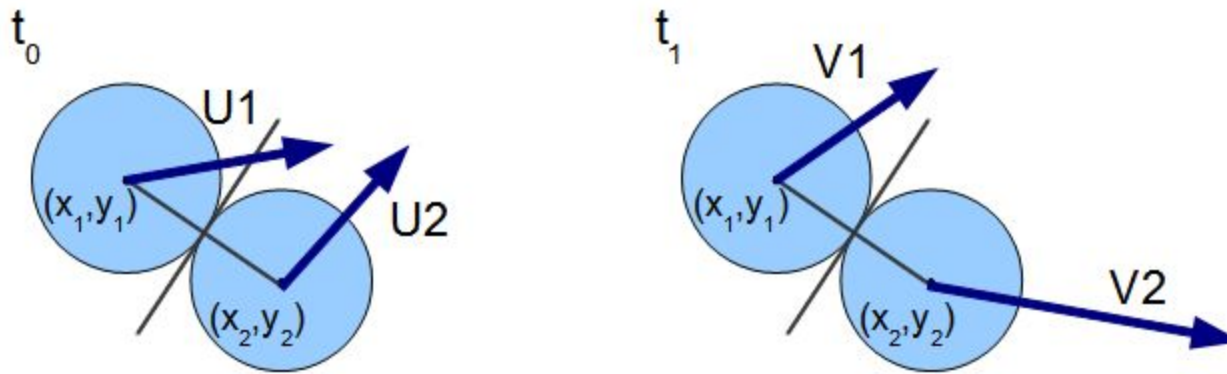
4.2 Resolución de colisiones

Colisiones (sin rotación)

- El impulso es aplicado en el instante de la colisión en la dirección de la normal de la colisión.
- Los cambios de velocidad están determinados por:
 - Conservación del momento lineal.
 - Coeficiente de restitución (elasticidad).

4.2 Resolución de colisiones

Conservación del momento lineal



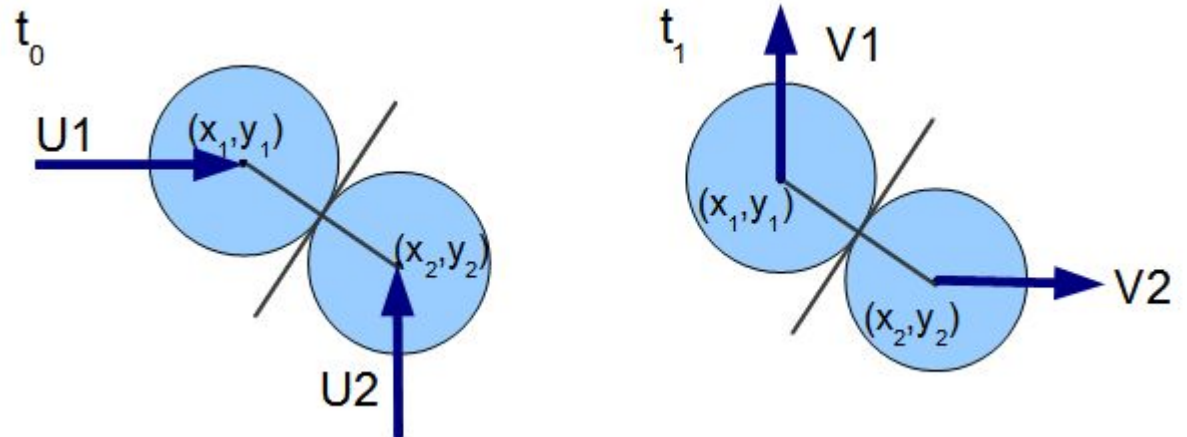
$$m_1 u_1 + m_2 u_2 = m_1 v_1 + m_2 v_2$$

4.2 Resolución de colisiones

Conservación del momento lineal

Asumiendo masas iguales para los elementos en colisión la magnitud de las velocidades son intercambiadas y el ángulo del vector de velocidad es reflejado según la normal al punto de contacto.

$$u_1 + u_2 = v_1 + v_2$$



4.2 Resolución de colisiones

Coeficiente de restitución.

- Cuando dos cuerpos chocan, sus materiales pueden comportarse de distinta manera. Hay materiales cuyas fuerzas se restituirán completamente, otros tipos cambian su forma, liberan calor, etc. modificándose la energía cinética total.
- Se define entonces un coeficiente de restitución que evalúa esta pérdida de energía cinética, según la naturaleza de los materiales.

4.2 Resolución de colisiones

Coeficiente de restitución.

$$K = (v_1 - v_2) / (u_1 - u_2)$$

K varía entre 0 y 1.

Si $K = 0$ choque perfectamente inelástico

Si $K = 1$ choque perfectamente elástico

4.3 Evitar solapamientos

Evitar solapamientos

- Si se calculasen las colisiones en el momento de contacto no sucederían los solapamientos.
- Los cálculos se hacen a intervalos de tiempo regulares (framerate), por lo que puede ser necesario “correr” los elementos para evitar solapamientos.
- Se debe elegir con cuidado la técnica para realizar las correcciones en la posición de los elementos, dado que pueden haber muchos objetos cercanos.
 - Intentar que la corrección no provoque solapamiento con otro elemento.
 - Si no es posible corregir la posición del objeto con el que se solapa luego de aplicar la corrección

Física en Videojuegos 2D

TEMARIO

1. Introducción
2. Movimiento y velocidad
3. Suma de vectores
4. Colisiones
- 5. Gravedad**
6. Masa y rozamiento
7. Otras Fuerzas y conceptos
8. Motores de física

5. Gravedad

- La gravedad es un efecto descrito por la ley de gravitación universal (4ta ley de Newton).
- La fuerza ejercida entre dos cuerpos separados cierta distancia es proporcional al producto de sus masas e inversamente proporcional al cuadrado de la distancia. Es decir que cuanto más masivos sean los cuerpos y más cercanos se encuentren, con mayor fuerza se atraerán.

$$F=(G * m1 * m2) / r^2$$

- La fuerza que ejerce la Tierra sobre los cuerpos es lo que denominamos fuerza de gravedad y dicha fuerza es la que origina la sensación de peso.

5. Gravedad

- Galileo predijo que dos cuerpos en el vacío en caída libre caen a la misma velocidad. Es decir que la gravedad afecta a los cuerpos de igual manera independientemente de la masa.
 - Esto se aprecia de esta manera porque la masa de estos es despreciable comparada con la de la Tierra.
- Para simular el efecto de la gravedad en videojuegos es suficiente con esta aproximación, definiendo un único vector de gravedad.

$$\mathbf{G} = (\pi, g)$$

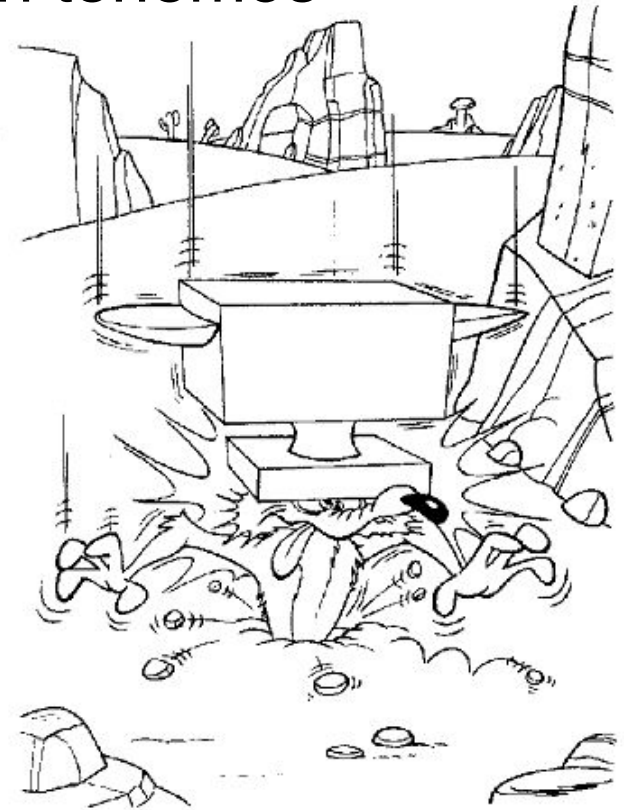


5. Gravedad

La gravedad modifica la velocidad de cada cuerpo al momento de actualizar su estado. Dado que la gravedad es una medida de aceleración tenemos que:

$$\Delta V = G * (t1 - t0)$$

$$V1 = V0 + \Delta V$$



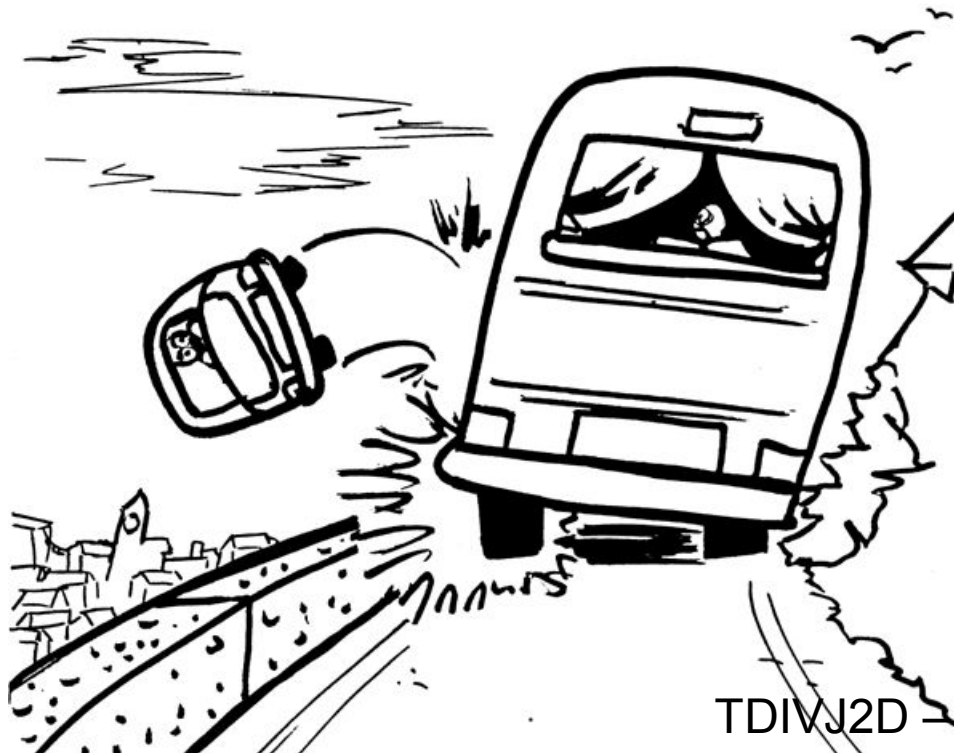
Física en Videojuegos 2D

TEMARIO

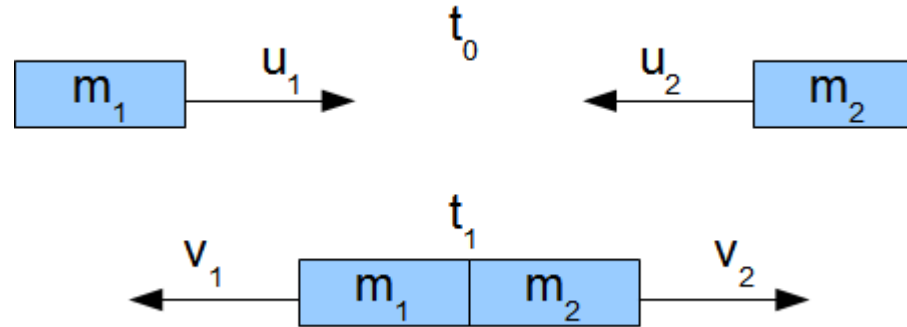
1. Introducción
2. Movimiento y velocidad
3. Suma de vectores
4. Colisiones
5. Gravedad
- 6. Masa y rozamiento**
7. Otras Fuerzas y conceptos
8. Motores de física

6. Masa y rozamiento

- Para darle más realismo a las simulaciones debe introducirse el concepto de masa, y con el, el de fuerza.
- Cuando dos cuerpos de diferente masa chocan, el cuerpo de menor masa sufrirá una perturbación mayor.



6. Masa y rozamiento



Principio de conservación del momento lineal

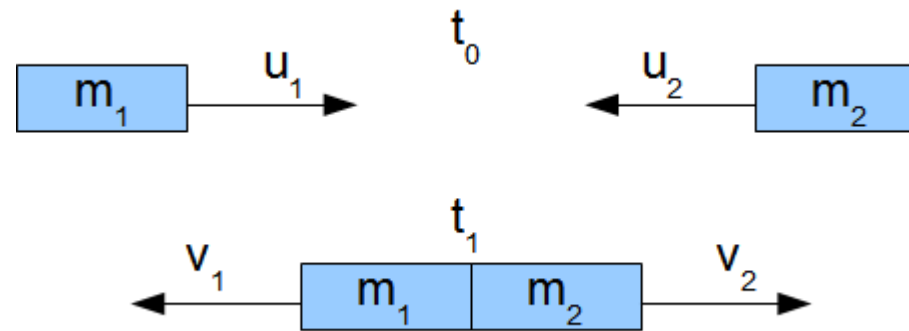
$$m_1 u_1 + m_2 u_2 = m_1 v_1 + m_2 v_2$$

En un choque elástico, la energía cinética inicial es igual a la final.

$$\frac{1}{2} m_1 u_1^2 + \frac{1}{2} m_2 u_2^2 = \frac{1}{2} m_1 v_1^2 + \frac{1}{2} m_2 v_2^2$$

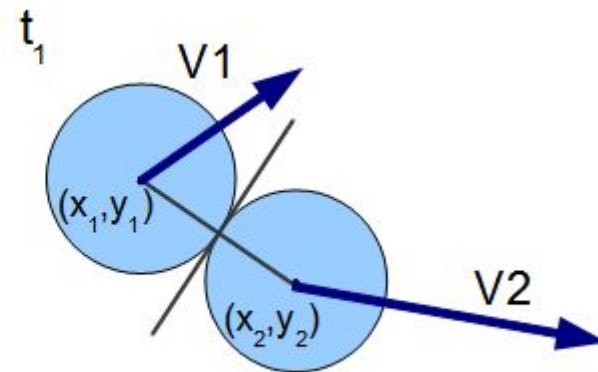
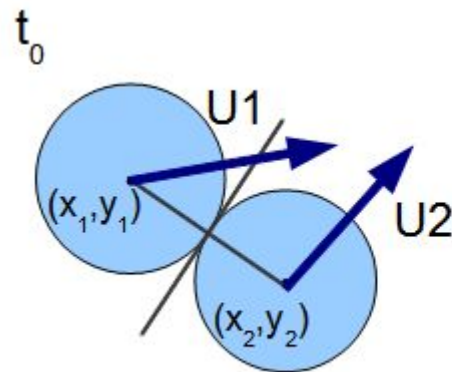
Dados m_1 , u_1 , m_2 y u_2 se despeja v_1 y v_2

6. Masa y rozamiento



$$v_1 = [2m_2u_2 + (m_1 - m_2)u_1] / (m_1 + m_2)$$

$$v_2 = [2m_1u_1 + (m_2 - m_1)u_2] / (m_1 + m_2)$$



6. Masa y rozamiento

Rozamiento con el medio:

- Con cuerpos de distinta masa podemos actualizar como son afectados por el rozamiento con el medio (comúnmente con el aire).
- Éste es producido por millones de partículas chocando contra el cuerpo, pero debido al costo computacional se modela mediante aproximaciones.

6. Masa y rozamiento

Rozamiento con el medio:

- La pérdida de velocidad del cuerpo al chocar con una partícula es proporcional a $1/(m_c + m_p)$
- El número de colisiones con partículas del medio en un intervalo de tiempo será proporcional a la distancia recorrida en dicho intervalo y del tamaño de la superficie que desplaza el medio. De esta manera obtenemos la siguiente fórmula para calcular la pérdida de velocidad:

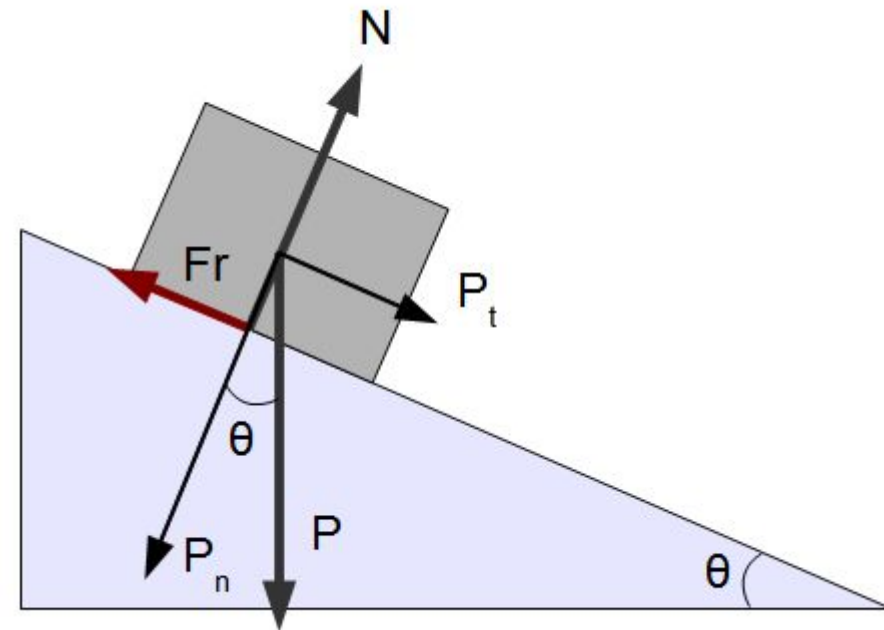
$$\text{vel_perdida} = v_c [1 / (m_c + m_p)] \Delta t \cdot \text{tamaño}$$

$$v_c = v_c - \text{vel_perdida}$$

6. Masa y rozamiento

Rozamiento estático:

- El rozamiento estático con la superficie es una fuerza.
- Es perpendicular a la normal en la superficie de contacto y opuesta al movimiento.
- $F_r = \mu P_t$ donde μ es la constante de rozamiento que depende del material de las superficies y $P_t = \sin(\theta)$.



Física en Videojuegos 2D

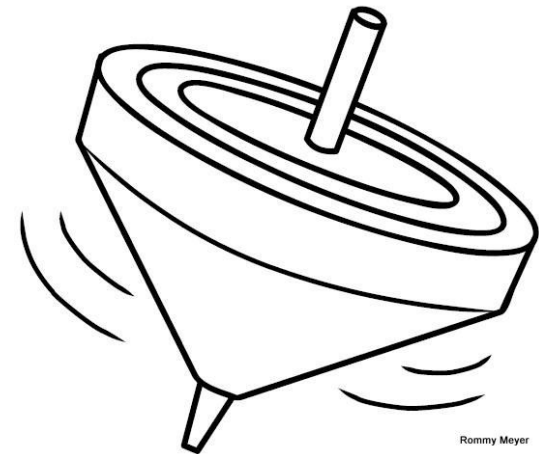
TEMARIO

1. Introducción
2. Movimiento y velocidad
3. Suma de vectores
4. Colisiones
5. Gravedad
6. Masa y rozamiento
- 7. Otras fuerzas y conceptos**
8. Motores de física

7. Otras Fuerzas y conceptos

Momento angular:

- Es la resistencia que ofrece un cuerpo a la variación de la velocidad angular. La velocidad angular es la velocidad de rotación respecto a un eje.
- De gran importancia para modelar colisiones con rotación. Para ello se aplican fórmulas de conservación de momento angular.



7. Otras Fuerzas y conceptos

Centro de masa:

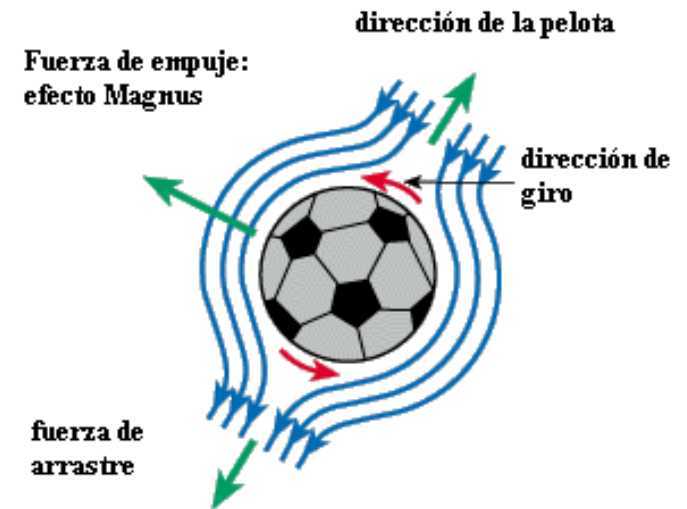
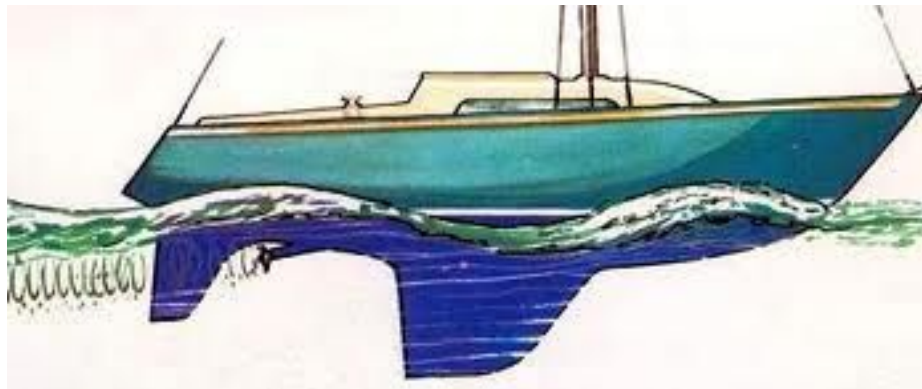
- El Centro de masa es el punto en el cual se puede considerar concentrada toda la masa de un objeto o de un sistema.
- Al aplicar fuerzas sobre el sistema es el punto que permanece en el centro de giro.
- Puede estar fuera del objeto.
- Muy útil para simplificar cálculos.



7. Otras Fuerzas y conceptos

Mecánica de fluidos:

- Rama de la física que estudia el movimiento de los fluidos y las fuerzas que provocan.
- A diferencia de los cuerpos rígidos no tienen forma dado que no resisten esfuerzos cortantes.



7. Otras Fuerzas y conceptos

Más:

- Atracción.
- Tensión.
- Sistemas de partículas.
- Articulaciones (joint)

Física en Videojuegos 2D

TEMARIO

1. Introducción
2. Movimiento y velocidad
3. Suma de vectores
4. Colisiones
5. Gravedad
6. Masa y rozamiento
7. Otras Fuerzas y conceptos
8. **Motores de física**

8. Motores de física

Chipmunk (C): <http://wiki.slembcke.net/main/published/Chipmunk>

Farseer (XNA) <http://www.codeplex.com/FarseerPhysics>

Bullet: <http://www.bulletphysics.com/>

Open Dynamics Engine: <http://www.ode.org/>

Havok: <http://www.havok.com/>

Physics2d.Net: <http://code.google.com/p/physics2d/>

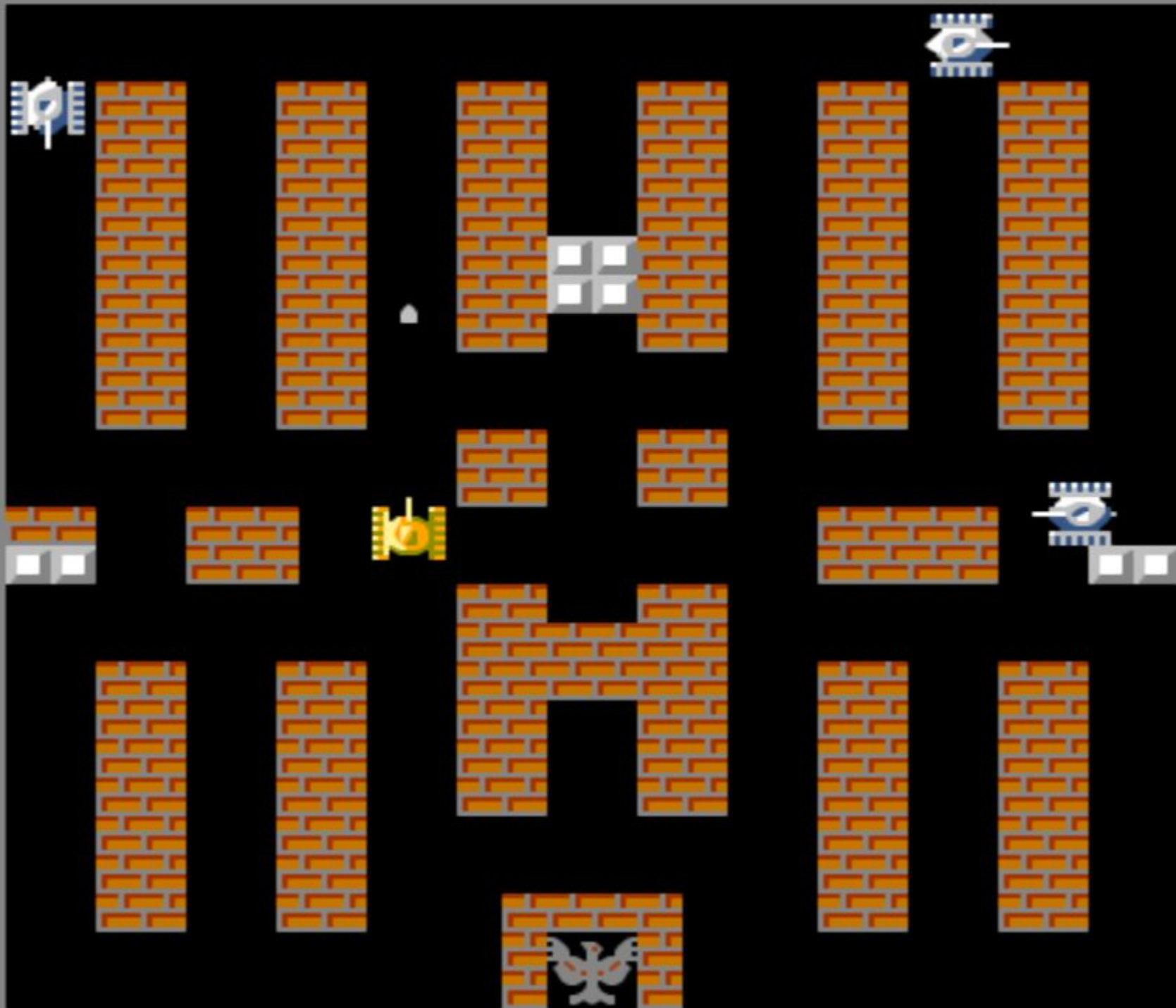
Box2D : <http://www.box2d.org/>

Pybox2d (Py): <http://code.google.com/p/pybox2d/>

Pymunk (Py): <http://code.google.com/p/pymunk/wiki/SlideAndPinJointsExample>

Referencias

- *Game Physics - David Eberly*
- *Game Physics Engine Development - Ian Millington*
- Física con ordenador: Curso Interactivo de Física en Internet
<http://www.sc.ehu.es/sbweb/fisica/>
- Box2D API Documentation 2.02 <http://programanddesign.com/box2d/>
- Box2D v2.2.0 User Manual: <http://www.box2d.org/manual.html>



IP
2

