

Overview

Chapter 1

Simulation #1

Simulation #2

Simulation #3

Chapter 2

Simulation #4

Simulation #5

Simulation #6

Chapter 3

Simulation #3

Conclusion

# Mechanics Simulations With JavaScript

Peter Krieg

Physics Fall Semester Thesis

December 3, 2014

# Overview - Why Did I Choose This Topic?

## Overview

### Chapter 1

Simulation #1  
Simulation #2  
Simulation #3

### Chapter 2

Simulation #4  
Simulation #5  
Simulation #6

### Chapter 3

Simulation #3

### Conclusion

- Use programming as a lens to view physics
- Examine mechanics in more detail
- Solve physics problems through simulations
- JavaScript high level language - viewable easily in web browser

# What is a simulation?

## Overview

### Chapter 1

Simulation #1  
Simulation #2  
Simulation #3

### Chapter 2

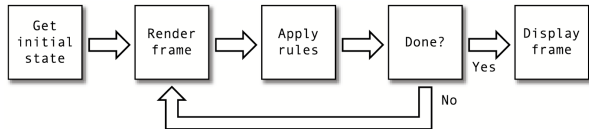
Simulation #4  
Simulation #5  
Simulation #6

### Chapter 3

Simulation #3

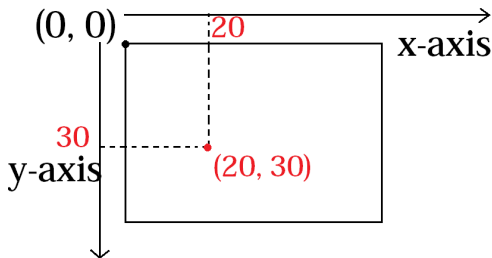
### Conclusion

- Animation vs. Simulation
- Frames per second
- File size



# Method of Basic Simulation

- HTML5 canvas application programming interface (API)
- Timer for each frame



Overview

Chapter 1

Simulation #1

Simulation #2

Simulation #3

Chapter 2

Simulation #4

Simulation #5

Simulation #6

Chapter 3

Simulation #3

Conclusion

# Chapter 1: Basic Kinematics and Aerodynamic Drag

Overview

Chapter 1

Simulation #1

Simulation #2

Simulation #3

Chapter 2

Simulation #4

Simulation #5

Simulation #6

Chapter 3

Simulation #3

Conclusion

- Three simulations
- Simulation #1: Basic bouncing ball
- Simulation #2: Bouncing ball with aerodynamic drag
- Simulation #3: Multiple bouncing balls

# Simulation #1: Basic Bouncing Ball

Overview

Chapter 1

Simulation #1

Simulation #2

Simulation #3

Chapter 2

Simulation #4

Simulation #5

Simulation #6

Chapter 3

Simulation #3

Conclusion

- Realistic g value
- $9.81 \frac{px}{s^2} = .1635 \frac{\frac{px}{s}}{frame} \times \frac{60frame}{s}$
- Coefficient of restitution ( $C_r$ )
- $C_r = \sqrt{\frac{KE_f}{KE_i}} = \sqrt{\frac{\frac{1}{2}mv_f^2}{\frac{1}{2}mv_i^2}} = \frac{v_f}{v_i}$
- $v_f = v_i * C_r$

# Simulation #2: Bouncing Ball With Aerodynamic Drag

Overview

Chapter 1

Simulation #1

**Simulation #2**

Simulation #3

Chapter 2

Simulation #4

Simulation #5

Simulation #6

Chapter 3

Simulation #3

Conclusion

- $f_d = -\frac{1}{2}C_d\rho Av^2$
- $f_d$  = force of drag
- $\rho$  = density of fluid
- $v$  = speed of object relative to fluid
- $C_d$  = drag coefficient (affected by texture, shape, viscosity, lift, etc)
- $A$  = cross-sectional area of object

# Simulation #3: Multiple Balls Bouncing

Overview

Chapter 1

Simulation #1

Simulation #2

**Simulation #3**

Chapter 2

Simulation #4

Simulation #5

Simulation #6

Chapter 3

Simulation #3

Conclusion

- Same physics as simulation #1
- Array of ball objects
- Each object has properties
- Each frame cycles through array, updating properties of each object



# Chapter 2: Planetary Motion

Overview

Chapter 1

Simulation #1

Simulation #2

Simulation #3

Chapter 2

Simulation #4

Simulation #5

Simulation #6

Chapter 3

Simulation #3

Conclusion

- 3 Simulations
- Simulation #4: Orbits
- Simulation #5: Escape velocity
- Simulation #6: Kepler's 2nd law

# Simulation #4: Orbits

Overview

Chapter 1

Simulation #1

Simulation #2

Simulation #3

Chapter 2

Simulation #4

Simulation #5

Simulation #6

Chapter 3

Simulation #3

Conclusion

- Newton's Law of universal gravitation
- $F_g = G \frac{m_1 m_2}{r^2}$
- Euler's Method to update velocity
- $x(t + dt) = x(t) + \frac{dx}{dt}(t) dt$

# Simulation #5: Escape Velocity

Overview

Chapter 1

Simulation #1

Simulation #2

Simulation #3

Chapter 2

Simulation #4

**Simulation #5**

Simulation #6

Chapter 3

Simulation #3

Conclusion

- $K_i + U_{g_i} = K_f + U_{g_f}$
- $\frac{1}{2}mv_{esc}^2 - \frac{GMm}{r} = 0 + 0$
- $v_{esc} = \sqrt{\frac{2GM}{r}}$
- $v_{esc} = \sqrt{\frac{2 * 1 \frac{px^3}{s^2} * 1000000}{410px}} \approx 69.843 \frac{px}{s}$
- Used bigger canvas, and plotted velocities during planet's travel

# Simulation #5: Escape Velocity

Overview

Chapter 1

Simulation #1

Simulation #2

Simulation #3

Chapter 2

Simulation #4

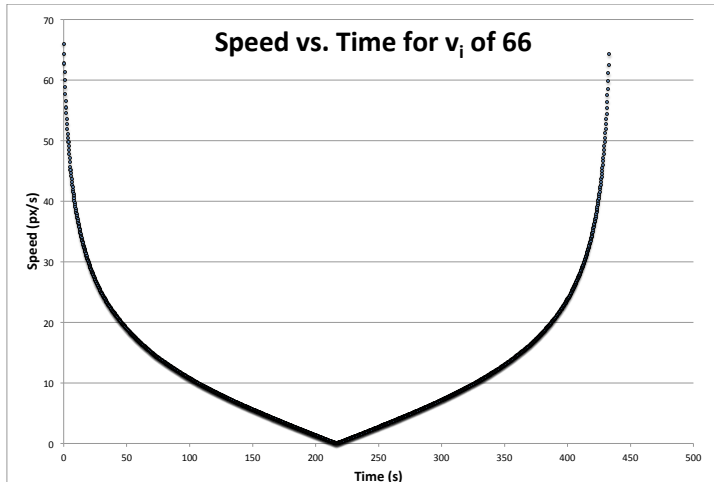
**Simulation #5**

Simulation #6

Chapter 3

Simulation #3

Conclusion



# Simulation #5: Escape Velocity

Overview

Chapter 1

Simulation #1

Simulation #2

Simulation #3

Chapter 2

Simulation #4

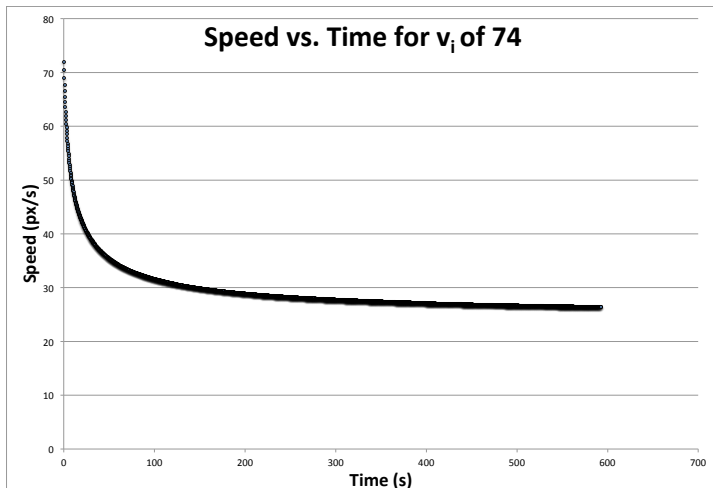
**Simulation #5**

Simulation #6

Chapter 3

Simulation #3

Conclusion



# Simulation #5: Escape Velocity

Overview

Chapter 1

Simulation #1

Simulation #2

Simulation #3

Chapter 2

Simulation #4

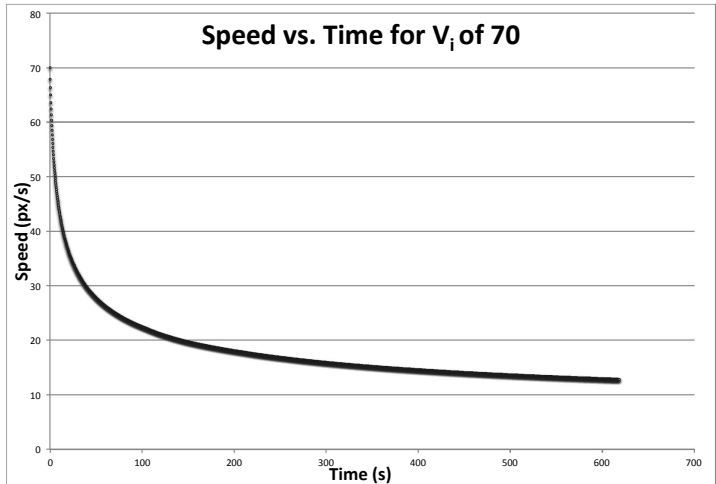
**Simulation #5**

Simulation #6

Chapter 3

Simulation #3

Conclusion



# Simulation #6: Kepler's 2nd law

Overview

Chapter 1

Simulation #1

Simulation #2

Simulation #3

Chapter 2

Simulation #4

Simulation #5

Simulation #6

Chapter 3

Simulation #3

Conclusion

- Early 1600's Johannes Kepler proposed laws explaining how planets orbit the sun
- Law #2: "The radius vector drawn from the Sun to a planet sweeps out equal areas in equal time intervals"
- Simulation shows constant  $\frac{dA}{dt}$

# Derivation of Kepler's 2nd Law

- Gravity force is *central* force

- $\vec{\tau} = \vec{r} \times \vec{F}_g = \frac{d\vec{L}}{dt}$

- $\vec{L} = \vec{r} \times \vec{p} = M_p \vec{r} \times \vec{v}$

- $L = M_p |\vec{r} \times \vec{v}|$

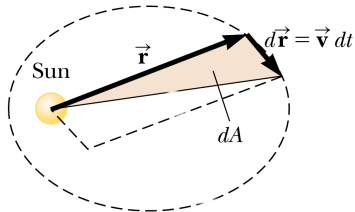


Figure : Relationship between  $\vec{r}$  and  $d\vec{r}$

Overview

Chapter 1

Simulation #1

Simulation #2

Simulation #3

Chapter 2

Simulation #4

Simulation #5

Simulation #6

Chapter 3

Simulation #3

Conclusion



# Derivation of Kepler's 2nd Law (Continued)

Overview

Chapter 1

Simulation #1

Simulation #2

Simulation #3

Chapter 2

Simulation #4

Simulation #5

Simulation #6

Chapter 3

Simulation #3

Conclusion

- $|\vec{r} \times d\vec{r}|$  area of parallelogram
- $dA = \frac{1}{2} |\vec{r} \times d\vec{r}| = \frac{1}{2} |\vec{r} \times \vec{v} dt| = \frac{1}{2} |\vec{r} \times \vec{v}| dt$
- From before,  $|\vec{r} \times \vec{v}| = \frac{L}{M_p}$
- $dA = \frac{1}{2} \left( \frac{L}{M_p} \right) dt$
- $\frac{dA}{dt} = \frac{1}{2} \left( \frac{L}{M_p} \right)$
- $L$  and  $M_p$  are constants

# Kepler's 2nd Law

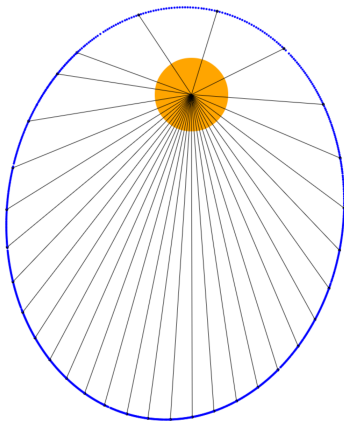


Figure : Screenshot of Kepler Law test simulation

Overview

Chapter 1

Simulation #1

Simulation #2

Simulation #3

Chapter 2

Simulation #4

Simulation #5

Simulation #6

Chapter 3

Simulation #3

Conclusion

# Chapter 3: Rotational Motion

Overview

Chapter 1

Simulation #1

Simulation #2

Simulation #3

Chapter 2

Simulation #4

Simulation #5

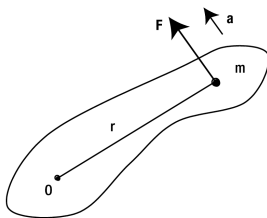
Simulation #6

Chapter 3

Simulation #3

Conclusion

- $\vec{\tau} = \vec{r} \times \vec{F}$
- $I = \int r^2 dm$



# Chapter 3: Rotational Motion

Overview

Chapter 1

Simulation #1

Simulation #2

Simulation #3

Chapter 2

Simulation #4

Simulation #5

Simulation #6

Chapter 3

Simulation #3

Conclusion

- Newton's 2nd law for rotation
- $\vec{T} = I\vec{\alpha}$
- Program updates  $\omega$  by calculating  $\alpha$  from  $T$  and  $I$ .

# Conclusion

Overview

Chapter 1

Simulation #1

Simulation #2

Simulation #3

Chapter 2

Simulation #4

Simulation #5

Simulation #6

Chapter 3

Simulation #3

Conclusion

- Simulations can be made very accurate with JavaScript
- Advantages of simulations involve sending instructions
- Future improvements could involve 3D

Overview

Chapter 1

Simulation #1  
Simulation #2  
Simulation #3

Chapter 2

Simulation #4  
Simulation #5  
Simulation #6

Chapter 3

Simulation #3

Conclusion

# Thank You