

INSTRUCTION SHEET

Momentum, Gravity, and Sound

"I can simulate the effects of gravity in an
XNA Game using

C# computer language."

"I can incorporate sound and music in an
XNA Game using

C# computer language."

LESSON -- SHUFFLEBOARD (Momentum and Gravity)

1) INTRODUCTION

- Demonstrate momentum and gravity in a video game while reinforcing the concepts learned.

2) NEW PROJECT – SHUFFLEBOARD

3) RESOURCES

- ShuffleBoardContent(Content) – Right Click – Add – Existing Item – Browse to Folder
- Puck1, Puck2, ShuffleCourt

4) VARIABLES

- Texture2D courtPic, puckPic;
Rectangle courtRec, puckRec0, puckRec1, puckRec2;
int puck = 0, speed = 0, counter = 0;
bool release = false;
GamePadState pad;

5) GAME1 CONSTRUCTOR

- graphics.PreferredBackBufferHeight = 940;
graphics.PreferredBackBufferWidth = 290;

6) INITIALIZE METHOD

- courtRec = new Rectangle(0, 0, 290, 940);
puckRec2 = puckRec1 = puckRec0 = new Rectangle(125, 900, 40, 40);

7) LOADCONTENT METHOD

- courtPic = Content.Load<Texture2D>("ShuffleCourt");
puckPic = Content.Load<Texture2D>("Puck");

8) DRAW METHOD

```
spriteBatch.Begin();  
  
spriteBatch.Draw(courtPic, courtRec, Color.White);  
  
spriteBatch.Draw(puckPic, puckRec0, Color.White);
```

```

spriteBatch.Draw(puckPic, puckRec1, Color.White);

spriteBatch.Draw(puckPic, puckRec2, Color.White);

spriteBatch.End();

```

9) UPDATE METHOD

- pad = GamePad.GetState(PlayerIndex.One);
- Determine which of 3 pucks to use and move it.
- switch (puck)
 - {
 - case 0:
 - puckRec0.Y -= speed;
 - break;
 - case 1:
 - puckRec1.Y -= speed;
 - break;
 - case 2:
 - puckRec2.Y -= speed;
 - break;
 - default:
 - // Reset the Game
 - count++
 - if (count >= 180)
 - {
 - puck = count = 0;
 - puckRec0.Y = puckRec1.Y = puckRec2.Y = 900;
 - }
 - break;
 - }
- // Speed of the Puck
 - if (release == false)
 - speed += (int)(2 * pad.Triggers.Right);
- // Release the Puck
 - if (speed > 0 && pad.Triggers.Right == 0)
 - release = true;
- // Bleed the Speed to 0
 - if (release == true)
 - {
 - speed = speed * 30 / 31;
 - if (speed == 0)
 - {
 - // Shot Complete
 - release = false;
 - puck++;
 - GamePad.SetVibration(PlayerIndex.One, 0, 0); // Foot Fault Reset
 - }
 - }

- switch (puck)


```

{
    case 0:
        puckRec0.Y -= speed;
        // Foot Fault – Vibration
        if (release == false && puckRec0.Y < 610)
            GamePad.SetVibration(PlayerIndex.One, 1, 1);
        break;
    case 1:
        puckRec1.Y -= speed;
        if (release == false && puckRec1.Y < 610)
            GamePad.SetVibration(PlayerIndex.One, 1, 1);
        break;
    case 2:
        puckRec2.Y -= speed;
        if (release == false && puckRec2.Y < 610)
            GamePad.SetVibration(PlayerIndex.One, 1, 1);
        break;
    case 3:
        // Reset the Game
        count++;
        if (count >= 180)
        {
            puck = count = 0;
            puckRec0.Y = puckRec1.Y = puckRec2.Y = 900;
        }
        break;
}

```

10) **RUN or F5**