

---

---

---

---

---

---

---

---

### XNA Recap

- XNA is a framework for writing games
- It is provided as a library of classes that your programs interact with to make games work
- Your games can run on Xbox 360, PC or Windows Phone
- XNA games are developed using Visual Studio 2010
  - Games are created as new project types

---

---

---

---

---

---

---

---

### XNA and Pong

- Last time we got a ball to move down the screen
- Now we need to make the ball bounce around the screen
- Now we need to discover how we can create paddles and control them using a gamepad or keyboard
- Then we can start building a game

---

---

---

---

---

---

---

---

## Controlling Ball Movement

```
int ballXSpeed = 3;
int ballYSpeed = 3;
```

- To manage the speed of the ball we can use a pair of member variables in our game class
  - One for the X speed and one for the Y speed
- Each time Update is called these are used to update the values of the X and Y position of the draw rectangle
- In a proper game we would calculate these values to make sure the game plays at the same speed on all displays

4

Creating Gameplay with XNA

---

---

---

---

---

---

---

---

---

---

## Moving the Ball

```
protected override void Update(GameTime gameTime)
{
    ballRectangle.X = ballRectangle.X + ballXSpeed;
    ballRectangle.Y = ballRectangle.Y + ballYSpeed;

    base.Update(gameTime);
}
```

- The Update method is where the speed values are used to update the rectangle position for the ball
- The next call of Draw will draw the ball in the new position

5

Creating Gameplay with XNA

---

---

---

---

---

---

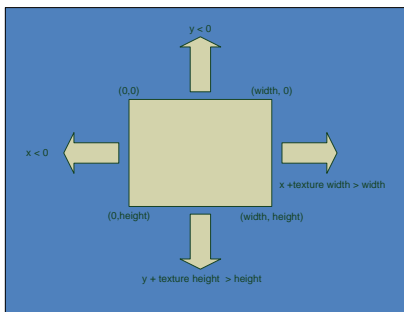
---

---

---

---

## Going off the Edge



6

Creating Gameplay with XNA

---

---

---

---

---

---

---

---

---

---

## Making the Ball Bounce

```
ballRectangle.X += ballXSpeed;
if (ballRectangle.X < 0 ||
    ballRectangle.X + ballRectangle.Width >
    GraphicsDevice.Viewport.Width)
{
    ballXSpeed = -ballXSpeed;
}
```

- When the ball reaches the edge of the screen it must change direction
- We can do this by reversing the sign of the speed value to reverse the effect of the update

7

Creating Gameplay with XNA

---

---

---

---

---

---

---

---

---

---

## Making a Paddle

- The paddle is made from a texture, just like the ball
- This time I've made a slightly more interesting one which uses transparency
- The paddle is loaded as a texture resource, just as the ball is



8

Creating Gameplay with XNA

---

---

---

---

---

---

---

---

---

---

## Loading GameTextures

```
protected override void LoadContent()
{
    ballTexture = Content.Load<Texture2D>("ball");
    lPaddleTexture = Content.Load<Texture2D>("lpaddle");
    rPaddleTexture = Content.Load<Texture2D>("rpaddle");
    ...
}
```

- When the game starts the LoadContent method is called to load textures and other game assets
- We now have three textures in the game

9

Creating Gameplay with XNA

---

---

---

---

---

---

---

---

---

---

## Scaling GameTextures

```
ballRectangle = new Rectangle(
    50, 50,
    Window.ClientBounds.Width / 20,
    Window.ClientBounds.Width / 20);
```

- Each game element will be drawn in a rectangle on the screen
- We need to scale the rectangle so that the element is a sensible size
  - This must allow for different sized screens

10

Creating Gameplay with XNA

---

---

---

---

---

---

---

---

---

---

## Drawing GameTextures

```
protected override void Draw(GameTime gameTime)
{
    graphics.GraphicsDevice.Clear(Color.CornflowerBlue);
    spriteBatch.Begin();

    spriteBatch.Draw(ballTexture, ballRectangle, Color.White);
    spriteBatch.Draw(lPaddleTexture, lPaddleRectangle,
        Color.White);
    spriteBatch.Draw(rPaddleTexture, rPaddleRectangle,
        Color.White);
    spriteBatch.End();
    base.Draw(gameTime);
}
```

- The Draw method draws all the objects in the game

11

Creating Gameplay with XNA

---

---

---

---

---

---

---

---

---

---

## Representing GamePad state in XNA

- The state of a gamepad is represented by an instance of the GamePadState class
- You can ask XNA to create an instance for any gamepad
- You can then read information from this instance to tell you about that gamepad



12

Creating Gameplay with XNA

---

---

---

---

---

---

---

---

---

---

## Reading the Gamepad

```
GamePadState pad1 = GamePad.GetState(PlayerIndex.One);
if (pad1.IsConnected)
{
    if (pad1.DPad.Up == ButtonState.Pressed)
    {
        lPaddleRectangle.Y -= lPaddleSpeed;
    }
}
else
{
    lPaddleRectangle.Y = ballRectangle.Y;
}
```

- This code links the gamepad for player 1 to the left hand paddle
- If the pad is not connected the paddle tracks the ball

13

Creating Gameplay with XNA

---

---

---

---

---

---

---

---

---

---

---

---

## Reading the Keyboard

- The keyboard is read in just the same way
- However, there is only one keyboard on the system
- You can plug a USB keyboard into an Xbox 360 if you wish
- An XNA game can check if keys are being held down
- This includes shift and control keys

14

Creating Gameplay with XNA

---

---

---

---

---

---

---

---

---

---

---

---

## Reading the Keyboard

```
KeyboardState keyboard = Keyboard.GetState();
if (keyboard.IsKeyDown(Keys.A))
{
    lPaddleRectangle.Y -= lPaddleSpeed;
}
```

- This code links the keyboard to the left hand paddle
  - In this version you press the A key to move the paddle up the screen
- Note that there is no way of telling whether or not the keyboard is present

15

Creating Gameplay with XNA

---

---

---

---

---

---

---

---

---

---

---

---

## Detecting Collisions

- We need to make the ball bounce off the paddles when the two collide
- In the console version of the game we tested to see if ball and paddle occupied the same part of the screen
- In the case of XNA we need to see if the rectangles which control the position of the ball and paddle intersect

16

Creating Gameplay with XNA

---

---

---

---

---

---

---

---

---

---

## Rectangle Intersection

```
if (ballRectangle.Intersects(paddleRectangle))
{
    ballXSpeed = -ballXSpeed;
}
```

- The Rectangle structure provides a method called Intersects which can be used to detect if two rectangles intersect
- If the paddle and ball rectangles intersect we must reverse the X direction of movement of the ball to have it bounce off the paddle

17

Creating Gameplay with XNA

---

---

---

---

---

---

---

---

---

---

## Completing the Game

- A finished game must also detect when the ball reaches the edges of the screen
- This is when a point has been scored
- I will leave you to create this code
- However, you will also need to draw text on the screen to display messages to the players
- This turns out to be very easy

18

Creating Gameplay with XNA

---

---

---

---

---

---

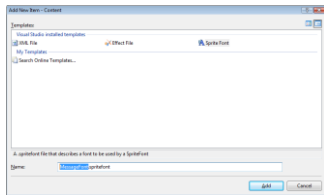
---

---

---

---

## Adding a SpriteFont



- A SpriteFont is a content item that lets you draw text on the screen
- It provides a set of character designs of a particular size

19

Creating Gameplay with XNA

---

---

---

---

---

---

---

---

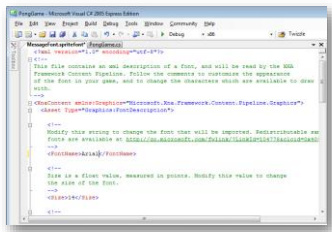
---

---

---

---

## SpriteFont XML



- The font used and the size are set in an XML file
- You can edit this to get different sizes and styles

20

Creating Gameplay with XNA

---

---

---

---

---

---

---

---

---

---

---

---

## Loading a Font

```
SpriteFont font;

protected override void LoadContent()
{
    // Load the bat and ball textures
    font = Content.Load<SpriteFont>("MessageFont");
}
```

- The Content Manager will fetch the font
- The font can be stored in a variable which a member of the game class
- You can use multiple fonts if you want different text styles

21

Creating Gameplay with XNA

---

---

---

---

---

---

---

---

---

---

---

---

## Using a Font

```
protected override void Draw( GameTime gameTime)
{
    graphics.GraphicsDevice.Clear(Color.CornflowerBlue);
    spriteBatch.Begin();
    spriteBatch.DrawString(
        font,
        "Hello world",
        new Vector2(100, 100),
        Color.White);
    // Draw the other textures here
    spriteBatch.End();
    base.Draw(gameTime);
}
```

- The DrawString method renders a string using the font that has been loaded

22

Creating Gameplay with XNA

---

---

---

---

---

---

---

---

---

---

## My Pong Game



23

Creating Gameplay with XNA

---

---

---

---

---

---

---

---

---

---

## Summary

- XNA is a Framework of classes that are used to write games
- You load textures into your game and use them to draw the display
- Texture drawing is controlled by rectangles, which give the position and size of the drawn item
- User input is obtained from objects that hold a snapshot of the state of an input device
- You can add font items to the game content that allow text to be drawn

24

Creating Gameplay with XNA

---

---

---

---

---

---

---

---

---

---