Creating Gameplay with XNA

Rob Miles

Department of Computer Science

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

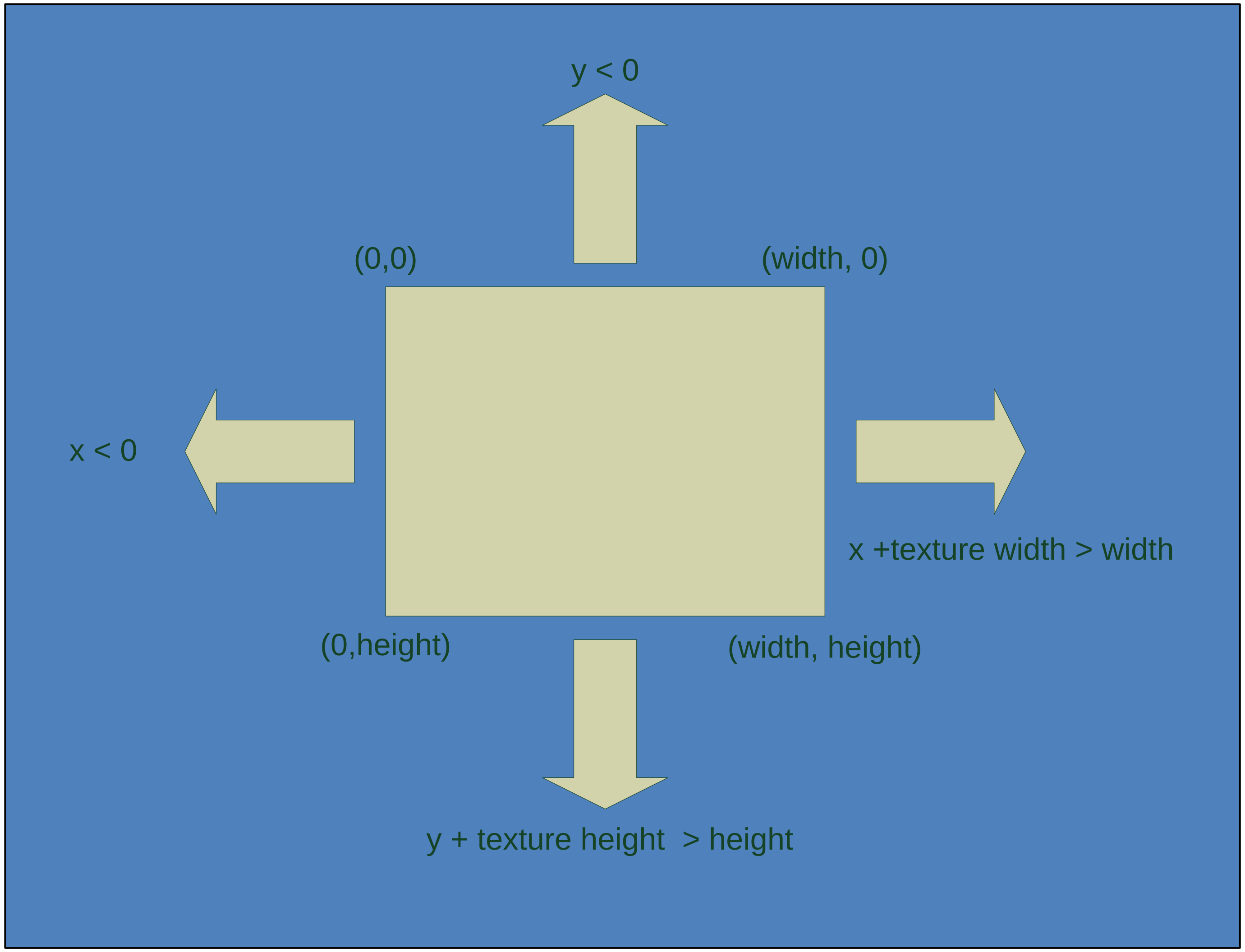
Moving the Ball

```
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

Going off the Edge



Making the Ball Bounce

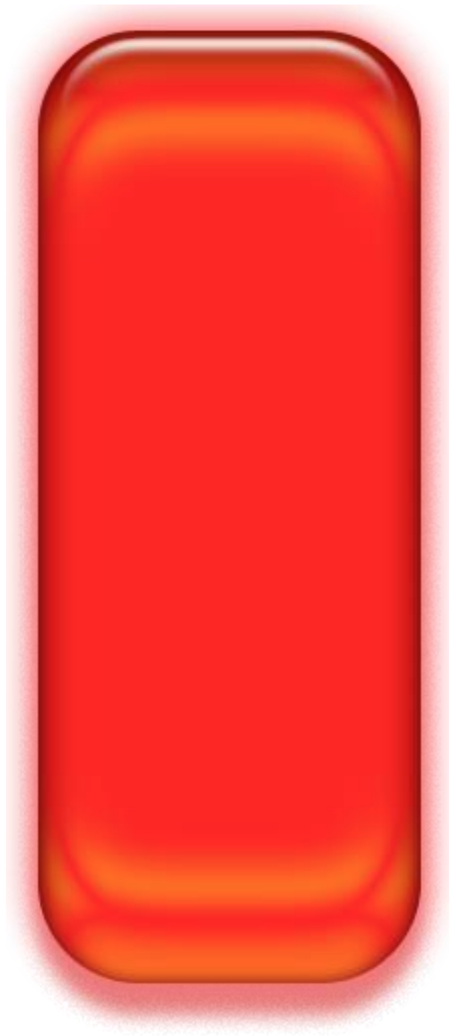
```
ballRectangle.X += ballXSpeed;

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

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



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

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

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

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

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

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

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

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

Rectangle Intersection

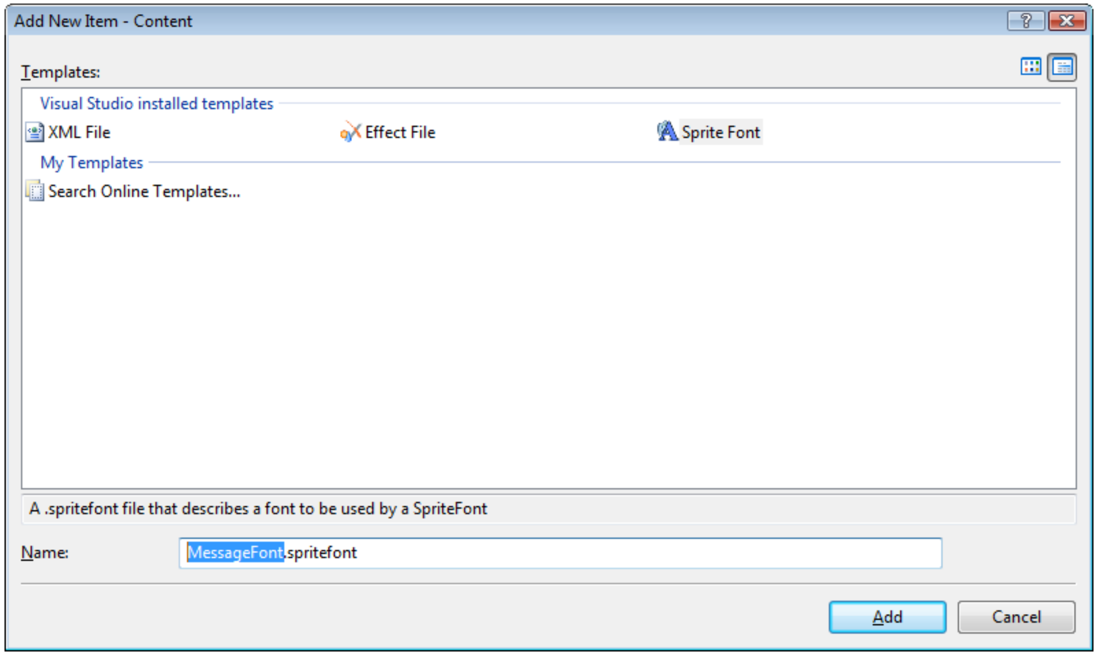
```
if (ballRectangle.Intersects(lPaddleRectangle))
{
    ballXSpeed = -ballXSpeed;
}
```

- 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

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

Adding a SpriteFont


Add New Item - Content

Templates:

Visual Studio installed templates

XML File | Effect File | Sprite Font

My Templates

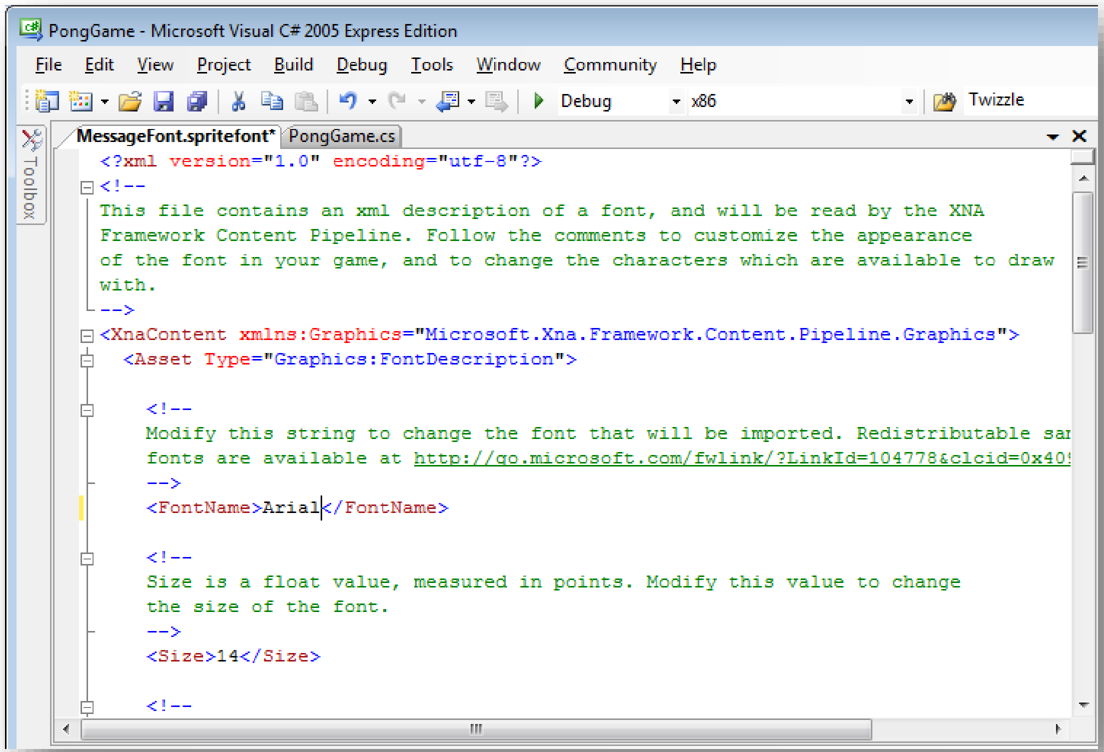
Search Online Templates...

A .spritefont file that describes a font to be used by a SpriteFont

Name: MessageFont.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

SpriteFont XML


PongGame - Microsoft Visual C# 2005 Express Edition
File Edit View Project Build Debug Tools Window Community Help
Debug x86 Twizzle
MessageFont.spritefont* PongGame.cs
Toolbox

```
<?xml version="1.0" encoding="utf-8"?>
<!--
This file contains an xml description of a font, and will be read by the XNA
Framework Content Pipeline. Follow the comments to customize the appearance
of the font in your game, and to change the characters which are available to draw
with.
-->
<XnaContent xmlns:Graphics="Microsoft.Xna.Framework.Content.Pipeline.Graphics">
  <Asset Type="Graphics:FontDescription">

    <!--
    Modify this string to change the font that will be imported. Redistributable sa
    fonts are available at http://go.microsoft.com/fwlink/?LinkId=104778&clcid=0x40
    -->
    <FontName>Arial</FontName>

    <!--
    Size is a float value, measured in points. Modify this value to change
    the size of the font.
    -->
    <Size>14</Size>

    <!--
```

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

Loading a Font

```
SpriteFont font;

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

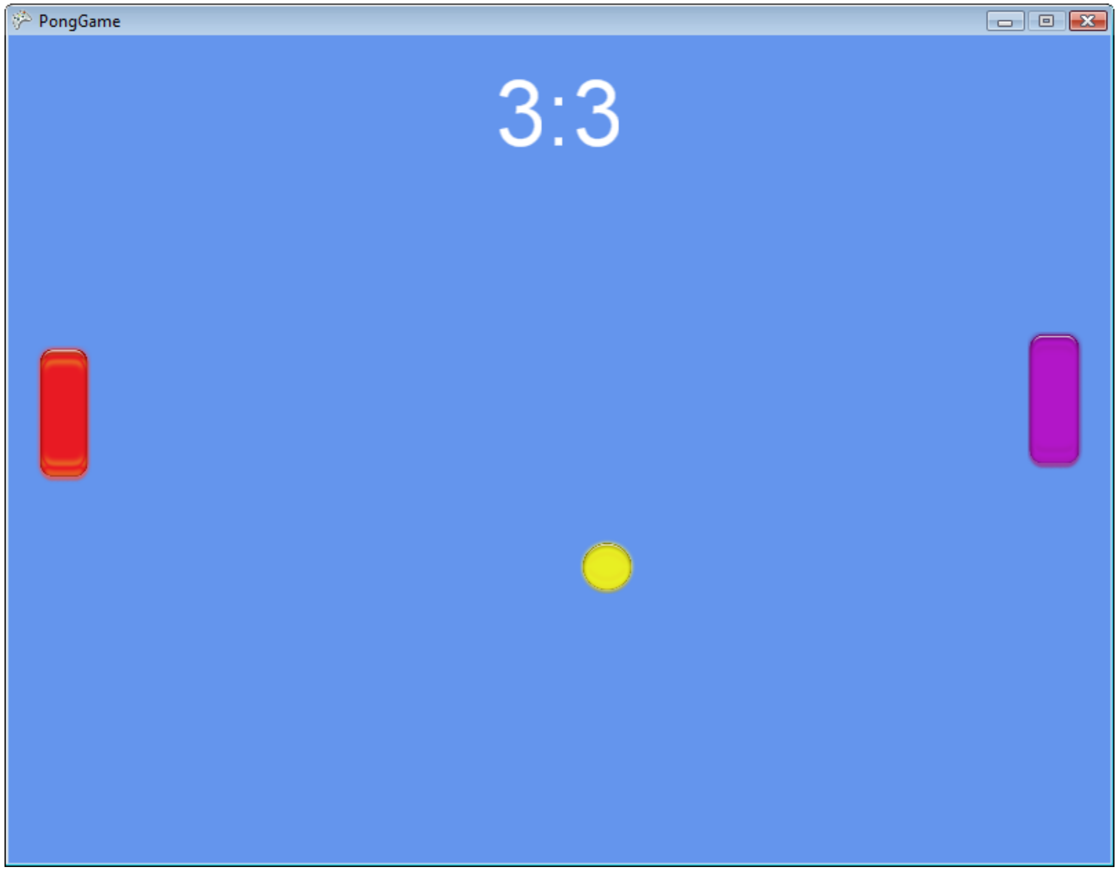
- The Content Manager will fetch the font
- The font can be stored in a variable which a member of the game class
- You can use multiple fonts if you want different text styles

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

My Pong Game



UNIVERSITY OF Hull

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