

```
using System;
using System.Collections.Generic;
using Microsoft.Xna.Framework;
using Microsoft.Xna.Framework.Audio;
using Microsoft.Xna.Framework.Content;
using Microsoft.Xna.Framework.GamerServices;
using Microsoft.Xna.Framework.Graphics;
using Microsoft.Xna.Framework.Input;
using Microsoft.Xna.Framework.Net;
using Microsoft.Xna.Framework.Storage;
namespace WindowsGame1
```

```
{
    /// <summary>
    /// This is the main type for your game
    /// </summary>
    public class Game1 : Microsoft.Xna.Framework.Game
    {
        GraphicsDeviceManager graphics;
        SpriteBatch spriteBatch;
        public Game1()
        {
            graphics = new GraphicsDeviceManager(this);
            Content.RootDirectory = "Content";
        }
    }
}
```

```
/// <summary>
/// Allows the game to perform any
/// initialization it needs to before
/// starting to run.
/// This is where it can query
/// for any required services and
/// load any non-graphics
/// related content. Calling
/// base.Initialize will enumerate
/// through any components
/// and initialize them as well.
/// </summary>
protected override void Initialize()
```

```
{
    // TODO: Add your initialization
    // logic here
    base.Initialize();
}
/// <summary>
/// LoadContent will be called once
/// per game and is the place to load
/// all of your content.
/// </summary>
protected override void LoadContent()
{
    // Create a new SpriteBatch,
    // which can be used to draw textures.
    spriteBatch = new SpriteBatch(GraphicsDevice);
    // TODO: use this.Content to load your game content
}
```

Free for all

A wealth of open-source and freeware tools and middleware are now available to games companies. 3D World explores the new open-source pipelines, and meets the studios who are trying to democratise game development

BY MARK RAMSHAW

Driven by a technology arms race, the games development industry now regularly works with budgets in the millions, spending huge sums on tools, staff and time in a bid to create software that will capture the public's imagination. While this has provided consumers with a choice of titles boasting incredibly high production values, it's also helped foster a very different, indie-level style of game development.

By reducing development costs and taking advantage of the various cheaper publishing opportunities now available (ranging from budget and self-publishing to digital distribution via services like Steam and Xbox Live Arcade), smaller development teams have fewer professional and creative restrictions and are able to work efficiently on projects closer to their own hearts - and to those of a substantial number of gamers looking for an alternative to those big-buck epics. And when it comes to reducing development costs, there are few solutions more effective than making use of tools and code that don't cost a cent.

When it comes to the development workflow, there is open-source software (OSS) and freeware available to handle just about every task, from version control right through to scripting and compiling (see 'The open-source pipeline', overleaf). Artists are particularly well catered for.

● A world of possibilities: game development with free and open-source tools ranges from titles like *Project Apricot* (main image), created entirely with open-source software, to free development frameworks released by major commercial companies, such as Microsoft's *XNA Game Studio* (overlaid code).