Getting Started

BigWorld Technology 2.0. Released 2010.

Software designed and built in Australia by BigWorld.

Level 2, Wentworth Park Grandstand, Wattle St
Glebe NSW 2037, Australia
www.bigworldtech.com

Copyright © 1999-2010 BigWorld Pty Ltd. All rights reserved.

Table of Contents

Chapter 1. Getting Started

BigWorld Technology is a mature middleware platform for developers of Massively Multiplayer Online Games and Virtual Worlds. Development of an online game usually requires an art pipeline used to create the look and feel of the game levels, and a development framework to create the game logic and AI.
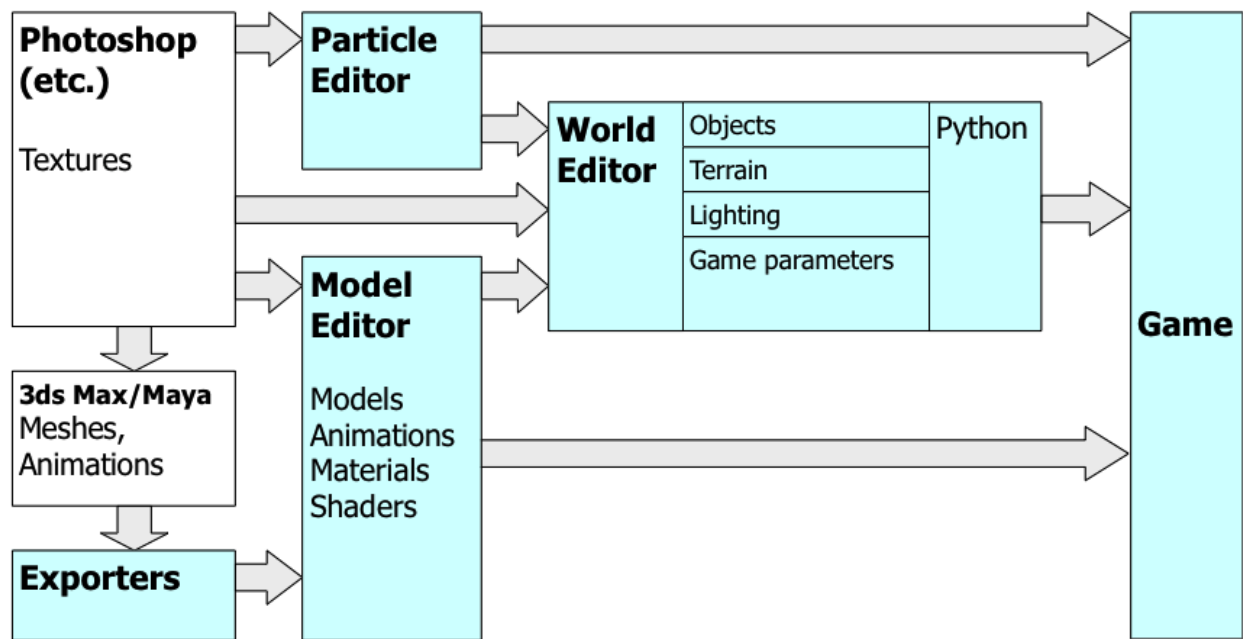
1.1. BigWorld Technology Art Pipeline

The BigWorld Technology art pipeline is used by artist to create game assets and includes the following steps (see image below):

- Textures are developed using a texture creation tool (Photoshop, Gimp or equivalent).

- Meshes, animations and morph targets are built using 3ds Max or Maya and exported using the supplied exporters.

- Particle systems are built using Particle Editor.

- Model Editor is used to edit specific Models' parameters (including Lod levels, animation blends, material overrides, and shader parameters).

- World Editor is used to build the final game levels (integrating all art assets which were generated in the previous steps).

  World Editor allows creating the scenery of the level as follows:

  - Sculpting outdoor terrain scenery.

  - Creating indoor shell systems such as dungeons.

  - Placement of art assets generated and exported using the 3D modeling tools mentioned above.

  - Placement of particles that have been generated using Particle Editor.

  - Controlling default weather and lighting of the world.

  - Other level specific parameters may be also be modified as required using World Editor.

BigWorld Technology Art Pipeline

1.2. BigWorld Technology Development Framework

The BigWorld Technology Development Framework is used by developers to develop game logic and AI and is based on the Python scripting language. This Framework includes the following components:

- Standard Python is used as the game engine scripting language.

- The BigWorld Technology engine exposes a rich set of documented Python APIs to be used for game scripting.

- Distributed Python objects are automatically instantiated on the game client and the game server to allow data synchronisation between the client and server.

- Advanced users can extend either the game client or the game server using C++.

Chapter 2. How to Setup Your Development Environment

- Review the BigWorld documentation for further details about BigWorld Technology.

  The Documentation Roadmap[1] lists all the available BigWorld documentation and will guide your reading.

- Install the game client machines.

  These machines will contain the game client and content creation tools which can be used by either developers or artists. For information regarding client machine installation procedures see the Client Installation Guide.

- Install one or more server machines.

  Developing and testing online game logic will require setting up a server. For information regarding server installation see the Server Installation Guide. For information regarding server tools installation see the Server Tools Installation Guide.

- First time users should review the following documents:

  - The chapter *Running BigWorld Technology Demos* on page 9 .

  - The Tutorial document.

- Setup your game directories.

  - For Indie customers, The BigWorld Technology installer will create a skeleton directory to be used as the first game project called `my_game`.

  - Customers using the Commercial edition or the Indie edition should create their second project using the script `tutorial/create_new_project.py`. For example:

    ```
    c:\Python25\python.exe c:\bigworld\tutorial\create_new_project.py
    myproject
    ```

  The created project directory will contain all your game code and assets.

- Customers using the Indie Edition should not change the original files installed by the BigWorld Technology Indie Edition installer as these changes may be lost when installing newer versions.

  It is recommended to rename the `my_game` directory to your game's name. The `run.bat` batch file included in this directory will run your game.

- When working on game aspects that require you to test online functionality (i.e., connected to a game server), you will need to synchronise the changes between the client and server before starting either. Synchronising can be performed in one of the following ways:

  - Assets can be stored on a shared network directory, therefore being avaialble for both the client and the server.

  - Assets can be committed to a source control system and synchronised on both the server and the client.

  Both methods outlined above have the potential for problems when used in an environment with a number of people working with the same data. As a result of these potential problems BigWorld has a set of recommendations to assist in preventing and avoiding problems in a long term development environment outlined in the Client Programming Guide section *Shared Development Environments*.

Chapter 3. Running BigWorld Technology Demos

Two demos are included in your package:

- Tutorial - This demo is a basic skeleton project containing a minimal game implementation. The Tutorial document will walk you through creating a similar project. New game projects created based on the instructions above (see chapter *How to Setup Your Development Environment* on page 7) will contain a copy of the Tutorial.

- Fantasy Demo - This demo is the BigWorld Technology official demo which shows most of the engine capabilities. It is not recommended to use this demo as a skeleton game as some of the code in it is not production quality and as it is usually too complicated to start with. Customers can reuse Fantasy Demo code in their game but should review it to make sure it fits their game requirements and that it will scale correctly for their game.

Running one of these demos require the following steps:

- Setup a client and a server machine as required above (see chapter *How to Setup Your Development Environment* on page 7).

- Synchronise the demo assets (tutorial/res or fantasydemo/res) to the server machine (see the Client Programming Guide section *Shared Development Environments*.) and update your .bwmachined.conf file in accordance.

- Start the server using the Cluster Control module of the Web Console (see Server Operations Guide for more details).

- For FantasyDemo: Start the client and choose the option "Search for servers on local network" to search for your server and connect to it.

- For Tutorial: follow the instructions on the bigworld/doc/bigworld_tutorial.pdf "Connecting the client to the server" chapter.

Chapter 4. Getting Support

We try to make our documentation as extensive as possible and recommend reviewing the documentation before raising support issues.

customers using the BigWorld Technology Commercial Edition can get support using our support system: http://support.bigworldtech.com

Indie customers can use the moderated support forum at https://forum.bigworldtech.com/.