

# Chapter 11

## Introduction to Games Programming



### 11.1 Introduction

Games programming is performed by developers/coders who implement the logic that drives a game.

Historically games developers did everything; they wrote the code, designed the sprites and icons, handled the game play, dealt with sounds and music, generated any animations required etc. However, as the game industry has matured games companies have developed specific roles including Computer Graphics (CG) animators, artists, games developers and games engine and physics engine developers etc.

Those involved with code development may develop a physics engine, a games engine, the games themselves, etc. Such developers focus on different aspects of a game. For examples a game engine developer focusses on creating the framework within which the game will run. In turn a physics engine developer will focus on implementing the mathematics behind the physics of the simulated games world (such as the effect of gravity on characters and components within that world). In many cases there will also be developers working on the AI engine for a game. These developers will focus on providing facilities that allow the game or characters in the game to operate intelligently.

Those developing the actual game play will use these engines and frameworks to create the overall end result. It is they who give life to the game and make it an enjoyable (and playable) experience.

### 11.2 Games Frameworks and Libraries

There are many frameworks and libraries available that allow you to create anything from simple games to large complex role playing games with infinite worlds.

One example is the Unity framework that can be used with the C# programming language. Another such framework is the Unreal engine used with the C++ programming language.

Python has also been used for games development with several well known games titles depending on it in one way or another. For example, Battlefield 2 by Digital Illusions CE is a military simulator first-person shooter game. Battlefield Heroes handles portions of the game logic involving game modes and scoring using Python.

Other games that use Python include Civilisation IV (for many of the tasks), Pirates of the Caribbean Online and Overwatch (which makes its choices with Python).

Python is also embedded as a scripting engine within tools such as Autodesk's Maya which is a computer animation toolkit that is often used with games.

### 11.3 Python Games Development

For those wanting to learn more about game development; Python has much to offer. There are many examples available online as well as several game oriented frameworks.

The frameworks/libraries available for games development in Python including:

- **Arcade**. This is a Python library for creating 2D style video games.
- **pyglet** is a windowing and multimedia library for Python that can also be used for games development.
- **Cocos2d** is a framework for building 2D games that is built on top of pyglet.
- **pygame** is probably the most widely used library for creating games within the Python world. There are also many extensions available for pygame that help to create a wide range of different types of games.

We will focus on pygame in the next two chapters in this book.

Other libraries of interest to Python games developers include:

- **PyODE**. This is an open-source Python binding for the Open Dynamics Engine which is an open-source physics engine.
- **pymunk** Pymunk is a easy-to-use 2D physics library that can be used whenever you need 2d rigid body physics with Python. It is very good when you need 2D physics in your game, demo or other application. It is built on top of the 2D physics library Chipmunk.
- **pyBox2D** pybox2d is a 2D physics library for your games and simple simulations. It's based on the Box2D library written in C++. It supports several shape

types (circle, polygon, thin line segments) as well as a number of joint types (revolute, prismatic, wheel, etc.).

- **Blender.** This is an open-source 3D computer graphics software toolset used for creating animated films, visual effects, art, 3D printed models, interactive 3D applications and video games. Blender's features include 3D modeling, texturing, raster graphics editing, rigging and skinning, etc. Python can be used as a scripting tool for creation, prototyping, game logic and more.
- **Quake Army Knife** which is an environment for developing 3D maps for games based on the Quake engine. It is written in Delphi and Python.

## 11.4 Using Pygame

In the next two chapters we will explore the core *pygame* library and how it can be used to develop interactive computer games. The next chapter explores *pygame* itself and the facilities it provides. The following chapter develops a simple interactive game in which the user moves a starship around avoiding meteors which scroll vertically down the screen.

## 11.5 Online Resources

For further information games programming and the libraries mentioned in this chapter see:

- <https://unity.com/> the C# framework for games development.
- <https://www.unrealengine.com> for C++ games development.
- <http://arcade.academy/> provides details on the Arcade games framework.
- <http://www.pyglet.org/> for information on the piglet library.
- <http://cocos2d.org/> is the home page for the Cocos2d framework.
- <https://www.pygame.org> for information on *pygame*.
- <http://pyode.sourceforge.net/> for details of the PyODE bindings to the Open Dynamics Engine.
- <http://www.pymunk.org/> provides information on *pymunk*.
- <https://github.com/pybox2d/pybox2d> which is a Git hub repository for *pyBox2d*.
- <https://git.blender.org/gitweb/gitweb.cgi/blender.git> Git Hub repository for Blender.
- <https://sourceforge.net/p/quark/code> SourceForge repository for Quake Army Knife.
- <https://www.autodesk.co.uk/products/maya/overview> for information on Autodesk's Maya computer animation software.