

УДК 811.111:629.735-519

Krupkin T., Vanik I.

Game Development with Unity

Belarusian National Technical University
Minsk, Belarus

In today's world, the creation of video games is one of the major sectors of the entertainment industry. The scale of the game industry is equivalent, for example, to the film industry. And compared to the rate of growth over the past five years, the video game industry is far ahead.

Games are made using game engines, a set of tools that allows developers to work with graphics, physics, scripts and other things. Engines can be written from scratch, but it will take an unreasonably long time. While it is developed for a specific project, there is no confidence that by the end of the engine development the genre or even the entire industry will not be outdated. Therefore, it is rational to use an off-the-shelf engine.

Unity is a cross-platform game engine developed by Unity Technologies. It became available to the public in June 2005 at the World Developers Conference of Apple Inc. Since then, the engine has been progressively upgraded to support various mobile, desktop, console, and virtual reality platforms [1]. Typically, a game engine provides many features that allow you to use them in a variety of games, including physical environment modeling, normal maps, dynamic shadows, and more. Unlike many game engines, Unity has three main advantages: an extremely productive visual workflow, probably the best asset market on the game market - Unity Asset Store and a high level of cross-platform support [2, p. 4-5]. Among the disadvantages there is the limitation of the

visual editor when working with multi-component schemes. So, in complex scenes visual work is difficult. The second disadvantage is the lack of support for Unity links to external libraries, which programmers have to configure themselves. It's quite a challenge for them. [2, p. 6-7].

Before start working with Unity, future developers should know that all scripts and game mechanics are written in C# programming language. Next, the game engine must be installed, Unity Technologies offers several options - free for individual development and students, paid versions for teamwork, as well as industrial builds for enterprises [3].

As for the interface, there are five main windows in the Unity user interface. They are the Scene, Game, Hierarchy, Project, and Inspector windows. Each window can be clicked and dragged into position [4].

A unique feature of Unity is that all 3D models, HUDs, sprites, particle systems, lights, literally everything in the scene are a GameObjects. Most of GameObjects will require some sort of graphical shell - a texture, a sprite, or a 3D model [5].

Textures or models cannot be created in the game engine without special plug-ins, but they can be purchased or downloaded for free from the Unity Asset Store, after which they can be imported into the project. There's at least one third-party modeling plug-in - ProBuilder, that allows users to model 3D components right inside Unity. Built-in Sprite Creator, Sprite Editor, and Sprite Renderer can help developers work with sprites or, in other words, with 2D textures [5].

As for the graphics, Unity supports Vulkan, DirectX, Metal, and OpenGL graphics APIs, depending on platform-specific API availability [4].

Everything that runs in a game exists in a Scene. Developers can have as many scenes as they want in their project. However, just adding their own GameObjects to the scenes isn't enough. To create a complete picture, game

developers need to take care of rendering, lighting, shaders, visual effects, etc. Unity has the functionality to handle all of these things. The engine uses Render Pipeline technology, which performs a series of operations that take the contents of a Scene, and displays them on a screen. These operations are: rendering, culling and post-processing. Unity offers built-in Render Pipeline or scripted Universal Render Pipeline and High Definition Render Pipeline. Unity users can choose a suitable one or even create their own [4].

Lighting in Unity works simply by mapping how light behaves in the real world. Shadows add a degree of depth and realism to a Scene. Lights can cast shadows from a GameObject onto other parts of itself, or onto nearby GameObjects [4].

Unity also provides a number of post-processing and full-screen effects that can significantly improve the appearance of any application with a minimum of setup time. For example, the Particle System models and displays many small images or grids, called particles, to create a visual effect. Each particle in the system represents a separate graphic effect element. The system models each particle together to create the impression of a complete effect [4].

GameObjects, textures, graphics are very important, but any game requires not only the place where the action takes place, but also the means of displaying abstract interactions and information. Unity provides three UI systems that developers can use to create user interfaces for the Unity Editor and applications made in the Unity Editor: UI Toolkit, the Unity UI package and Immediate Mode Graphical User Interface [4].

To set any functionality for a GameObject, Unity users have to add a Component. Unity allows developers to create their own Components using scripts. Scripts are written in the C# programming language and allow game developers to set some behavior for their objects, create all kinds of

dependencies, add functionality to their buttons and text, and much more.

There is a way not to write hundreds of lines of code. Visual scripting in Unity allows creators to develop gameplay mechanics or interaction logic with a visual, graph-based system, instead of writing traditional lines of code. Examples of Visual Scripting are Bolt, PlayMaker, and others [6].

Thus, using an off-the-shelf game engine can greatly simplify the development of new games, make their production cheaper, and significantly reduce the time to launch. Unity game engine is one of the most popular for a reason. The engine has continued to be updated since 2005, and during that time it has grown so much that game development has become many times easier. Unity has automated many of the development processes, making it much more enjoyable.

References:

1. Unity [Electronic resource]. – Mode of access: <https://www.bisacorporation.com> – Date of access: 22.04.2022.
2. Hocking, J. Unity in action / J. Hocking. – Manning Publications Co., 2015. – 326 p.
3. Plans and pricing [Electronic resource]. – Mode of access: <https://store.unity.com/> – Date of access: 22.04.2022.
4. Unity User Manual 2022.2 (alpha) [Electronic resource]. – Mode of access: <https://docs.unity3d.com/2022.2/Documentation/Manual/> – Date of access: 23.04.2022.
5. Unity: Developing Your First Game with Unity and C# [Electronic resource]. – Mode of access: <https://docs.microsoft.com/en-us/archive/msdn-magazine/2014/august/unity-developing-your-first-game-with-unity-and-csharp> – Date of access: 23.04.2022.
6. Visual scripting [Electronic resource]. – Mode of access: <https://unity.com/products/unity-visual-scripting> – Date of access: 23.04.2022.