

**MINISTRY FOR INFORMATION TECHNOLOGIES AND
COMMUNICATIONS OF THE REPUBLIC OF UZBEKISTAN
TASHKENT UNIVERSITY OF INFORMATION TECHNOLOGIES**

To protect HoD

_____ **Djumanov J.H.**

«___» _____ **2016 г.**

BACHELORS GRADUATION WORK

«Development games in Unity3d»»

Graduate

(sign)

Mirjalolov B.T.

Instructor

(sign)

Сайфуллаева Н.А.

Reviewer

(sign)

Yakhubov A.B.

(ф.и.о.)

LSE consultant

(sign)

Abdullayeva S.M.

(ф.и.о.)

Ташкент 2016

**MINISTRY FOR INFORMATION TECHNOLOGIES AND
COMMUNICATIONS OF THE REPUBLIC OF UZBEKISTAN
TASHKENT UNIVERSITY OF INFORMATION TECHNOLOGIES**

Faculty: *Computer engineering*

Department: *Computer systems and networks.*

Direction (specialty): **5610600 – «Technique and servicing technology»
(services industry)**

«A P P R O V E D»

HoD «CS»

_____ **J.H. Djumanov**

«_____» _____ **2016 г.**

**To perform graduate work
ASSIGNMENT**

Mirjalolov Bobur Tokhir o'g'li

Subject of graduate work: «Development games in Unity3d»

Approved by order of the university on: «17» november 2015 №1195

The term of delivery of the finished work: «30» may 2016

Contents of settlement and explanatory information:

Contents of settlement and explanatory notes

Introduction

- 1. Analytical review**
- 2. Development environment**
- 3. Projection and development**
- 4. Examples of the finished application**
- 5. Conclusion**
- 6. Presentation**

List of graphic materials: Presentation in Microsoft Office 2016

Assignment received: «20» yanuary2016.

Instructor:
Assignment received:

Sayfullayeva N.A.
Mirjalolov B.T.

Consultants on separate sections of graduate work

	Руководитель:	Sign	
		Assignment given	Assignment received
<i>Introduction</i>	Sayfullayeva N.A.		
<i>Chapter 1</i>	Sayfullayeva N.A.		
<i>Chapter 2</i>	Sayfullayeva N.A.		
<i>Chapter 3</i>	Sayfullayeva N.A.		
<i>Chapter 4</i>	Abdullayeva S.M.		
<i>Conclusion</i>	Sayfullayeva N.A.		

9. Progress chart

№	Name of section work	Term	Implementation
1	Analytical review	15.02	16.02
2	Development environment	29.03	30.03
3	Projection	20.04	21.04
4	Development	25.04	26.04
5	Life Safety and environment	15.05	16.05
7	Conclusion	30.05	30.05

Graduate _____
(sign)

« _____ » _____ **2016г.**

Instructor _____

« _____ » _____ **2016г.**

Abstract

In the graduation-qualifying work developed 2D game on a cross-platform development environment Unity3d. various plug-ins, vector shapes have been used, which, after completion exported to raster graphics, and used in the formulation.

Аннотация

В выпускной квалификационной работе разработана 2D игра на кроссплатформенной среде разработки Unity3d. Были использованы различные плагины, векторные формы, которые после завершения экспортировались в растровую графику и применялись в разработке.

Mavhum

Ushbu bitiruv malakaviy ishida 2 o'lchovli, ya'ni 2D o'yini Unity3d xoch-platformali muhitda yaratildi. Turli pluginlar, vector shakllari tugatish davrida rast formalariga o'girilgan va o'yin yartish paytida ishlatilgan.

Content

	Introduction	6
Chapter I	Analytical review	8
1.1	Introduction to Game Development	8
1.2	Overview of Development Environment	10
1.3	Specifications	11
1.4	Pros and cons.....	12
1.5	Overview 2D game - «Jelly Run».....	15
	Summary	15
Chapter II	Development environment	16
2.1	Basics	16
2.2	Asset workflow	19
2.3	The main windows	20
2.4	Creating gameplay	24
2.5	Editor features	26
	Summary	30
Chapter III	Projection and development	31
3.1	Creating physical models	31
3.2	Sprites and sprite tools	33
3.3	Scripts	35
3.5	Building app	37
	Summary	39
Chapter IV	Life Safety and environment	41
4.1	What is risk, and what is stressor.....	41
4.2	Working conditions: the concept, the determining factors. Assessment of working conditions.....	44
4.3	Electromagnetic fields and radiation.....	49
	Conclusion	54
	References	55
	Application	56

Introduction

Real-time 3D games have been around for well over ten years now. We've all played them, created assets in the style of our favorites, and maybe even "mod"ed a few of them. But with the Unity game engine leaping ahead in its effort to provide a free or low-cost means of authoring for desktop, mobile, or console games, the only barrier left to creating your own games is your level of commitment and the number of hours you are willing or able to devote. Times have changed. 3D has become affordable not only in the movie industry, as seen by the number of titles featuring CG (computer graphics), but also in the game industry, where we've seen a shift in casual games from 2D to a 3D format. With Unity's bold move to offer a robustly featured free version of its engine, a radical change in the pricing models of the high-end engines has rocked the industry. The cost of the engine is no longer a barrier to taking your game from a nebulous idea to a working prototype and even on to a marketable product. Whether your interest is in casual games or you have more ambitious aims, if you have no previous scripting experience but are eager to bring your art assets and story to life, this may be just the book to get you under way. In today's modern game engines, the heavy programming is handled by the engine itself, so the logic and game play can be scripted by those with more creativity than traditional programming knowledge. If you have the desire to create video games but have no experience with programming or game development, this is the book for you. Unity is a powerful game development ecosystem for creating 2D and 3D games. With its basic but still powerful free version, Unity has blasted away the barriers to learning game development. While the Unity editor is visual and intuitive in nature, you will have to learn to script in order to complete your game.

Scripting, or programming in general, is learned in very basic steps. Chances are you need to overcome some unjustified fears about how amazingly difficult scripting appears to be. The basics of UnityScript really aren't difficult. We do things everyday that are just like the procedures followed in scripting. There are books, manuals, and tutorials galore on programming that can show you how to

really get into the details. After going through the steps in this manual, you should feel a lot more comfortable learning from those tutorials. At least you'll have a basic understanding of what other authors are saying. I'm a beginner of Unity myself. I also know what's it's like to learn other programming languages. I've had to go through the bewilderment just like you. I hope this manual will make clear the concepts of scripting that I've learned. I hope teaching you from a "new-user" perspective will turn on the lights for you, too.

Chapter I

Analytical review

Everyone loves jelly, because they are spread around the world as a candy. Considering this factor occurred to me to create together with the already created similar games, but in Unity3d.

1.1 Introduction to Game Development

Compartmentalizing Environments

Any given game engine, in conjunction with the machine it is running on, can only draw so many objects onscreen at any one time. Additionally, machines or devices can only hold so many textures and meshes in memory at any one time. When asked why there are game levels, a room full of gamers will give any number of reasons—the primary one being as a measure of accomplishment. While that is a valid reason as far as game play and story goes, the real truth is that the technology becomes the limiting factor. Although streaming data may become more prevalent in the future, at this time, we will soon hit a limit as to how much geometry and textures can be loaded into memory. Obviously, the amount will vary with the platform and graphics card, so the first decision is the minimum platform your game should be able to run on. That will serve as a basis for determining how vast each level can be. Think in terms of creating towns or areas of interaction inside box canyons. The steepness and height of the walls will block the view of neighboring clusters of settlements or whatever the assets are. Twisting entries to the areas can block line of sight to other polygon-heavy areas and make occlusion culling possible.

Designing structures works in much the same way. Long, straight vistas where everything must be rendered are far less economical than creating hallways and passages that block the view of all but the immediate area. If you can effectively narrow visibility to smaller spaces, you are free to fill those spaces with more objects, to increase the visual sophistication of the area.

Content

With the technical considerations addressed, you can start thinking about the actual content you would like to have in your game. As with anything else, it bears some critical thinking, if you want your game to be a success.

The Requirements

Having weighed the various technical, artistic, and production considerations, you are probably anxious to get down to details. This is a good time, however, to make a rough list of the basic requirements and decide what is optional and what is mandatory before you get started.

- **Entertaining:** Above all, the adventure game should be entertaining. Whether it comes from humorous interaction, intriguing animation sequences, or intellectual conundrums, game play and environment must draw the player in and keep him engaged.
- **Story/Goal:** The game is not a Big Book of Puzzles. The tasks or puzzles should at least have a logical, mechanical, or physical tie to the world's defined properties (as in the case of magic being part of its description) and, preferably, tie in with the premise of the story.
- **Object interaction:** Interaction with objects is one of the defining features in all aspects of the genre. Whether the player is allowed to gather objects or must only work with what is immediately at hand, the typical point-and-click functionality is a mainstay.
- **Object-to-object interactivity:** A big part of the task-solving component of adventure games is the player's ability to combine multiple objects to attain the key that lets him solve a problem or repair a mechanism in order to move the game forward. While this is not always present in all adventure games, it is a common feature.
- **Conversations:** Player dialogue with NPCs (non-player characters) definitely falls under the bells-and-whistles category, but can be a great addition to

your game. Keep in mind that you don't want the player to miss any clues within them. For topic choices that don't contain clues or pertinent information, be sure to make the replies entertaining, so the player will be rewarded for his diligence.

- **Inventory:** Also not a requirement but a much loved component is the inventory system. In order to allow the player to collect everything that is not nailed down, the game must keep it all readily available for use. Sophisticated or large games should also provide a means of scrolling through the player's stash, in case the onscreen representation of the contents of the inventory is limited.
- **Save/Restore:** Unlike many of today's casual games, even the shorter chapter-style adventure games are rarely played from start to finish in one sitting. Players may devote an hour or so in an evening to playing the game, stopping when they are stuck and starting up again when their mind is fresh. Saving the game is a necessity when the game is expected to give the player several hours of enjoyment over a period of days or weeks.
- **Music and sound effects:** As defined by the early graphical games, sound effects helped reinforce and enhance object interaction. Music not only sets the mood for the individual worlds or environments but is also commonly used as part of the solution to puzzles or tasks in game.

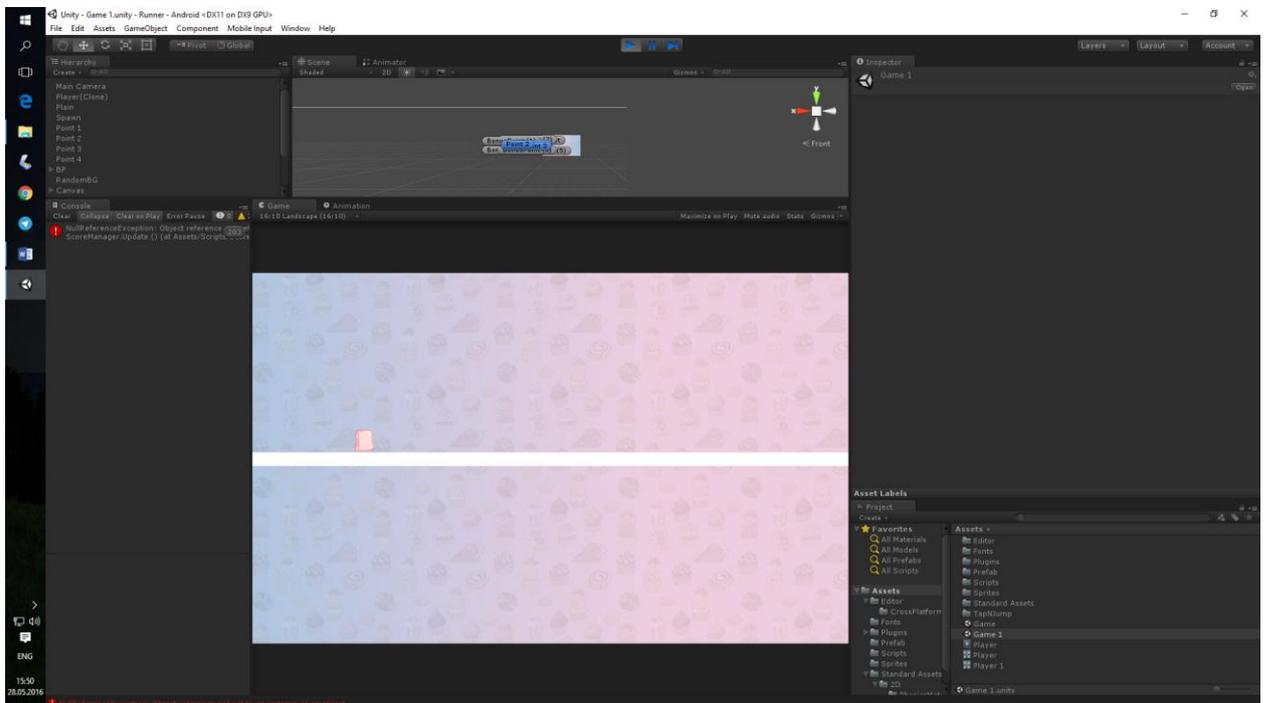
-

1.2 Overview of Development Environment.

Unity - a multi-platform tool for 3D design -games. But it is also possible to write and 2D-games. Unity provides a comprehensive set of professional qualities and applications "quite accidentally" contains the most powerful engine this side of a million dollars. Birth the program may be considered in June 2005, when it was presented to the world the first version (June 2005, Unity 1.0.1.). And since then the development of this product does not stop for a minute. On Currently available version 3, which implements all of the most the latest 3D technology. On Figure 1

shows Unity editor. Number of application users is growing in geometric sequence, 16 February 2010, the number of registered users has exceeded Unity 3D 100 thousand. This engine provides a fast and flexible way of porting its application on virtually any popular operating system. An important feature of the engine, due to which he popular, is its ability to run on any application Internet browser, you simply need to install the plug - Unity Web Player, as well as the possibility of developing for iPhone, iPod touch, iPad, and 3 version adds support for Android.

Figure 1



1.3 Specifications.

List of System Requirements.

For development:

- OS: Windows 7 SP1+, 8, 10; Mac OS X 10.8+(Windows XP & Vista are not supported; and server versions of Windows & OS X are not tested.).
- GPU: Graphics card with DX9 (shader model 2.0) capabilities. Anything made since 2004 should work.

- The rest mostly depends on the complexity of your projects.

Additional platform development requirements:

- iOS: Mac computer running minimum OS X 10.9.4 version and Xcode 6.x.
- Android: Android SDK and Java Development Kit (JDK).
- Windows 8.1 Store Apps / Windows Phone 8.1: 64 bit Windows 8.1 Pro and Visual Studio 2013 Update 2+.
- WebGL: Mac OS X 10.8+ or Windows 7 SP1+ (64-bit editor only)

Generally content developed with Unity can run pretty much everywhere. How well it runs is dependent on the complexity of your project. More detailed requirements:

- Desktop:
 - OS: Windows XP SP2+, Mac OS X 10.8+, Ubuntu 12.04+, SteamOS+
 - Graphics card: DX9 (shader model 2.0) capabilities; generally everything made since 2004 should work.
 - CPU: SSE2 instruction set support.
 - Web Player (deprecated): Requires a browser that supports plugins, like IE, Safari and some versions of Firefox
- iOS: requires iOS 6.0 or later.
- Android: OS 2.3.1 or later; ARMv7 (Cortex) CPU with NEON support or Atom CPU; OpenGL ES 2.0 or later.
- WebGL: Any recent desktop version of Firefox, Chrome, Edge or Safari
- Windows Phone: 8.1 or later
- Windows Store Apps: 8.1 or later

1.4 Pros and cons.

Pros:

- Insanely fast development speeds. It follows agile game creation and facilitates quick prototyping and continuous releases.

- Easy and quick import process of resource subsystem leading to an optimized unified Assets pipeline and supports most image, audio, video and text formats and 3D packages.
- Excellent Integrated Level editor with supporting JavaScript and C# for scripting.
- Superior support for de-bugging and tweaking as all the game variables are displayed during gameplay which enables systematic tweaking and de-bugging processes at runtime.
- Vast Unity3D communities and marketplaces which offer wide built components for sound, physics, rendering, controls etc. The Assets store is very much reminiscent of any Phone App stores.
- Multi-threading, collections, I/O and expressive LINQ library functions using Mono as script host.
- Multi-platform development and deployment platforms for Consoles, Desktops, Browsers and Mobiles.
- Excellent and easy 3D surround/panning implementation and Audio engine.

Cons:

- Hogs more memory, this may cause OOM errors in mobile devices and debugging issues.
- As no source codes are provided, performance issues are hard to find, address and fix.
- Expensive if you need to use its entire features (render textures, stencil buffer support etc).
- Developing a large AAA game title needs a lot of optimizations.
- Additional costs to purchase Mobile Pro licenses for effective deployment.
- Outdated version of mono restricting your game to certain size and constant struggle with the GC.

- Easy to get un-organized with your hierarchy structure or directory and has too many folders, assets etc even with nested items.
- More is desired from the terrain system and better of using other programs for creating and texturing terrain.
- NO support for HTML5 – WebGL for now.
- Switching build targets requires re-importing everything and time taking when working on a multiplatform game.

An overall assessment of UNITY among Game Developers and Game Engine are indeed rated very high and most of us have developed numerous work-around the shortcomings and believe with newer improved versions releases, an excellent Game Assets Stores, relatively low-cost development options and competition from Unreal , CRY will certainly push UNITY to further explore new horizons in game development realm.

1.5 Overview 2D game - «Jelly Run».

This game is a jumper., which should jump over some objects. After that, the player will be scored game points on scoreboard or some counter.

The game engine is written in Unity. This engine it provides a quick and easy programming of 2D and 3D scenes. All cooperation is working out of scripts that are attached to gaming objects. Each object may have several attached scripts. In one frame, the frame, the program performs Update () method in each active script, and redraw the scene depending on the changes.

In the game there are 3 main object:

- Script Storage - it is an empty object, which is not drawn to stage, and to which are attached the game scripts, which can not be associated with a particular game object.
- Main Camera - this is the main and only a camera through the user is looking at the scene.

- Game objects or Sprites - it is necessary objects for creating game. In the game there are the following basic scripts :
- Play
- Player
- Menu
- RandomBG (Random background)
- MoveMonster
- Bonus
- Restart
- Scoremanager
- Spawn
- Exit

Each script contains one or more classes, if MonoBehaviour class inherits from, and can be attached to the game objects, its name must match the name of the script.

Summary

In this chapter we learned exactly how to use a Unity3d and what characteristics are needed on the computer that would have to Use it to create games. Also considered 2D game “JellyRun”. In the following chapters, we learn about the opportunities for Unity3d

Chapter II

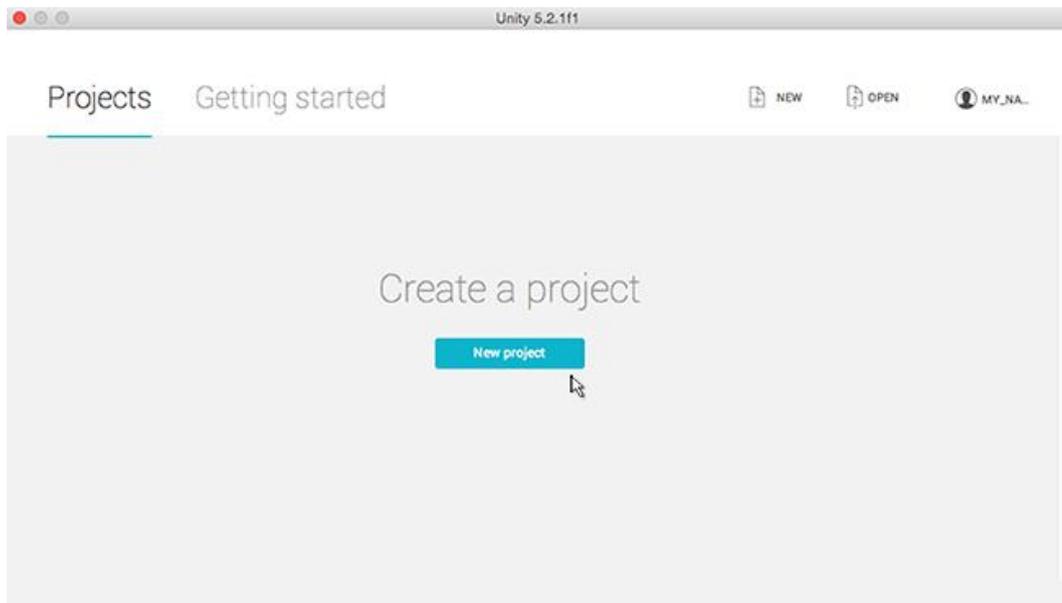
Development environment

2.1 Basics

Starting Unity for the First Time

Whenever you launch the Unity editor, the Home Screen displays. If you have no existing Unity projects on your computer, or Unity doesn't know where they are, it asks you to create a project. To get started, you can click on New project which will take you to the Home Screen's Create Project view. Alternatively, if you already have a Unity project on your computer, you can open it from this screen.

Figure 2. The Home Screen displays on launch, click on 'New project' to get started.



Creating a Project

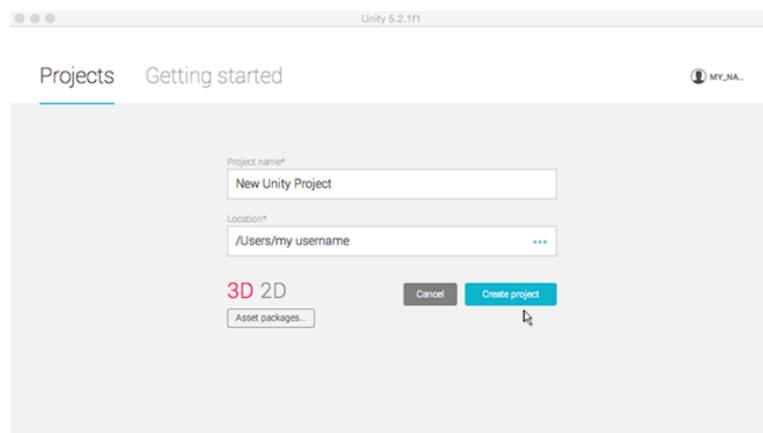
Whenever you start the Unity editor, the Home Screen displays. From here, you can select NEW in the top right corner, to switch to the Create Project view.

Figure 3.



In the top right corner of the Home Screen, select ‘New’ to see the Create Project view. To bring up the Home Screen’s Create Project view when you are already in the Unity editor, select New Project... from the File menu. File>New Project - Displays the Home Screen’s Create Project view from within the Unity editor. From the Home Screen’s Create Project view, you can name, set options, and specify the location of your new project.

Figure 4.

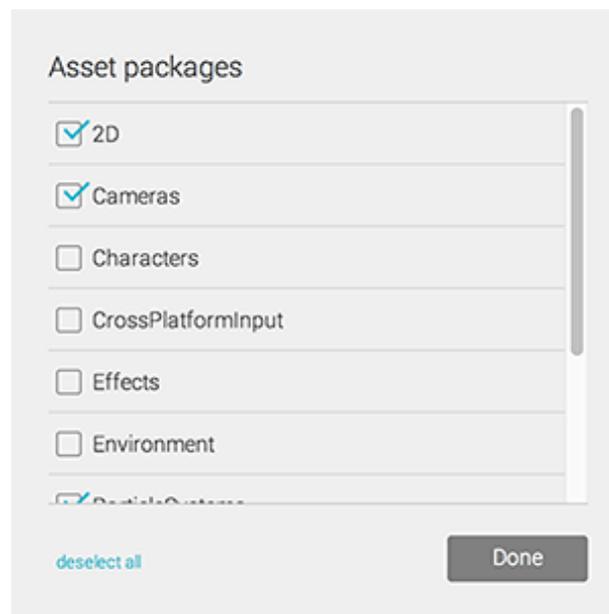


To create a new project:

1. The name defaults to New Unity Project but you can change it to whatever you want. Type the name you want to call your project into the Project name field.
2. The location defaults to your home folder on your computer but you can change it. **EITHER** (a) Type where you want to store your project on your computer into the Location field. **OR** (b) Click on the three blue dots ‘...’. This brings up your computer’s Finder (Mac OS X) or File Explorer (Windows OS).
3. Then, in Finder or File Explorer, select the project folder that you want to store your new project in, and select “Choose”.
4. Select 3D or 2D for your project type. The default is 3D, coloured red to show it is selected. (The 2D option sets the Unity editor to display its 2D

- features, and the 3D option displays 3D features. If you aren't sure which to choose, leave it as 3D; you can change this setting later.)
5. There is an option to select Asset packages... to include in your project. Asset packages are pre-made content such as images, styles, lighting effects, and in-game character controls, among many other useful game creating tools and content. The asset packages offered here are free, bundled with Unity, which you can use to get started on your project. **EITHER:** If you don't want to import these bundled assets now, or aren't sure, just ignore this option; you can add these assets and many others later via the Unity editor. **OR:** If you do want to import these bundled assets now, select Asset packages... to display the list of assets available, check the ones you want, and then click on Done.
 6. Now select Create project and you're all set!

Figure 5.

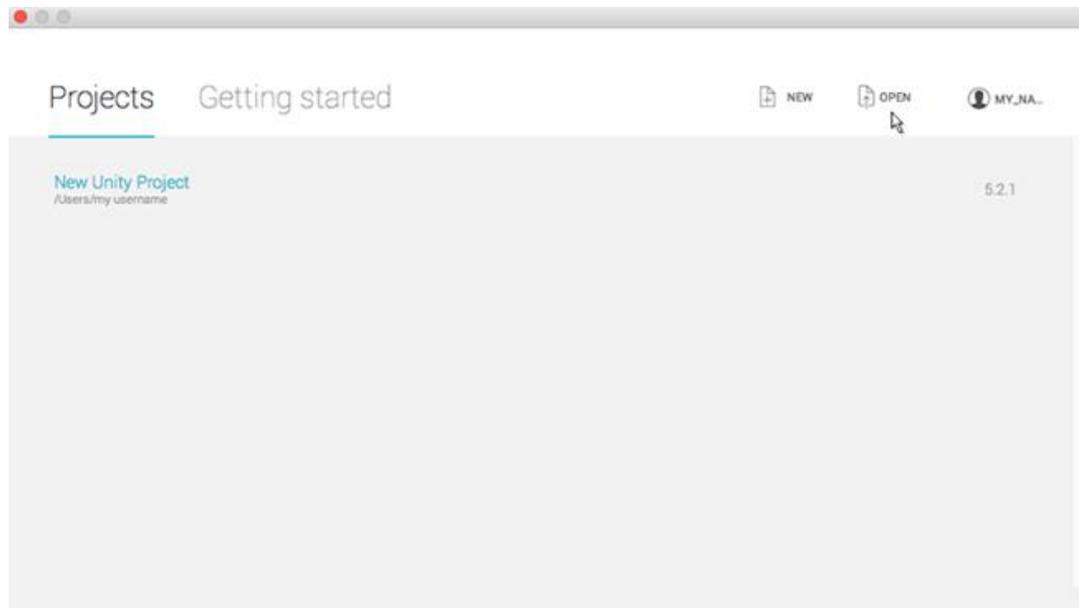


Opening a Project

When you start the Unity editor, the Home Screen's Open Project view displays. From here you can choose the project you want to open. To bring up the Home Screen's Open Project view when you are already in the Unity editor, select Open Project from the File menu. The Home Screen's Open Project view lists all the projects the Unity editor knows about. (If the editor is newly installed

and doesn't know the location of your existing projects, it prompts you to create a new project.

Figure 6.



Click on any of the projects listed to open them. If your project is not listed, you need to tell the editor where it is.

To locate and open an existing project which isn't listed:

1. Select Open. This brings up your computer's Finder (Mac OS X) or File Explorer (Windows OS).
2. In Finder or File Explorer, select the project folder that you want to open and select "Open".

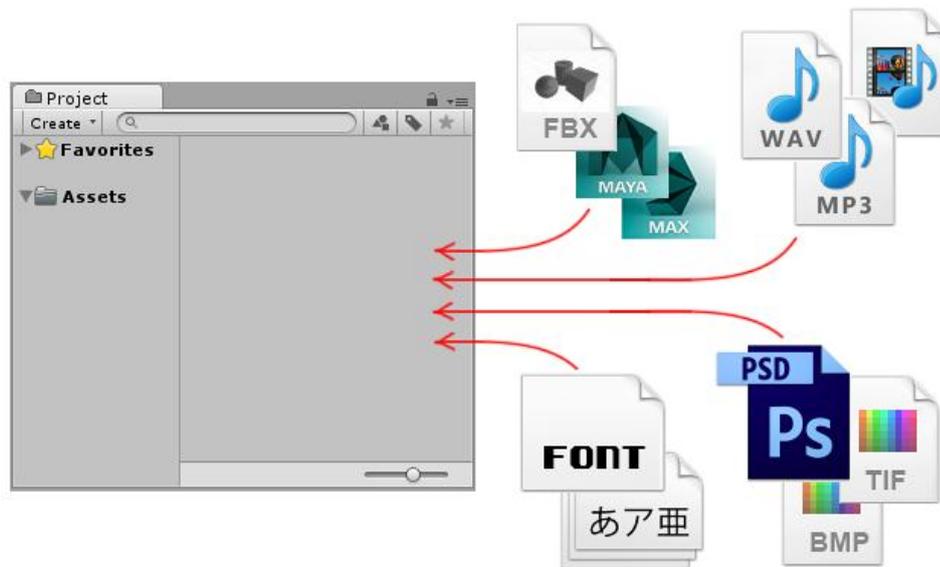
2.2 Asset workflow.

These steps will give you a general overview of the basic principles of working with assets in Unity.

An asset is representation of any item that can be used in your game or project. An asset may come from a file created outside of Unity, such as a 3D model, an audio file, an image, or any of the other types of file that Unity supports.

There are also some asset types that can be created within Unity, such as an Animator Controller, an Audio Mixer or a Render Texture.

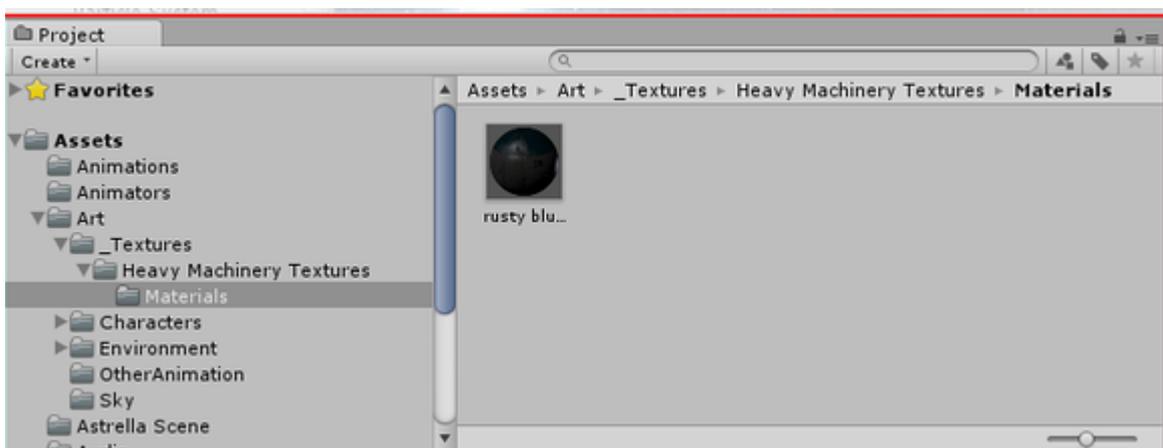
Figure 7.



2.3 The Main Windows

The Project Window

Figure 8.

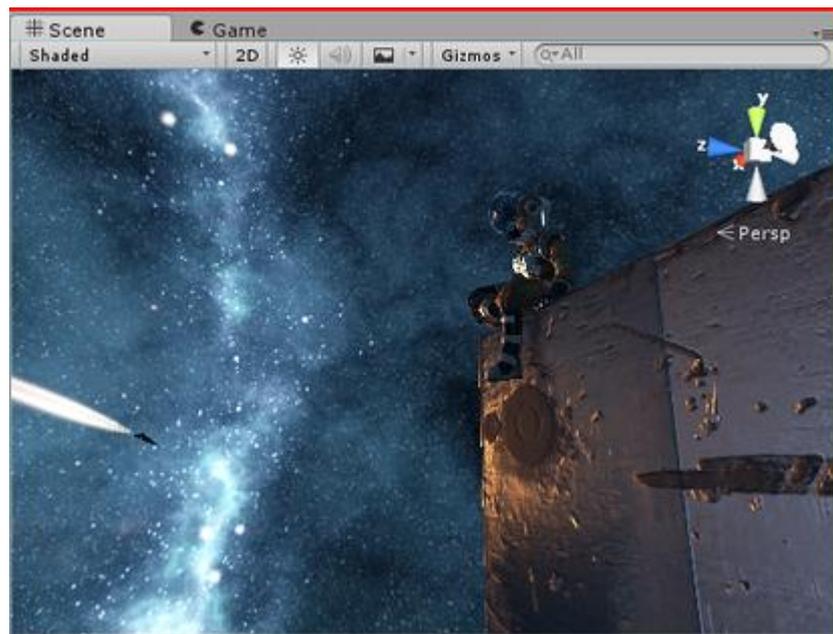


In this view, you can access and manage the assets that belong to your project.

The Scene View

The Scene View is your interactive view into the world you are creating. You will use the Scene View to select and position scenery, characters, cameras, lights, and all other types of Game Object. Being able to Select, manipulate and modify objects in the Scene View are some of the first skills you must learn to begin working in Unity.

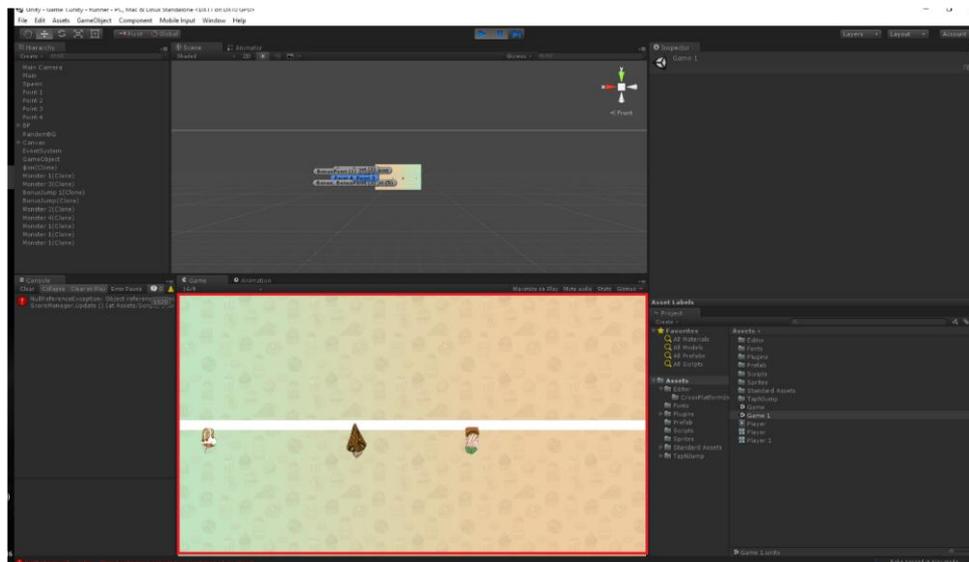
Figure 9.



The Game View

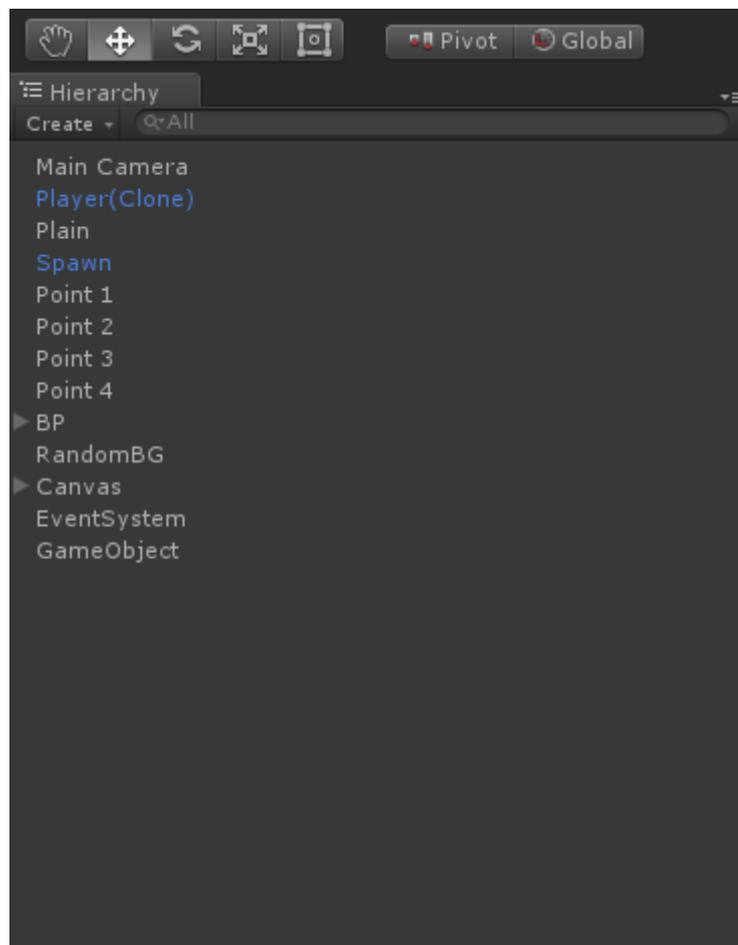
The Game View is rendered from the Camera(s) in your game. It is representative of your final, published game. You will need to use one or more Cameras to control what the player actually sees when they are playing your game. For more information about Cameras, please view the [Camera Component page](#). Use the buttons in the Toolbar to control the Editor Play Mode and see how your published game will play. While in Play mode, any changes you make are temporary, and will be reset when you exit Play mode. The Editor UI will darken to remind you of this.

Figure 10.



The Hierarchy Window

Figure 11.



The Hierarchy contains every GameObject in the current Scene. Some of these are direct instances of asset files like 3D models, and others are instances of Prefabs, custom objects that will make up much of your game. You can select objects in the Hierarchy and drag one object onto another to make use

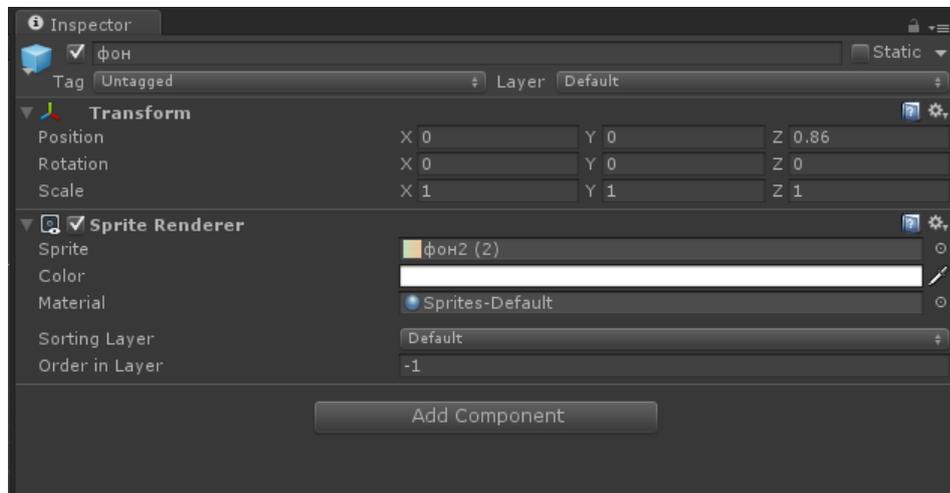
of Parenting (see below). As objects are added and removed in the scene, they will appear and disappear from the Hierarchy as well. By default the GameObjects will be listed in the Hierarchy window in the order they are made. It is possible to re-order the GameObjects by dragging them up or down, or making them children or parents.

Multi-Scene Editing

It is possible to have more than one Scene open in the Hierarchy at the same time.

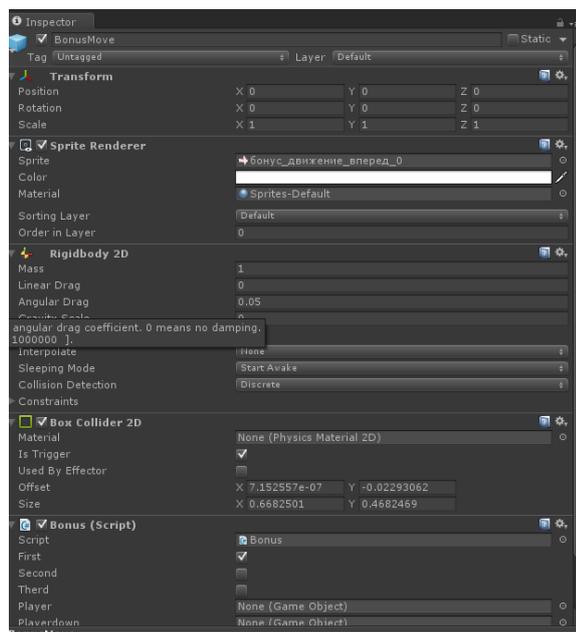
The Inspector Window

Figure 12.



Inspecting Game Objects.

Figure 13.



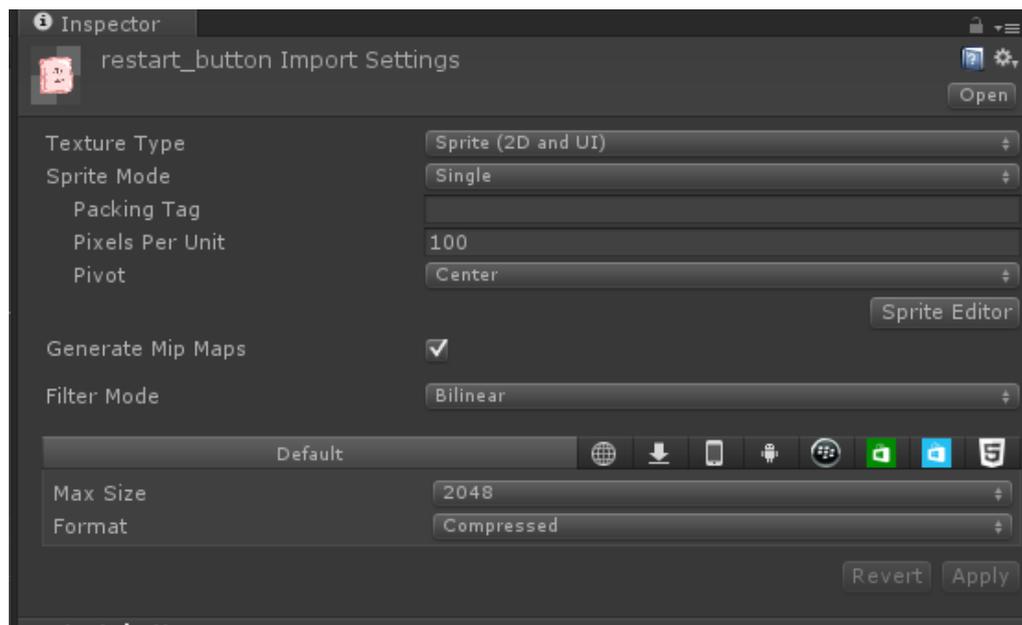
The Inspector is used to view and edit the properties and settings of Game Objects, Assets, and other preferences and settings in the Editor.

When you select a GameObject in the Hierarchy or Scene View, the Inspector will show the Properties of all Components and Materials on that object and allow you to edit them. The image above shows the inspector with the default 3D camera GameObject selected. In addition to the object's position, rotation and scale values, all the properties of the camera are available to edit.

Inspecting Script Variables

When Game Objects have custom Script components attached, the public variables of that script are also shown in the inspector and can be viewed and edited like the properties of Unity's built-in components. This allows you to set parameters and default values in your scripts easily without modifying the code

Figure 14.



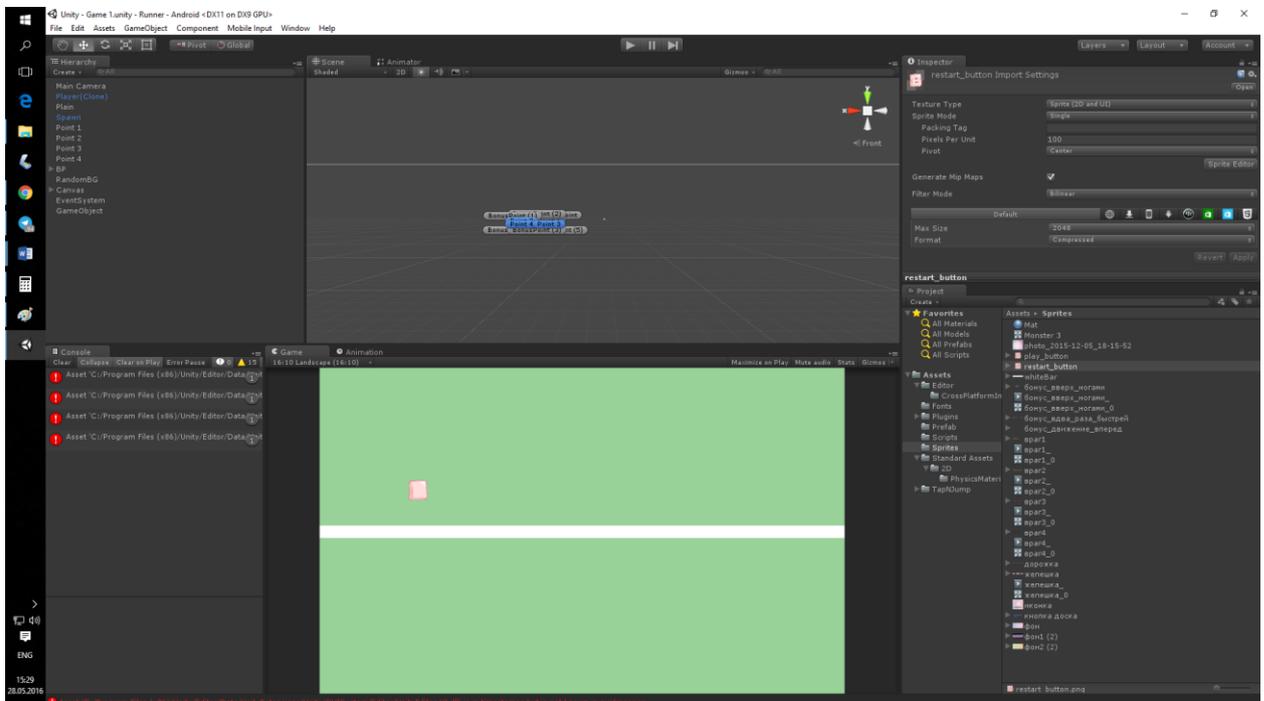
2.4 Creating Gameplay

Unity empowers game designers to make games. What's really special about Unity is that you don't need years of experience with code or a degree in art to make fun games. There are a handful of basic workflow concepts needed to learn Unity. Once understood, you will find yourself making games in no time. With the

time you will save getting your games up and running, you will have that much more time to refine, balance, and tweak your game to perfection.

This section will explain the core concepts you need to know for creating unique, amazing, and fun gameplay. The majority of these concepts require you to write Scripts. For an overview of creating and working with Scripts, read the [Scripting](#) page.

Figure 15.

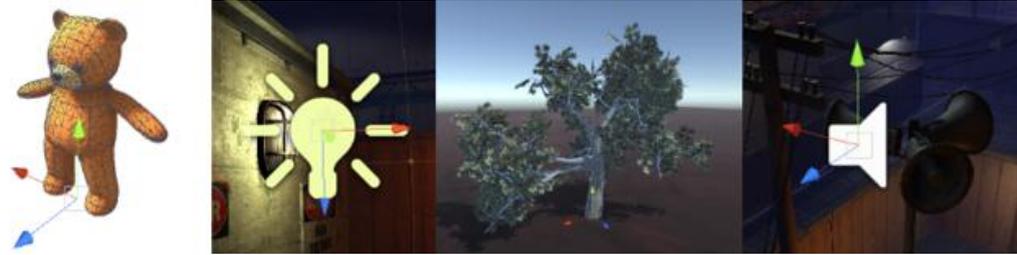


GameObjects

The GameObject is the most important type of object in Unity. It is very important to understand what a GameObject is, and how it can be used.

What are GameObjects?

Figure 16.



The answer to this question is that GameObjects are containers. They can hold the different pieces that are required to make up a character, a light, a tree, a sound, or whatever else you would like to build. So to really understand GameObjects, you have to understand these pieces which are called **Components**.

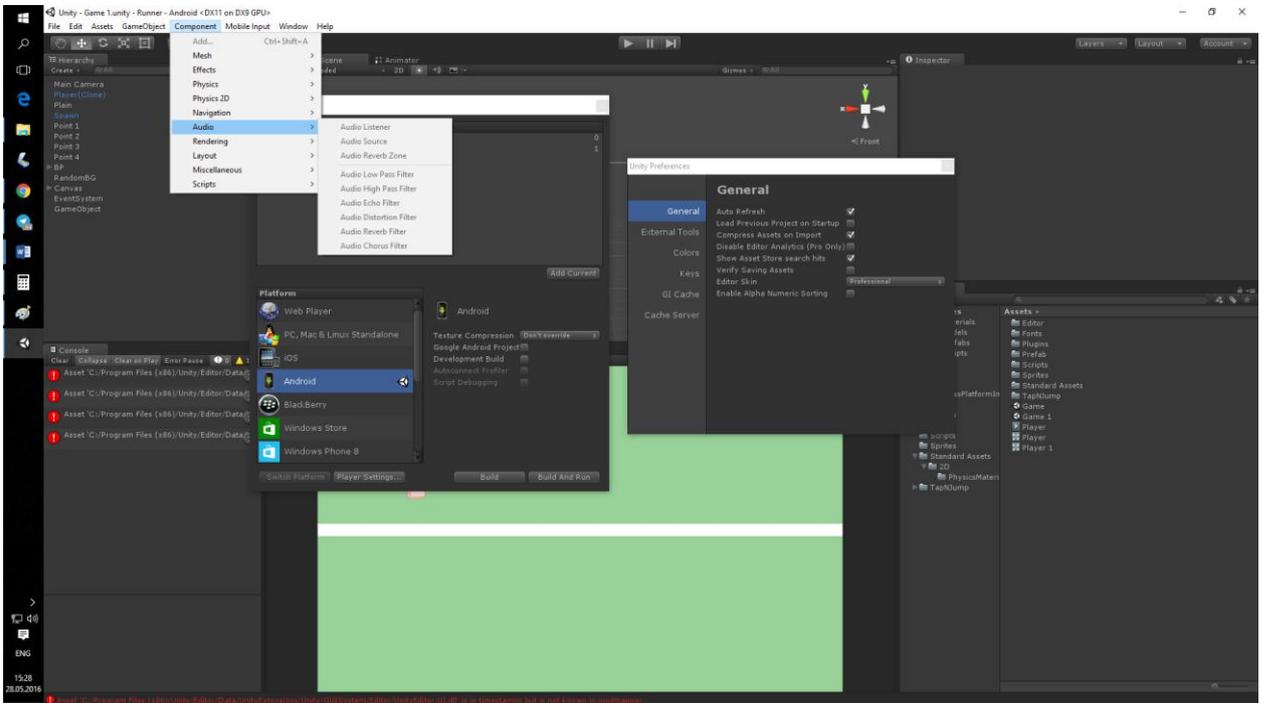
Depending on what kind of object you want to create, you will add different combinations of Components to the GameObject. Think of a GameObject as an empty cooking pot, and Components as different ingredients that make up your recipe of gameplay. Unity has lots of different built-in component types, and you can also make your own Components using Scripts.

This section explains how Game Objects, Components and Scripts fit together, and how to create and use them.

2.5 Editor Features

This section details some of the editor's basic features, which you will find useful in most projects - from choosing preferences, integrating with version control system, to preparing your project for a build.

Figure 17.



2D and 3D Mode Settings

When creating a new project, you can specify whether to start the Unity Editor in 2D mode or 3D mode. However, you also have the option of switching the editor between 2D Mode and 3D Mode at any time.

Switching Between 3D and 2D Modes

To change modes between 2D or 3D mode:

1. Bring up to the Editor Settings Inspector, via the Edit>Project Settings>Editor menu.
2. Then set Default Behavior Mode to either 2D or 3D.

Figure 18.

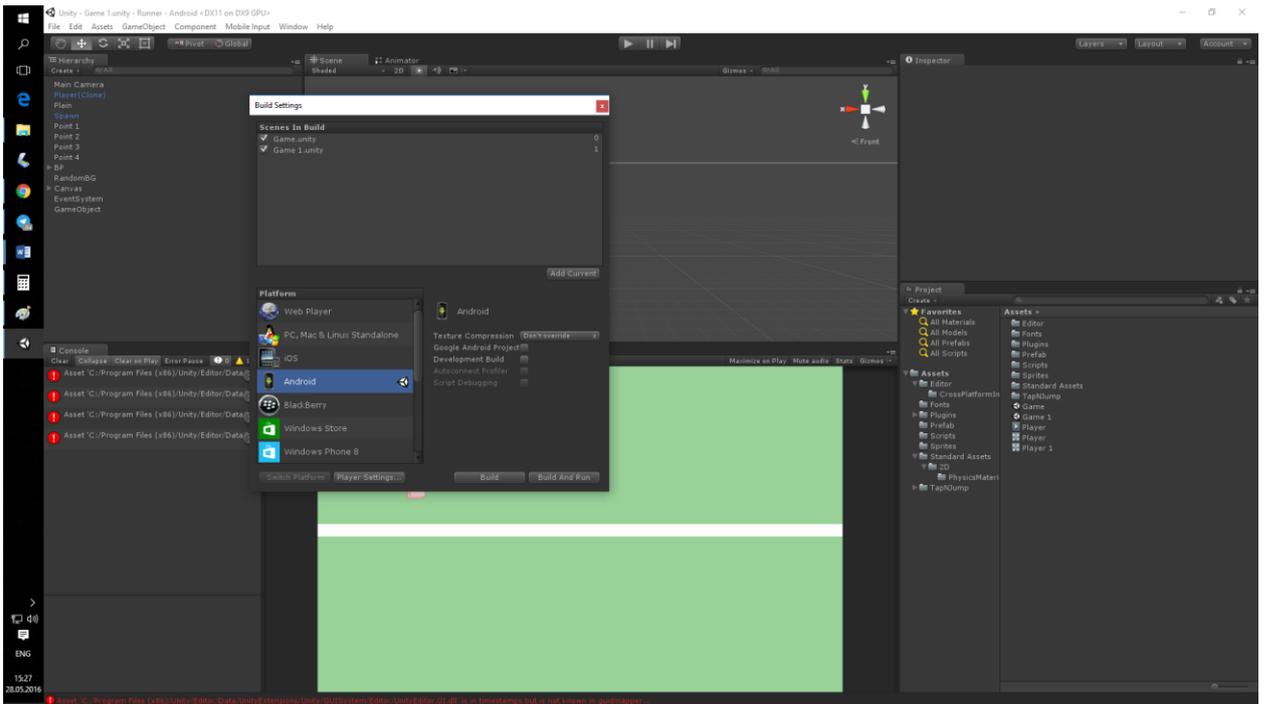
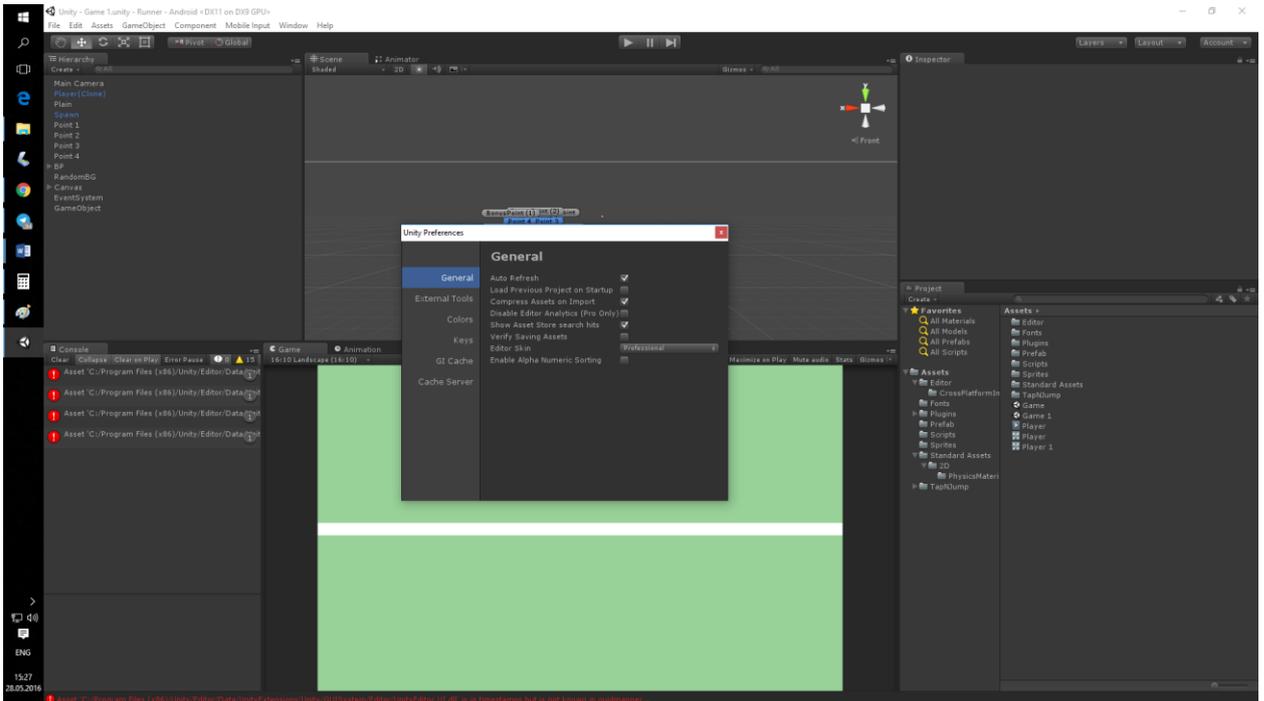


Figure 19.



Build Settings

The Build Settings window allows you to choose your target platform, adjust settings for your build, and start the build process. To access the Build Settings window, choose “Build Settings” from the File menu. Once you have specified

your build settings, you can click “Build” to create your build, or click the “Build and Run” to create and run your build on the specified platform (Figure 2.).

A few things to watch out for:

- Even though Visual Studio comes with its own C# compiler, and you can use it to check if you have errors in your c# scripts, Unity still uses its own C# compiler to compile your scripts. Using the Visual Studio compiler is still quite useful, because it means you don't have to switch to Unity all the time to check if you have any errors or not.
- Visual Studio's C# compiler has some more features than Unity's C# compiler currently has. This means that some code (especially newer c# features) will not give an error in Visual Studio but will give an error in Unity.
- Unity automatically creates and maintains a Visual Studio .sln and .csproj file. Whenever somebody adds/renames/moves/deletes a file from within Unity, Unity regenerates the .sln and .csproj files. You can add files to your solution from Visual Studio as well. Unity will then import those new files, and the next time Unity creates the project files again, it will create them with this new file included.
- Unity does not regenerate the Visual Studio project files after an AssetServer update, or a SVN update. You can manually ask Unity to regenerate the Visual Studio project files through the menu: Assets->Sync MonoDevelop Project

Assets

The **Assets** folder is the main folder that contains the assets that can be used by a Unity project. The contents of the Project view correspond directly to the contents of the Assets folder. Most API functions assume that everything is located in the Assets folder and so they don't require it to be mentioned explicitly.

However, some functions *do* need to have the Assets folder included as part of a pathname.

Standard Assets

When you import a standard asset package (menu: Assets > Import Package) the assets are placed in a folder called **Standard Assets**. As well as containing the assets, these folders also have an effect on script compilation order;

Figure S1.

Summary

We've made great strides in this chapter. we understand the structure of what makes up a scene, and the one necessary game object, the Main Camera. We know that game objects are containers for components, and how to find the component quickly in Unity's Component Reference. We are familiar with the default components of Unity's primitive shape Game Objects, Cameras, and Lights. We know how to import assets and add some color to our scene with skyboxes and to our game object with materials.

Chapter III. Projection and development

3.1 Creating physical models

At this point I'm using Adobe Illustrator to create physical models in the vector.

Adobe Illustrator is a professional quality graphic art program. Its applications are diverse, from creating print to web graphics and, for our purposes, posters. Although the program is complex and requires a significant amount of time to truly master, it is quite easy to learn the basics and create work with a professional appearance. Designing a poster does not require any special artistic talent or skills; it just requires having the patience to get acquainted with the program. As with most computer applications, I recommend using the trial-and-error methodology to figure out the intricacies of this program. You will make numerous mistakes, and frequently want to make changes. Thankfully Illustrator is equipped with an undo option. In fact, undo [Crtrl+Z] will become a very familiar action as you design your poster.

I drew all the elements in this program, and then save them in PNG format without a background with the desired me permission.

Figure 24.

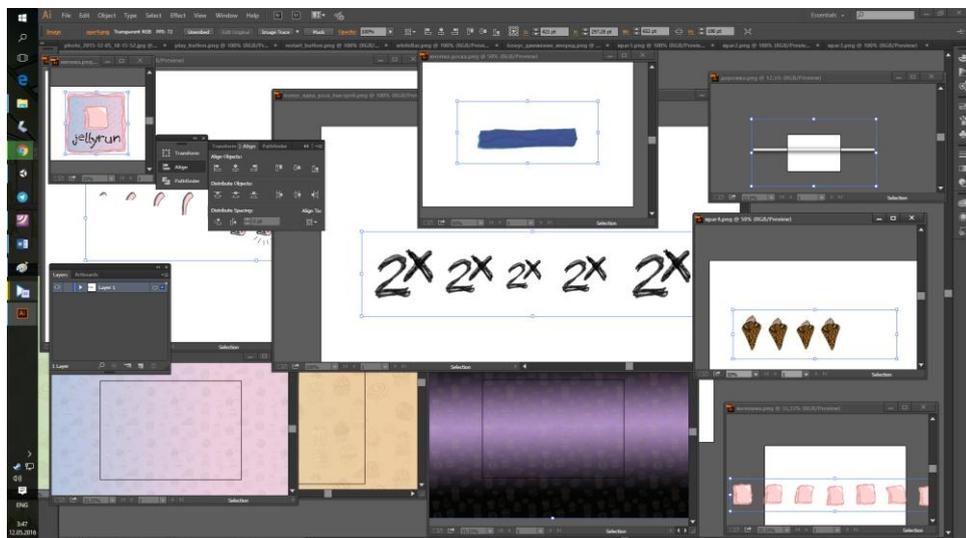
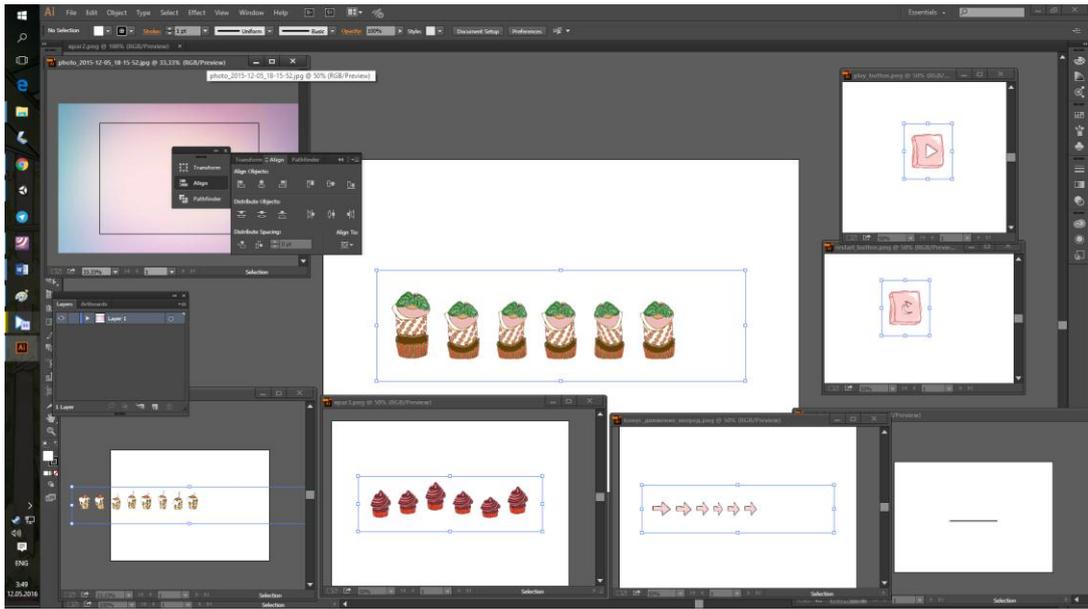


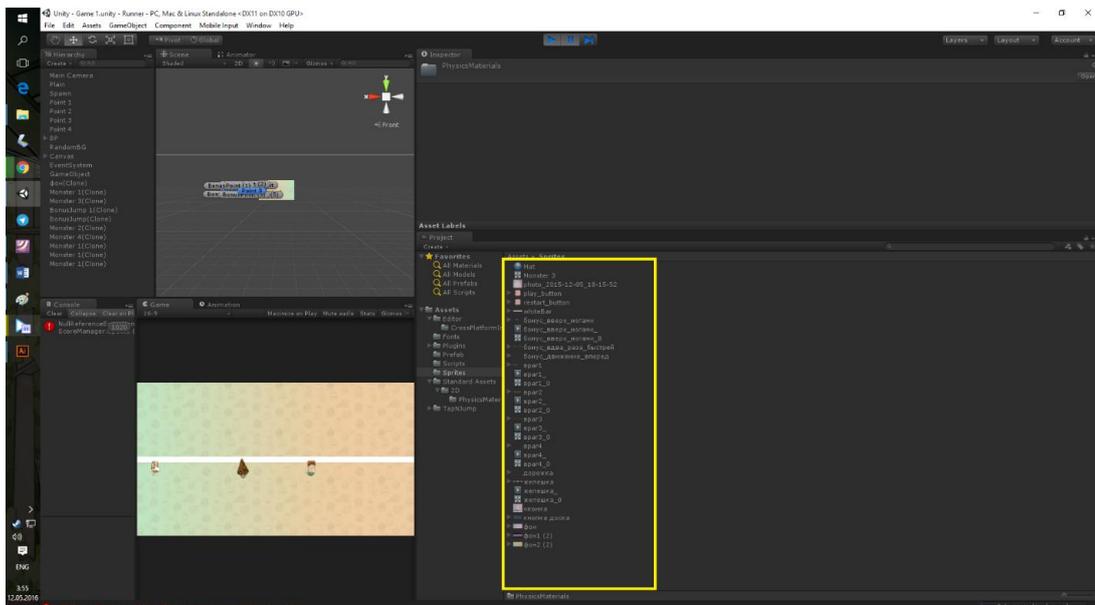
Figure 25.



After saving we must move saved files to assets directory as
... \Runner\Assets\Sprites.

Now we can see saved files in our Project window.

Figure 26.



3.2 Sprites and sprite tools

Sprites

Sprites are 2D Graphic objects. If you are used to working in 3D, Sprites are essentially just standard textures but there are special techniques for combining and managing sprite textures for efficiency and convenience during development.

Unity provides a placeholder Sprite Creator, a built-in Sprite Editor, a Sprite Renderer and a Sprite Packer

Sprite Tools

Sprite Creator

Use the Sprite Creator to create placeholder sprites in your project, so you can carry on with development without having to source or wait for graphics.

Sprite Editor

The Sprite Editor lets you extract sprite graphics from a larger image and edit a number of component images within a single texture in your image editor. You could use this, for example, to keep the arms, legs and body of a character as separate elements within one image.

Sprite Renderer

Sprites are rendered with a Sprite Renderer component rather than the Mesh Renderer used with 3D objects. Use it to display images as Sprites for use in both 2D and 3D scenes.

Sprite Packer

Use Sprite Packer to optimize the use and performance of video memory by your project.

Importing and Setting Up Sprites

Sprites are a type of Asset in Unity projects. You can see them, ready to use, via the Project View.

There are two ways to bring Sprites into your project:

1. In your computer's Finder (Mac OS X) or File Explorer (Windows), place your image directly into your Unity project's Assets folder.

Unity detects this and displays it in your project's Project View.

2. In Unity, go to Assets>Import New Asset... to bring up your computer's Finder (Mac OS X) or File Explorer (Windows).

From there, select the image you want, and Unity puts it in the Project View.

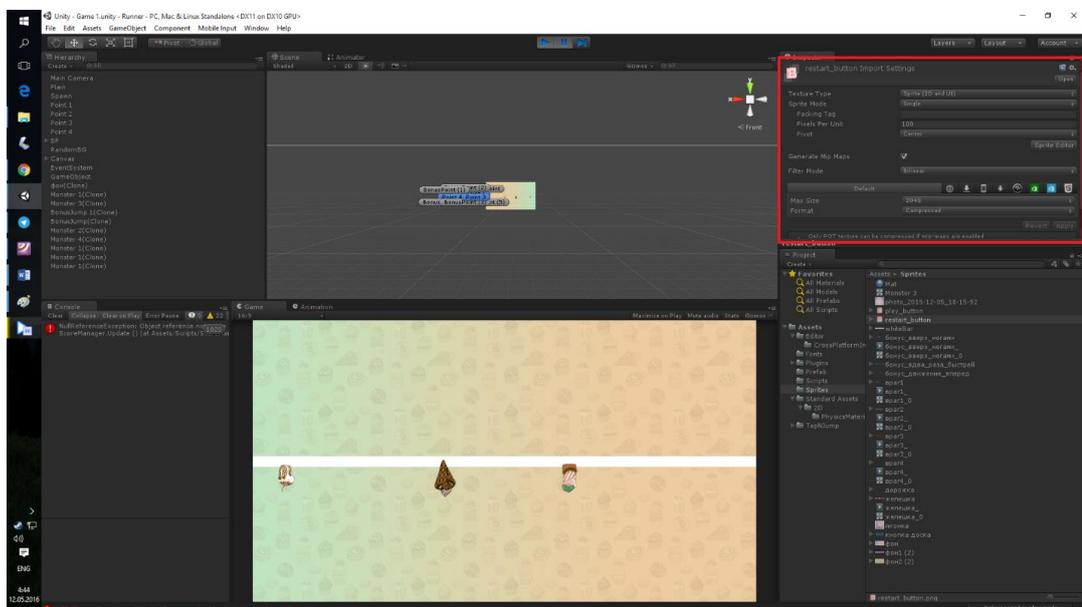
Setting your Image as a Sprite

If your project mode is set to 2D, the image you import is automatically set as a Sprite.

However, if your project mode is set to 3D, your image is set as a Texture, so you need to change the asset's Texture Type:

1. Click on the asset to see its Import Inspector.
2. Set the Texture Type to Sprite (2D and UI).

Figure 27.



After setup, we need to assign the dynamics and the role of sprites.

3.3 Scripts

Scripting is an essential ingredient in all games. Even the simplest game will need scripts to respond to input from the player and arrange for events in the gameplay to happen when they should. Beyond that, scripts can be used to create graphical effects, control the physical behaviour of objects or even implement a custom AI system for characters in the game.

Scripting is a skill that takes some time and effort to learn; the intention of this section is not to teach you how to write script code from scratch but rather to explain the main concepts that apply to scripting in Unity.

Creating and Using Scripts

The behavior of GameObjects is controlled by the Components that are attached to them. Although Unity's built-in Components can be very versatile, you will soon find you need to go beyond what they can provide to implement your own gameplay features. Unity allows you to create your own Components using scripts. These allow you to trigger game events, modify Component properties over time and respond to user input in any way you like. Unity supports two programming languages natively:

- **C#** (pronounced C-sharp), an industry-standard language similar to Java or C++;
- **UnityScript**, a language designed specifically for use with Unity and modelled after JavaScript;

In addition to these, many other .

Learning the art of programming and the use of these particular languages is beyond the scope of this introduction. However, there are many books, tutorials and other resources for learning how to program with Unity.

Creating Scripts

Unlike most other assets, scripts are usually created within Unity directly. You can create a new script from the Create menu at the top left of the Project

during gameplay. To enable the Update function to do its work, it is often useful to be able to set up variables, read preferences and make connections with other GameObjects before any game action takes place. The **Start** function will be called by Unity before gameplay begins (ie, before the Update function is called for the first time) and is an ideal place to do any initialization.

Note to experienced programmers: you may be surprised that initialization of an object is not done using a constructor function. This is because the construction of objects is handled by the editor and does not take place at the start of gameplay as you might expect. If you attempt to define a constructor for a script component, it will interfere with the normal operation of Unity and can cause major problems with the project.

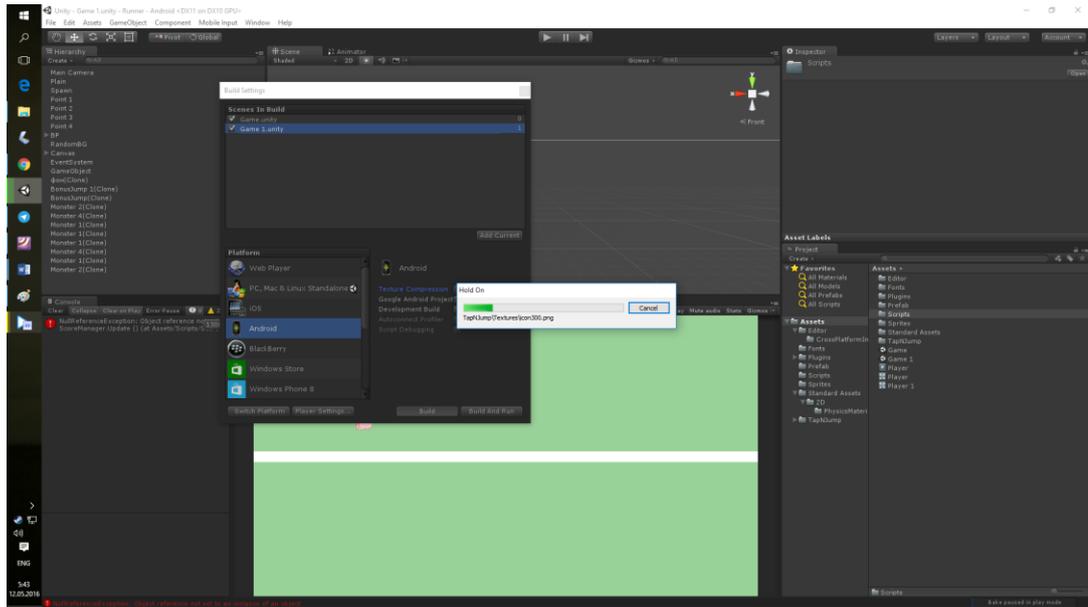
Accessing Components

The simplest and most common case is where a script needs access to other Components attached to the same GameObject. As mentioned in the Introduction section, a Component is actually an instance of a class so the first step is to get a reference to the Component instance you want to work with. Typically, you want to assign the Component object to a variable, which is done in C# using the following syntax: (Figures S2 - S7)

3.4 Building application

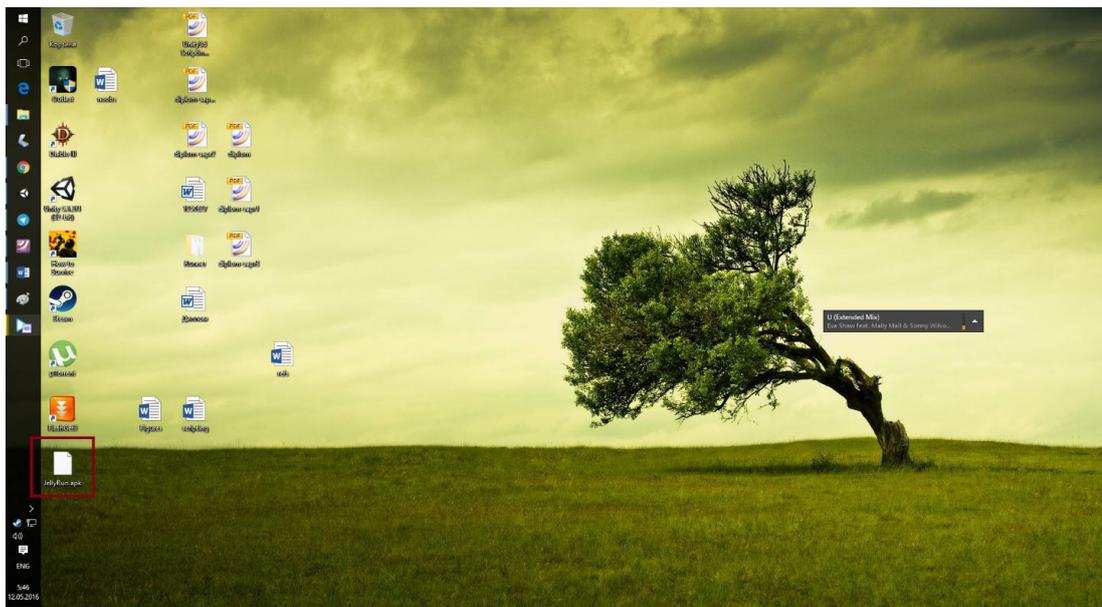
After work on errors, we will be able to see that something like this game is born, it remains only to compile.

Figure 31.



At last JellyRun is ready to play!

Figure 32.



Summary

You've certainly come a long way in one chapter. Unity provided you with some common scripts in the Standard Assets package as a starting point. You understand the structure of gameplay and the life cycle of a game object from `Awake()` to `Destroy()`, and you got a little introduction to optimization and code

refactoring. The game player has appeared and interacted with your game. You can create scripts and attach them to game objects, and you've picked up a few Unity classes and functions to use on your game objects. In short—you are scripting! The next chapter is all about adding action to the game, starting with simple game object

movements and continuing on to an introduction to Unity's Mecanim animation system. This is how you put the “action” into “action-packed”!

CHAPTER 4. LIFE SAFETY AND ECOLOGIES.

4.1. What is risk, and what is stressor

Risk is the probability characteristic of the threat that arises in this case for the environment (and human) for possible anthropogenic effects or other phenomena or events. The system of environmental risk assessment any effect (whether chemical factor or energy field), cause changes in biological systems (both positive and negative) is called a stressor. In this sense, any ecotoxicants - certainly a stressor.

What is risk assessment: The concept of risk assessment involves two elements: risk assessment (Risk Assessment) and risk management (Risk Management).

Risk assessment - is the scientific analysis of origin, including the identification, determination of the degree of danger in a particular situation. The Applied Ecology concept of risk associated with hazards for ecological systems and processes taking place in them. For environmental indicators damage (environmental risk) in this case are: the destruction of biota, harmful, sometimes irreversible impact on the ecosystem, environmental degradation associated with its pollution, increasing the likelihood of specific diseases, the alienation of land, destruction of forests, lakes, rivers, Seas (eg Aral), etc.

What is risk management: Risk management - an analysis of the risk situation, the development and support of administrative decision, usually in the form of regulation aimed at reducing the risk of finding ways to reduce the risk.

The development of the theory of risk has led to the formation of consistent principles that characterize the attitude of the society to ensure trouble-free operation of man-made objects - sources of environmental hazards:

- The principle of zero risk - reflects the confidence that the risk will not be applied;
- The principle of successive approximation to absolute safety, ie a zero risk, involving the study of certain combinations of alternative structures, technologies, etc .;
- The principle of minimal risk, according to which the danger level is set as low as is realistically achievable on the basis of the justification of any cost to protect human rights;
- A balanced risk, which are accounted for in accordance with various natural hazards and anthropogenic impacts, studied the risk of each event and the conditions in which people are at risk;

- The principle of acceptable risk, based on an analysis of relations "cost-risk", "risk-benefit", "cost-benefit". The concept of acceptable risk comes from the fact that the complete elimination of the risk or virtually impossible or economically unfeasible. In accordance with the established rational safety at which optimizes the cost of the prevention of risk and the extent of damage in case of emergencies.

Stages of risk assessment: The first step (step) risk assessment is hazard identification - determination of real danger to human environment. Here the big role of scientific research. Trying to identify the danger is reduced to finding danger signals, allocation of such a signal on an existing background.

For the hazard identification are important methods of testing, selection (such as various products), modeling the behavior of various substances in the environment, monitoring and diagnosis (symptom assessment, impacts). It should be noted that all the issues assessment, diagnosis and prognosis should be referred to the monitoring system. Diagnosis begins with the observation of deviations - to these deviations need to properly define "disease." Almost all the data obtained by monitoring demand estimates mostly diagnostic.

When hazard identification is the first question, what is the danger in the calculation of risk, what is its value, ie, necessary to determine the probability of occurrence of the hazard and the likelihood of adverse effects. To determine the risk calculation can be used foresight, intuition and extrapolation.

On the stage of the procedure under consideration of risk assessment analysis carried out on a *qualitative level*.

The second phase - exposure assessment - an assessment of the ways in which and through which the medium on which the quantitative level, at what time and at what length of exposure there is a real and anticipated exposure; is an assessment of doses received, if it is available, and evaluation of the number of people who are exposed to such exposure and for which it seems likely.

The number of exposed population is one of the most important factors for deciding the priority of protective measures, which occurs when using the results of the risk assessment for the purpose of "risk management".

Ideally, the assessment of exposure based on actual monitoring data of pollution of the various components of the environment (air, indoor air, soil, drinking water, food). Often, however, this approach is not feasible due to the high costs. In addition, it is not always possible to evaluate the association of pollution with a specific source and it is not sufficient to predict future exposure. Therefore, in many cases, use different mathematical models of dispersion of emissions, their settling on the ground, diffusion and dilution of pollutants into groundwater and / or surface waters.

The third stage - evaluation of "dose-effect" - is **the search for quantitative laws** that connect receives a dose of the substance to the prevalence of a negative (health) effects, ie the probability of its development.

Such laws are usually identified in the toxicological experiments. However, extrapolation of a group of animals in the human population is associated with too many uncertainties. Dependence "dose-response" reasonable epidemiological data, more reliable, but it has its own area of uncertainty.

Evaluation phase "dose-effect" is fundamentally different for carcinogens and non-carcinogenic.

For non-carcinogenic toxic substances methodology is based on the concept of threshold actions and recognizes possible to establish so-called "reference dose» (RED) or reference concentration (RFC), which under the action of the human population, including sensitive subgroups it does not create a risk of any perceptible deleterious effects during the entire lifespan. A similar concept is in some WHO documents - "tolerable intake of» (tolerable intake - TI).

In assessing the dependence "dose-effect" for carcinogens, the effect of which is always considered as having no threshold, preference is given to the so-called linearized multistage model (linearized multistage model). The model chosen as the basis of a unified approach to extrapolate from high to low doses. This basic parameter for the calculation of the risk to human health is the so-called factor of inclination (slope factor), which is usually used by 95% upper confidence limit of the slope of the "dose-response". Slope factor expressed in (mg / (kg · day)) - 1 is a measure of the risk occurring per unit dose of carcinogen. For example, if a person is exposed to daily lifelong exposure to the carcinogen dose of 0.02 (mg / (kg · day)) - 1, the added risk, is obtained by multiplying the dose by a factor of tilt is estimated 4×10^{-5} . In other words, it is likely development of four additional cases of cancer per 100 000 people., Exposed to this level of exposure.

The final stage of risk assessment procedures - risk **characterization** - is the result of the previous stages and includes an assessment of potential and known adverse effects of human health; risk assessment of carcinogenic effects, the establishment of the hazard ratio of general toxic effects, the analysis and characterization of the uncertainties associated with the assessment and compilation of all the information on risk assessment.

The magnitude of risk is defined as the product of the probability of damage I Events W_i , causing the damage:

$$R = I W_i.$$

As the risk assessment procedure is complex and largely suffering known uncertainty, in order to standardize research agency US Environmental Protection (EPA) has developed and approved a plan to carry out such work. It contains a

description of the sequence of solving the problem, accounting uncertainties and assumptions in order to obtain a certain degree of uniform approximate information on the likelihood of adverse environmental effects.

The plan includes an environmental risk assessment stages (Fig. 4):

1. Formulation of the problem and develop a plan of analysis of the situation.
2. Analysis of the environmental situation.
3. Processing of the data, forming conclusions and presenting materials to the customer.

As a rule, environmental risk assessment carried out in the form of contract research performed in order to obtain information of a prospective or retrospective nature and necessary to the customer (legal, administrative structures, etc.) for administrative decision-making. Therefore, unlike scientific ecotoxicological studies, during which considered the objective laws of reactions to the action of the stressor ecological community in determining ecotoxicological risk environment as objects to be studied and "protection", the characteristics of biological systems can act with anthropocentric value and sometimes individual elements of the human nature, the perceived public opinion, as relevant.

Summing up: Methodology for environmental risk assessment is not fully developed. In most cases, its findings are qualitative, descriptive. Attempts to introduce methods for quantifying face serious difficulties. This is due to the complexity of ecosystems, the complex influence of stressors on the environment (not only chemical but also the physical and biological nature), insufficient knowledge of the danger characteristics of Ecotoxic huge amount of xenobiotics used by man, etc. In this context, in the opinion of environmentalists, is currently an environmental risk assessment is largely an art.

4.2. Working conditions: the concept, the determining factors. Assessment of working conditions

Introduction: In modern conditions of market economy in our country has placed new demands on the organization of work in the enterprise. In this regard, an important issue is the establishment of effective working conditions. Therefore, in practice, work on the scientific management to improve working conditions consumes about one quarter of all planned expenses.

Of great importance to improve working conditions is due to the fact that they are basically working environment in which human activity occurs during labor. Their status is in direct proportion to the level of human performance, the results of his work, health, attitude to work. Improving working conditions

significantly affect the increase in its productivity. Therefore, in practice, the cost of implementing them pay off an average of 3 - 5 years.

Performing any work for a long time the body is accompanied by fatigue, manifested in the reduction of human health. Along with the physical and mental work a significant impact on fatigue and provides the surrounding work environment, i.e. the conditions in which it takes work.

Based on the foregoing, it is possible to determine the purpose of the course work - the study of the basic directions of improvement of working conditions and the determination of their economic efficiency.

To achieve this goal it is necessary to perform a number of tasks: 1) to give a definition of the concept of working conditions; 2) Describe the factors determining the conditions of work; 3) consider the main methods of assessment of working conditions; 4) to analyze the main directions of improvement of working conditions; 5) to consider the mechanism of calculating the cost-effectiveness of measures to improve working conditions.

Course work consists of two chapters: the first chapter contains theoretical information about working conditions, their elements; The second chapter is devoted to methods and ways to improve working conditions.

What is working conditions: Working conditions - a combination of environment factors influencing the functional state of the body work, their health and performance at work. They are determined using equipment, technology, objects and products of labor, the system of protection of workers, service jobs and external factors which depend on the state of the production facilities that create a certain climate. Thus, based on the nature of the work, the working conditions are specific for each production plant and site, and for each workstation.

Working conditions - is a complex objective social phenomenon emerging in the labor process under the influence of inter-related factors of socio-economic, technical and organizational nature and natural character and influence health, human performance, his attitude to work and the degree of job satisfaction, the effectiveness of labor and other economic results of production, the level of life and comprehensive development of man as the main productive forces of society.

This definition gives a complete description of the working conditions: their essence as an objective phenomenon, the formation mechanism and guidelines on influence working person, effectiveness, impact on social development. This definition concretize term "work environment" and "conditions work" represents the unity of the two sides. On the one hand, the factors affecting generation working conditions, and on the other – elements, making up working conditions. The elements concern working conditions, such as temperature, gas content, and so

on. E., That is, everything that directly affects the working man, his health, performance and social development.

Factors determining working conditions: According to the authors of works of VP Pashuto and VV Adamchuck, the factors shaping the working conditions can be divided into the following groups:

The production, among which are: psychophysiological, sanitation, aesthetic;

Production factors - is the most extensive group of factors leading features of this specific form of production and working conditions. Among them are several sub-groups: physiological, hygienic, aesthetic, and some other (Household, organizational, material and so on.).

Psychophysiological factors due to the content of labor and its organization, so they are sometimes called labor. They may also be called technical-technological, t. To. They are determined by the peculiarities of used machinery and technology, the level of mechanization and automation of labor, equipment degree jobs, especially raw materials. It - physical activity, which is associated with dynamic and static operation; neuro-psychic load in the form of voltage (accuracy), neuro-emotional stress and intellectual load (the amount of information processed, the number of one-off production of important objects of observation, etc.); the monotony of the work process (diversity, the pace of labor). The elements of this group, with the exception of physical effort and monotony do not have the approved standards.

Currently, there are several psychophysiological approaches to integrated assessment of the working conditions.

Job analysis approach is to study the content of the labor process, and the result of the work of the unit is based on the estimation of the overall working conditions.

Another approach involves the classification of working conditions on the severity and intensity on the basis of physiological and psycho-physiological parameters, dynamics and fatigue performance, reflected in one way or another body's response to the workload and working conditions.

The severity of labor - characteristics of the labor process, reflecting the strain on the musculoskeletal system and functional systems (cardiovascular, respiratory, etc.) To ensure its activities.

Consider the classification of the severity of labor. According to the medical and physiological skills developed Institute of Labor on the basis of these criteria can wash all work is divided into six categories of severity.

The first category of severity (light) - is the work performed under optimal environmental conditions and the optimum value of physical and mental stress. The reaction of the body indicates the normal functioning optimally, i.e. workload

corresponds to the physiological features of the body. Such conditions in healthy people contribute to the improvement of health, achieving high efficiency and productivity.

The second group includes *work of gravity*, in which the normal functional state is not reduced during the work shift. Maximum allowable concentrations and an additional level of harmful and dangerous production factors does not exceed the requirements of normative-technical documents. Efficiency is not broken, do not have occupational diseases.

The third category includes *the severity of operation* in which the result of increased stress or poor working conditions formed reaction of the body to the point of pathological, ie, reaction inherent premorbid state of the organism. Rest or better working conditions can quickly eliminate the negative effects.

When *the fourth category of gravity works* in healthy people formed deeper borderline states, physical indicators deteriorate arise due to production status before the disease.

The fifth category included the work in which due to the poor working conditions in healthy people at the end of the period of reaction formed, characteristic of pathological functional state of the organism. After a good rest, these states may disappear. However, some people may go into production due to occupational disease.

The sixth category is characterized by *the severity of the particularly unfavorable conditions*. Pathological reactions occur rapidly and are irreversible, accompanied by serious violations of the vital organs.

Hygiene factors - is, roughly speaking, the foreign work environment, namely climate (temperature, relative humidity, air velocity), the purity of air quality (presence of vapors, gases, aerosols), lighting, noise, vibration, ultrasound, various radiation, and other biological effects. Almost all of them are normalized by setting standards, sanitary standards and requirements and quantitatively evaluated using the methods of sanitary research.

Numerous studies hygienists and physiologists work found that the human body have a significant impact hygienic working environment factors. Some of them have an adverse effect on the employee, which reduces efficiency, impairs health and sometimes leads to occupational diseases. Therefore it is necessary to know not only the cause of these factors, but also to have an idea about how to reduce their negative impact on the body work. Particular attention should be paid to the influence of adaptive environmental factors (weather conditions, noise, vibration, light), the negative impact of which can be greatly reduced through the use of active means of improving the work process.

Aesthetic - these are the **factors**, the impact of which can cause a person corresponding to the conditions of labor in terms of artistic perception of the environment (meaning the use of color, form, music in the work of man). These elements are used in decision-art design quality of jobs, tools, work clothes, aids, as well as in architectural and interior design. An important element in modern conditions are functional music and painting production facilities. Their use is a positive impact on the work of artist and contributes to its performance.

Quantitative estimates of the elements of this group do not have. Determination aesthetic level of working conditions by using peer review methods.

Socio-psychological: Socio-psychological group of factors determined by the composition and characteristics of the company (the socio-demographic composition of the staff, the interests of employees, management style in the company offices, etc..). Under the influence of these factors formed the moral and psychological climate in the enterprise, manifested in the level of staff stability, its cohesion, the nature of relationships between groups of workers, the mood, the discipline of labor, labor activity and creative initiative. Social and psychological factors do not yet have the units, norms and standards. But the case studies in the form of oral questioning, questioning creating an objective basis for the creation of their measurement and evaluation.

The main directions of assessment of working conditions: Each of the factors considered, especially sanitation and psycho-physiological, have an impact on the health and performance of the worker. If some of them act simultaneously, they reinforce the overall impact on the person. Medico-physiological studies established the maximum allowable values for each of the factors under which they do not adversely affect the health worker. Knowing these limits and objectively determine the actual value of specific values at the workplace factors that can be targeted to carry out actions to improve working conditions.

Since the manufacturing conditions of work are considered from the point of view of their impact on the body running, an assessment of their actual state should be based on the effects of such an impact on human health. It is very important to take into account the diversity of factors that shape working conditions.

The main directions of assessment of working conditions: Above the classification of work has been presented in severity. To assess the state of psycho-physiological functions working techniques are used, aimed at measuring muscle strength and endurance, speed receipt and processing of information, the response time to visual stimulus.

On the performance is greatly influenced by such factors as the vibration of the working environment. When studying the effect of vibration on the human body should first consider the oscillation frequency. The typical relationship between the oscillation frequency of the vibration and the occurrence of diseases listed in Annex 2.

From the application data 2 that the development of vibration disease occurs to a lower limit of the oscillation frequency of 35 Hz and the upper - 25 Hz. When vibration disease decreases the sharpness of touch, tactile, thermal and pain sensitivity. The most pronounced symptoms of the disease - a negative change of the blood vessels.

Evaluating a hygienic factor as workstation lighting, are taken into account the following features. When you create optimal conditions for the activity of the visual analyzer, please note that effectiveness depends not only on the lighting but also on such important functions of the eye, as a reference sensitivity, visual acuity, speed and stability of the parts distinguishing clear vision. It is known that the maximum contrast sensitivity (the ability to distinguish between the brightness of the eyes of the painted surface) provided the background brightness in the range of 100 - 3,200 cd / m². Outside these values, it goes down. In contrast sensitivity affect size inspects surfaces.

In a production environment it is very important to distinguish the details as soon as possible. The speed of visual perception increases as the lighting. It increases rapidly when illuminated in Luke 70-75. Slower - while covering 1000-1200 to Luke and more. A clear perception of the subject eye retains a certain time. Calm eyes to keep a clear image of the part in question, i.e., a clear vision of sustainability, expressed in terms of time of clear vision to the total time the part in question. Sustainability clear vision depends on the nature of the work performed, the conditions of the working environment, in particular on the duration of illumination (with improved lighting dramatically increases the length of clear vision).

At the same voltage resistance of a clear vision with less intense visual work will be higher than the more strenuous.

4.3. Electromagnetic fields and radiation

Negative factors in the system "man - habitat" are divided into physical, chemical, biological, psycho-physical. When exposed to the human body, any of them can cause injury or occupational disease. Each negative factor characterized probability level and the exposure area as a person, and the components of its environment.

Electromagnetic fields and radiation is a major factor in the production sector, the negative impact on the human body - is ionizing radiation, and infrared (SRI), ultraviolet (UVB) and laser radiation;

Electromagnetic fields (EMF) and radiation (EMR) are detected at any point in space and production facilities throughout the environment of the biosphere. An electromagnetic field is created by an alternating electric current. Permanent electrical current creates a constant electrostatic field.

The sources (natural and artificial) are: - Powerful radio station; industrial electrical equipment; research facilities; control and measuring devices; line of electromagnetic transmission; atmospheric electricity; radio emission from the sun and galaxies.

Electromagnetic radiation permeated the surrounding area. The man is a source of low intensity EMR. In nature there are natural sources of electromagnetic radiation.

The natural sources of EMF include atmospheric electricity, solar radio emission and galaxies (CMB evenly spread in the universe); electric and magnetic fields of the Earth (storm - low emission EMI).

The harmful effects of EMR on man increased in the second half of XX century. due to the increased role of man-made sources of EMI: 1) production - a device for the induction of various materials (furnace, foundry aggregate); sources to ionize gas, producing a plasma; Device for welding plastics and compression; power lines, especially high; Switchgear; measuring devices, etc .; 2) in the home - electrical wiring; 3) the radio stations, television stations, power transmitters and antenna system etc.

The impact of electromagnetic fields on the human body is strictly individual, and it did not affect all of the energy of EMF, but only that its part of which is absorbed by the body. Reflected or transmitted through the body of the energy EMF does not have any effect on him. The negative impact on the person manifests in the appearance of the feeling of heaviness in the temples and back of the head, headache; in the weakening of memory; in sleep disturbances, immunobiological reactivity of the organism, the endocrine system, central and peripheral nervous system; in violation of the cardiovascular system; changes in metabolism in brain tissues of the blood; in ocular (lens opacities, cataracts); in violation of metabolic processes.

To protect workers from the adverse effects of EMF, the following measures:

- Hygienic regulation, the definition is extremely remote intensity EMF in the workplace;

- Preventing the spread of people in areas of high intensity EMF (sanitary protection zones; devices of automation and remote control);
- The use of RMS: 1) a decrease in the power source - a decrease in the parameters of the radiation source (protected amount) - the main absorbers - graphite, rubber, etc .; 2) screening of the radiation source - usually used metal plates that provide a rapid decay of the field in the material. However, in many cases, cost-effective use of the screen instead of the metal wire mesh, foil and radio-absorbing materials, cell lattice;
- The use of personal protective equipment (screens, overalls, goggles);
- Carrying out the standard treatment and preventive measures.

Electromagnetic fields are used for cleaning semiconductor materials growing semiconductor crystals and films, containment gases extrusion of synthetic materials.

Biological effects: The main danger - EMF exposure can not be detected by the senses. Under the influence of the EMF energy is absorbed by body tissues of human. As a result, in the body of standing waves, which concentrates heat. This increases the temperature of the human body, there is local heating of tissues and individual cells.

Particularly dangerous for the heating of a weak thermal insulation (brain, eye, lens, organs of the intestinal tract).

EMF changes the orientation of the cells, reduces the activity of the molecules, causing cataract, skin disease "string of pearls".

EMF is functionally pathological nervous and cardiovascular systems: increased fatigue, sleep disturbance, hypertension, neuro-psychiatric disorders.

EMF Protection:

- Protection of the amount - a reduction in the emission source.
- Protection of the time - reducing staff time to acceptable values.
- Protection of the distance - increasing the distance m / s from springs and jobs.
- Screening jobs or sources.
- Protection is carried out by remote control, process automation, alarm systems, fencing areas.
- Apply personal protective equipment: gowns, and others. Radio-protective overalls in performance; glasses with metallic glasses, which absorb electromagnetic radiation.

Control: Used equipment PZ-9; PZ-10 for 300 MHz-300 GHz. The measurement is made in the area of finding personnel at 2 m 3 levels: 0.5, 1, 1.5 m. The whole room is partitioned into grid with a pitch of 1 m and the measurement occurs at the points of intersection with the radiation power max.

Accident prevention: Hazardous Area Equipment

Danger zone - a space in which the possible effects on operating dangerous or VPF.

The danger is localized around moving machine parts, cutting tools, gear and other gears, conveyors, hoisting machines and mechanisms.

The presence of the danger zone causes the possibility of injury e. current to thermal, electromagnetic, ionizing radiation, ultrasound.

Dimensions of the danger zone may be constant (between the belt and pulley) and variables. In the design of the equipment is provided or no human contact with OZ or availability of remedies.

Remedies:

- Collective - signaling
- Remote control systems
- Individual
- On the principle of action
- Fencing
- Blocking

Blocking (blocking) - exclude the possibility of man's entry into the danger zone, or eliminate PF to stay human in the area. This type of protection is used where the work can be carried out to remove or open the guard. By the principle of action of the locking devices are divided into mechanical, electrical, photovoltaic, radiation, hydraulic, pneumatic.

Mechanical lock - a system that provides the connection between the fences and the brake (launchers) devices. When the guard removed the equipment can not be started in the job. So dangerous are blocked entrances to the premises where people stay forbidden.

Electric lock - used in electrical equipment with a voltage of 500 volts or higher. Provides switch equipment only with fences on the principle of limit switches.

Photovoltaic blocking - based on the principle fence the danger zone light beams.

Radiation blocking - based sensors and radiation detectors.

Protective device: Prevents the appearance of a person in the danger zone.

They are used for insulation systems, drive cars, processing zones, fencing systems and the live bands of radiation barrier at the height of the working area.

Constructive solution depends on the type of fencing equipment.

Safety devices: To automatically disconnect units and machines at the exit of any equipment parameters of range, which excludes emergency mode. For

installations under pressure - a safety valve and membrane uzlyl; thermal relays, water constipation - to prevent explosions compressors; travel stops, weight; braking systems; weak links (shear tabs; clutches that do not transmit motion at high torque).

Signaling device: Provide information about the work of the process equipment, as well as about the dangers and harmful factors, which in this case arise. By appointment divided into three groups:

- Operational security
- warning
- Identification

By way of transmission:

- Sound (sirens)
- Combined
- Visual (by smell)
- Odarizatsionnaya (smell)

For visual use artificial light sources: placards, color painting, check boxes (manual).

RAM is used during the tests on the stands automatically.

Warning - signs, posters, a system of signs (prohibiting warning - yellow).

Remote Control Systems: Characterized in that the control and management of the equipment is carried out from the areas remote from the danger zone.

Monitoring is performed either visually or by telemetry.

The parameters of the equipment received from sensors on the center console. DU is used in workshops with explosive and flammable materials, toxic substances

Conclusion

This is it, my commencement into the world as a budding game developer. Start small and build on what you've learned so far. Don't discourage myself by starting with that dream project that would take a dozens of developers, many years, and millions of dollars to produce. Whatever our goal in game development, enjoy every step of the journey and each small accomplishment as a victory.

As a result of graduation qualifying thesis i've created developed:

- Stripes for game;
- Wrote lot of scripts;
- Work in Monodevelop and fix errors;
- Finally build application and test it.

Engineering knowledge can take you to the road, but it's up to you to cross it and the path may not always be as you expect it to be. Find what works for you and try to enjoy the journey of making games.

Development games in unity3d it is very interesting and amusing, the mechanic and structuring very at the high level, it is possible to create big projects which as a result will bring huge profit and benefit.

References

1. Jeff W. Murray C# Game Programming Cookbook for Unity 3D CRC PRESS
2. “Learn Unity3D Programming with UnityScript. Unity’s JavaScript for Beginners.
3. Janine Suvak - Learn Unity 3D Programming with UnityScript. Unity’s JavaScript for Beginners - 2014
4. Sue Blackman - Beginning 3D Game Development with Unity 4 All-in-One, Multi-Platform Game Development 2nd Edition – 2013
5. Watkins A. - Creating Games with Unity and Maya How to Develop Fun and Marketable 3D Games - 2011
6. <http://developer.android.com/sdk/index.html>
7. <https://unity3d.com/unity/system-requirements>
8. <https://support.unity3d.com/hc/en-us/categories/200434305-Graphics>
9. <http://docs.unity3d.com/Manual/Sprites.html>
10. https://www.cla.purdue.edu/vpa/etb/resources/ad41700_unity3d_workshop_01.pdf
11. <http://gamedev.stackexchange.com/>

Application Scripting *Figure S1.*

```
using UnityEngine;
using UnityEngine.SocialPlatforms;

public class SocialExample : MonoBehaviour {

    void Start () {
        // Authenticate and register a ProcessAuthentication callback
        // This call needs to be made before we can proceed to other calls in the Social API
        Social.localUser.Authenticate (ProcessAuthentication);
    }

    // This function gets called when Authenticate completes
    // Note that if the operation is successful, Social.localUser will contain data from the server.
    void ProcessAuthentication (bool success) {
        if (success) {
            Debug.Log ("Authenticated, checking achievements");

            // Request loaded achievements, and register a callback for processing them
            Social.LoadAchievements (ProcessLoadedAchievements);
        }
        else
            Debug.Log ("Failed to authenticate");
    }

    // This function gets called when the LoadAchievement call completes
    void ProcessLoadedAchievements (IAchievement[] achievements) {
        if (achievements.Length == 0)
            Debug.Log ("Error: no achievements found");
        else
            Debug.Log ("Got " + achievements.Length + " achievements");

        // You can also call into the functions like this
        Social.ReportProgress ("Achievement01", 100.0, result => {
            if (result)
                Debug.Log ("Successfully reported achievement progress");
            else
                Debug.Log ("Failed to report achievement");
        });
    }
}
```

Figure S2

```
“Play”
using UnityEngine;
using System.Collections;

public class Play : MonoBehaviour
{

    void Start()
    {

    }

}
```

```

// Use this for initialization
public void play(){
    Application.LoadLevel ("Game");

}
public void Exit(){

    Application.Quit ();
}
}

```

Figure S3

“menu”

```

using UnityEngine;
using System.Collections;

public class Menu : MonoBehaviour {

    // Use this for initialization
public void Start () {

}

    // Update is called once per frame
public void Click () {
    Application.LoadLevel ("Game 1");
}
}

```

Figure S4

“Exit”

```

using UnityEngine;
using System.Collections;

public class Exit : MonoBehaviour {

    // Use this for initialization
void Start () {

}

    // Update is called once per frame
void Update () {
    if (Input.GetKey (KeyCode.Escape)) {
        Application.Quit ();
    }

}

}

```

Figure S5

“bonus”

```

using UnityEngine;
using System.Collections;

public class Bonus : MonoBehaviour {
    public bool first;
}

```

```

public bool second;
public bool therd;
public GameObject player;
public GameObject playerdown;
public Vector2 velocity = new Vector2(2, 0);
// Use this for initialization
void Start () {
    player = GameObject.Find ("Player(Clone)");

}

// Update is called once per frame
void Update () {

    GetComponent<Rigidbody2D>().velocity = velocity;
    transform.position = new Vector3(transform.position.x, transform.position.y,
transform.position.z);
    Destroy (gameObject,8f);
}

void OnTriggerEnter2D(Collider2D collider){

    Destroy (this.gameObject);//free up some memory
    if (first) {
        iTween.MoveBy (player, iTween.Hash ("x", 2, "easeType", "easeInOutExpo",
"delay", .1));
    }
    if (second) {

        iTween.MoveBy (player, iTween.Hash ("y", -7, 0, "easeInOutExpo", "delay",
.0));

        player.GetComponent<Rigidbody2D>().gravityScale = -3;
        player.GetComponent<Player>().playerUpwardsForce = -500;
    }
    if (therd) {
        iTween.MoveBy (player, iTween.Hash ("y", 7, 0, "easeInOutExpo", "delay", .0));
        player.GetComponent<Rigidbody2D>().gravityScale = 3;
        player.GetComponent<Player>().playerUpwardsForce = 500;
    }

}
}

```

Figure S6

```

“Random background”
using UnityEngine;
using System.Collections;

public class RandomBG : MonoBehaviour {

    public GameObject[] texture;
    public Transform[] spawnPoint;
    // Use this for initialization
    void Start () {

```

```

        //Camera.main.backgroundColor = colors[ UnityEngine.Random.Range(0,colors.Length-
1)];
        int randsprite = Random.Range(0, spawnPoint.Length);
        int spawnEnemyIndex = Random.Range(0, texture.Length);
        Instantiate(texture[spawnEnemyIndex], spawnPoint[randsprite].position,
spawnPoint[randsprite].rotation);

    }

    // Update is called once per frame
    void Update () {

    }

}

```

Figure S7
“Restart”

```

using UnityEngine;
using System.Collections;
using UnityEngine.UI;

public class Restart : MonoBehaviour {

    // Use this for initialization
    void Start () {
        //iTween.MoveBy (gameObject, iTween.Hash ("x", 15, "easeType", "easeInOutExpo",
"delay", .1));
    }

    // Update is called once per frame
public void Click () {
        Application.LoadLevel ("Game 1");
        ScoreManager.score = 0;
    }
}
“Score manager”
using UnityEngine;
using System.Collections;
using UnityEngine.UI;

public class ScoreManager : MonoBehaviour
{
    public static int score = 0;
    Text text;

    // Use this for initialization
    void Awake()
    {
        text = GetComponent<Text>();
    }

    // Update is called once per frame
    void Update()
    {
        text.text = "Score: " + score;
    }
}

```

Figure S8

“Spawn after GG”

```
using UnityEngine;
using System.Collections;

public class spawn : MonoBehaviour
{

    public GameObject[] enemy;
    public float spawnTime;
    public Transform[] spawnPoint;

    void Start()
    {

        InvokeRepeating("Spawn", spawnTime, spawnTime);

    }

    void Spawn()
    {

        int spawnPointIndex = Random.Range(0, spawnPoint.Length);
        int spawnEnemyIndex = Random.Range(0, enemy.Length);
        // float spawnTimeIndex = Random.Range(minRespawn, maxRespawn);
        Instantiate(enemy[spawnEnemyIndex], spawnPoint[spawnPointIndex].position,
spawnPoint[spawnPointIndex].rotation);

    }

}
```