

Unity Game Engine



Ryan Bax & Laura de Kort & Jordan Parezys



Introduction

We started designing and developing the game using the Unity Game Engine (v2018.2). We used a student personal license so we did not have to buy the pro engine. The game will not be published commercial so we don't have to take copyright into account. The engine works together with Visual Studio. Every line of code we wrote is in C#.



Learning the game engine

We started the first week with viewing interactive tutorials. You can find them on the Unity learn website (URL's at the end of the document). Almost 4 full days we have been viewing tutorials and experimenting with the engine. Then we started searching for good assets to put in our city. We had to take into account that we did not have any money resources available so we needed to search for free assets. We found some cool and interesting assets and placed them in our world. The week after we almost had a full build world ready.

Then we started learning how gameobjects and scripts can be linked together so actually everything can be programmed. After we knew how that worked, the real work started, we started programming from the morning until the evening. In the meantime, we also made changes to our city and the user interface. At the beginning of the 3rd week, we knew well how the engine works and how everything is connected together. Then we started implementing lots of fun things like the fish aquarium.

Megan & logo

Laura started drawing our own designed Megan, an hour later we were all in love with our own new mascotte: Megan!! We send the drawing to the marketing division and they transformed the drawing into a transparent png picture. After Megan, Laura started drawing the MeCity Logo.

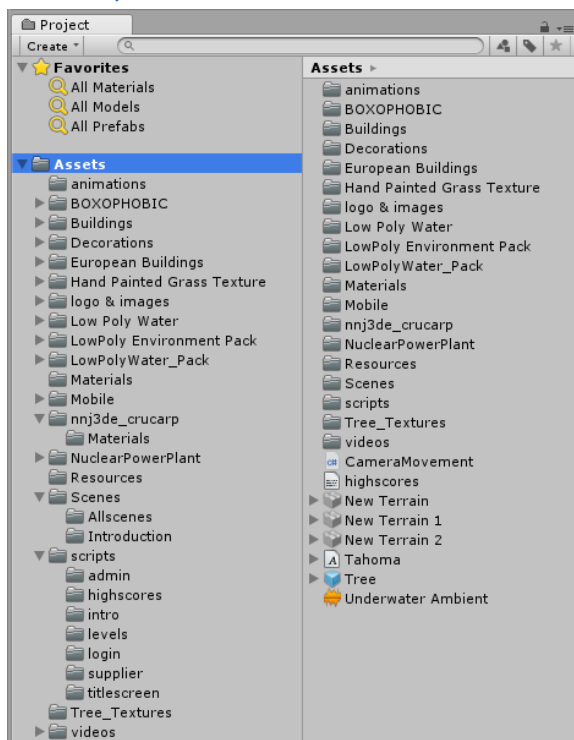


Installation

You can easily install the unity game engine and visual studio: <https://store.unity.com/>

We used the personal license. If visual studio isn't installed on the computer, it will be installed together with the engine.

MeCity files



In the assets folder you can find all the objects, files, images, animations... we used to make this game. We will try to explain every single folder/file.

Animations

In the animation folder, you can find all the animations that are in our game. The animations in the introduction level for the camera, making the buttons bigger when on-hover,...

Boxophobic

When you start the introduction or the supplier level (when you open the city) we used a skybox (background of the city). This is a sky texture with animated clouds (this animation is in this folder, not the animations folder)

Buildings

This speaks for itself, in this folder you can find all the building assets from the city.

Decorations

In this folder you can find the roads, trees, bushes,...

European Buildings

In this folder there are also building assets for the city.

Hand Painted Grass Texture

This folder is for the terrain color of the city.

Logo & images

In this map you can find all the images used in game (3d & 2d). Here you can also find audio files (created by Laura).

Low Poly Water

This folder includes the animated water you can see in MeCity.

(Low poly environment & Low poly water pack) are also folders from low poly water.

Materials

This is the material used for the turning cubes.

Mobile

This folder also includes building assets for MeCity.

Nnj3de_crucarp

This folder includes the animated fishes used for the high score screen.

Nuclear Power Plant

The Nuclear Power Plant for MeCity.

Resources

In this map the xml files are stored and is needed to load the xml files in the scripts. There is also a cruscab prefab in it, which is needed to load the fish model in the addfish script.

Cruscarp: fish prefab

Eventpopup: Event popup prefab for in the supplier level

IntroScriptsXML.xml: Texts for the introduction level

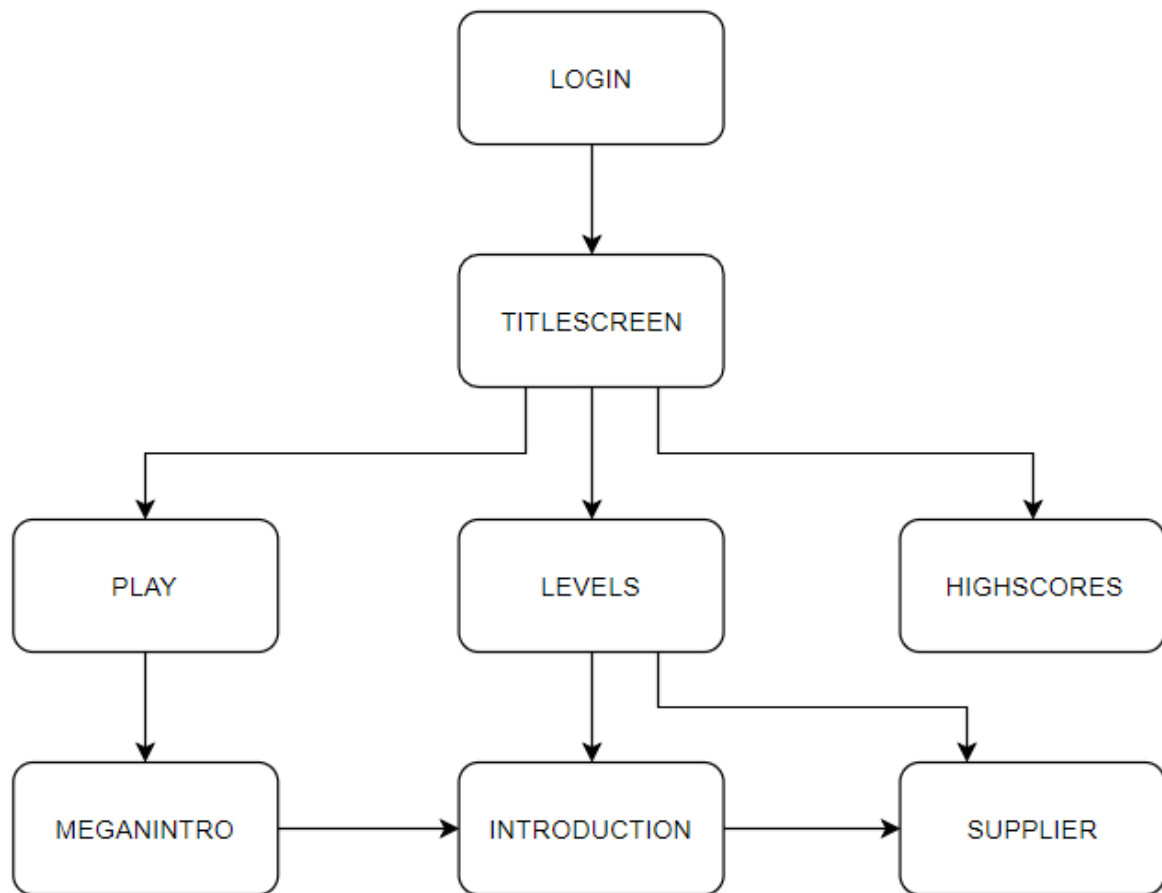
InvoicePopUp: invoice popup prefab for in the supplier level

ScriptsXML.xml: Texts and answer texts for the event system in the supplier level. Every answer also contains an influence; this number is used to add or subtract from the happiness slider

SupplierIntroXML.xml: Texts for the supplier level introduction/tutorial

Scenes

In this folder you can find all the scenes (screens). The folders allscenes and introduction are needed for occlusion cluttering (occlusion cluttering = the camera only loads the models that are in view of the camera).



- Login: the login scene is used to get input from the user to get the player names.
- Titlescreen: the main screen, used to navigate around.
- Meganintro: Megan giving her explanation with on the background a video.
- Introduction: the introduction level.
- Supplier: The supplier level.
- Levels: Making the user able to select which level he wants to play.
- Powerplant scene: just don't delete this, used for the city.
- Streets scene: just don't delete this, used for the city.
- Highscores: aquarium, where the fishes swim and the highscore list is located.

Scripts

In this map you will find all the scripts for the game.

Admin/decrypt: Used to decrypt the online highscores.txt file (script linked with button on titlescreen)

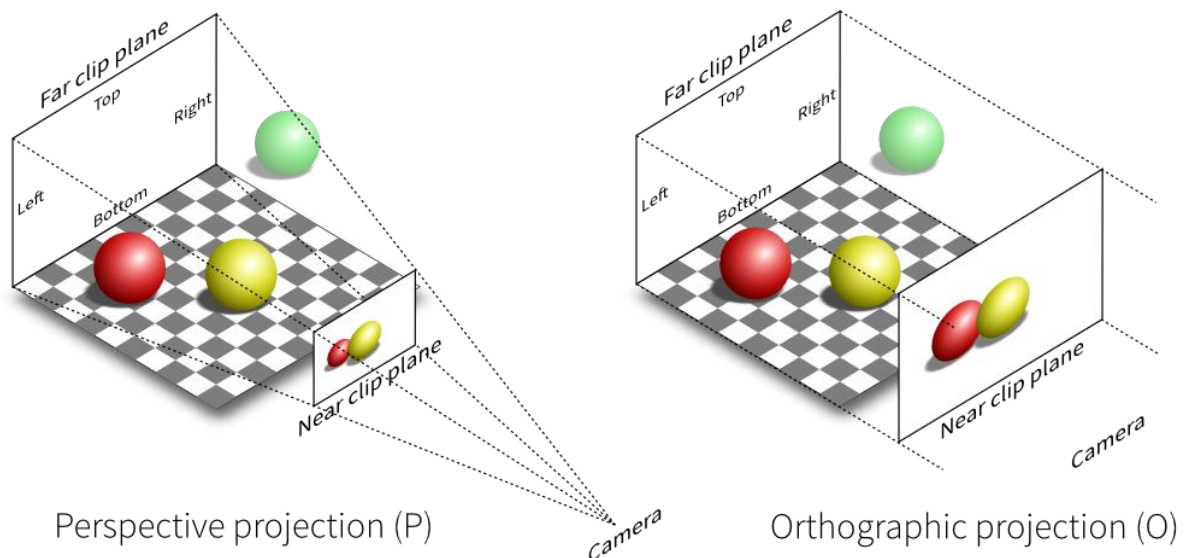
Admin/encrypt: used to encrypt the online highscores.txt file (script linked with button on titlescreen)

Highscores/addfish: Used to copy the online highscores file offline and to initiate the fishes in the fish aquarium

Highscores/btnCloseHighscores: Used to close the top highscores list screen.

Highscores/btnScoresOnClick: Used to open the top highscores list screen.

Highscores/Camerascript: The aquarium ingame camera is an orthographic camera, this script controls the way the camera zooms in- and out as the fishes move away from each other.



Highscores/MoveRandomly: Used to real life like move the fishes.

Intro/MeganIntro: Script used to load the second text in the megan intro screen (when you press play) when you press next, when you press next again the introduction level is loaded.

Introduction/Clickhouse: Script for enabling the Megan introduction canvas when clicking on the house.

(also all canvases are disabled when this object is loaded in the game)

Introduction/CloseCanvas: Script used to close the last producer canvas and to start the city overview animation.

Introduction/EnableClickhouse: script used to activate the startcanvas (panel on top of the screen: click on the house to begin).

Introduction/EnableNextLevel: Script used to enable the next level button after the overview animation is finished playing.

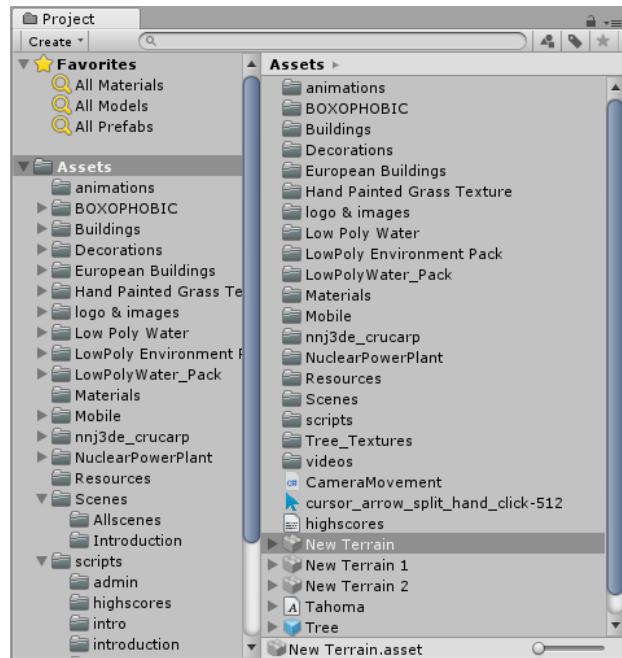
Introduction/LoadDgo: Script used to close the supplier canvas and to enable the dgo canvas, to load the text from the xml file and to play the animation.

Introduction/LoadProducer: Script used to disable the TgoCanvas and to enable the ProducerCanvas, to load the text from the xml file and to start the animation.

Introduction/LoadSupplier: Script to close the MeganCanvas and to open the SupplierCanvas, to load the text from the xml file and to start the animation.

Introduction/LoadTgo: Used to disable the DgoCanvas and to enable the TgoCanvas, to load the text from the xml file and to start the animation.

Levels/LevelHighscore: Script used to load the current player name and score in the bottom left corner in the level scene.



Login/Login: Script used to validate the user username input field and writes the player and the beginning score in the Datascript script.

Supplier/changeDGO: Script used for the market canvas. Randomly generating the energy market values and used to make the dropdown and buttons working.

Supplier/CheckEndOfGame: Script used for win and lose control.

Supplier/CloseWindow: Simple script used to close a canvas.

Supplier/CubeSatisfaction: Used for the cubes turning above the houses.

Supplier/EventSystem: Used to load the text, buttons and happiness influence in the events canvas (xml file).

Supplier/HomeCanvasScript: Used to randomly generate the values of the house canvas (clicking on a random house).

Supplier/Introduction: Script used to load the introduction/tutorial of the supplier level.

Supplier/LoadPause: Script used to pause the game.

Supplier/OpenWindow: Used to open a canvas.

Supplier/Statisfaction: general script used to do different things like: Energy increases/decreases, money increases/decreases, popup generating,...

Supplier/SupplierStart: Script used to disable all canvases at the start of the level.

Supplier/TarifScript: Script used for the tariff canvas.

AudioMute: Script used for muting and unmuting the audio (linked with button).

Cameracontrol: Used for the in game camera controls (and validation).

ClosePauseMenu: Script used for closing the pause menu en resuming the game.

Datascript: Script used to save the current players name and score.

Drive: Script used to let the cars drive around (they use waypoint gameobjects to move from point a to point b).

Encryption: Script used to encrypt and decrypt the offline highscores.txt file and to append the current player name and score to the online encrypted highscores.txt file

QuitGame: Script used to quit the game and load all the scenes when you press for example on the start, levels, highscores,... buttons.

Tree_textures

The textures for the trees.

Video's

The title screen and Megan intro videos are saved here. They play in a loop.

Other

Cursor_arrow_split_hand_click-512: the in-game cursor.

Tahoma: Mecoms lettertype used.

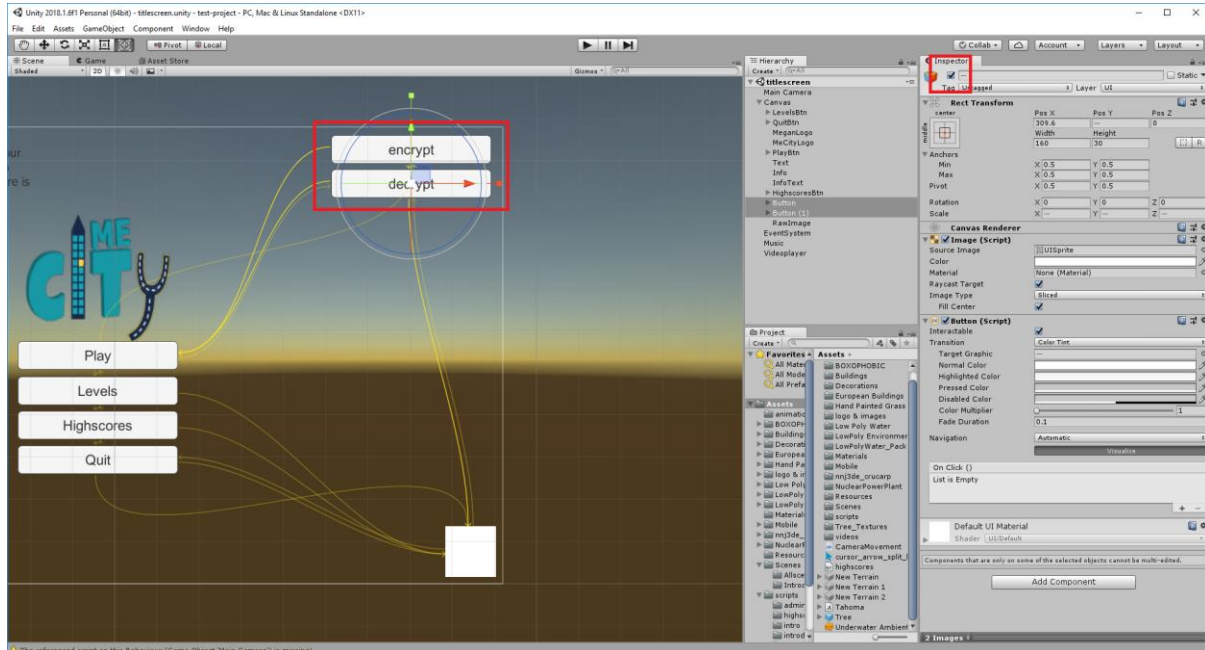
Underwater ambient: aquarium sound (replaced by Laura with fishes\$\$), still kept it in the assets folder.

Highscore

The way we save high scores is a little bit complicated. When we start the game we ask for a username. We save the name and a beginning score to the static Datascript. While playing the game the score will be changed by random events. When you click on the button highscores at the titlescreen, this happens: We have an online file highscores.txt, (now located at I:/Cbrasp/MeCity/highscores.txt, during the game-event this file was located on the transfer disc of ryabax, which includes every username with his score. This file is encrypted using UTF8(iv and key), CBC(mode) and PKCS7(padding). When the highscores scene gets loaded we make a copy of that file and copy it to Appdata/local/MecomsCity/MeCity/highscores.txt which will then be decrypted so we can use that file to load all the fishes and the highscore list. When it's game-over playing the supplier level and you press the quit button, your name and highscore will be saved and appended to the online encrypted highscores.txt file.

Encrypting and Decrypting the highscores file

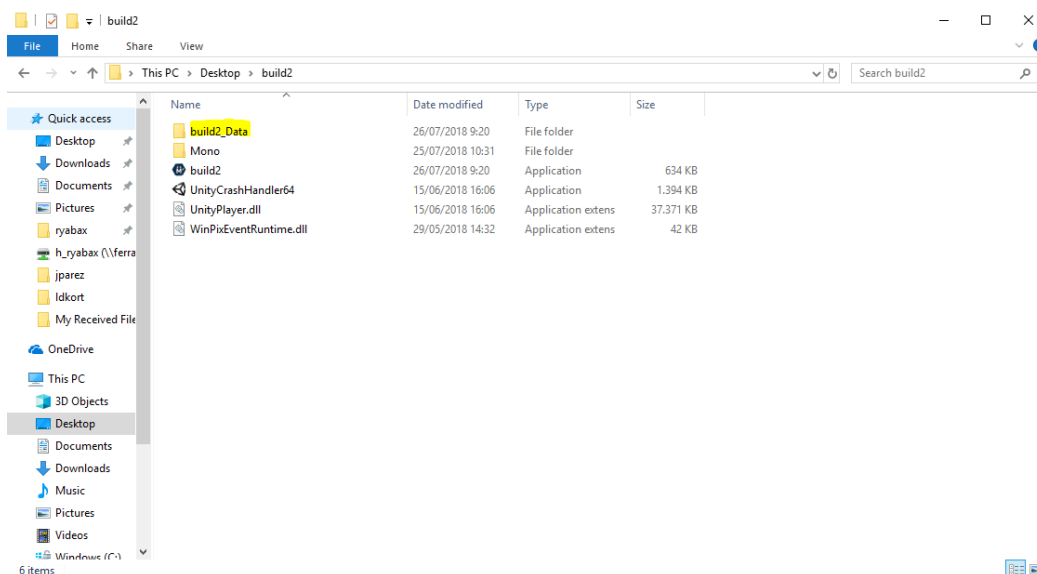
When you open the online highscores.txt file it will be encrypted. To decrypt the file; on the titlescreen are encryption and decryption buttons, these can easily be disabled in the game engine by pressing the buttons in the scene editor and unchecking the gameobject box in the inspector and saving the scene (ctrl+s). Be sure to always close the file before pressing encrypt or decrypt. Also don't press twice on any of these buttons otherwise the whole file will be corrupted. In the console at the bottom left corner there will be a sentence: "Encrypt/decrypt successful".



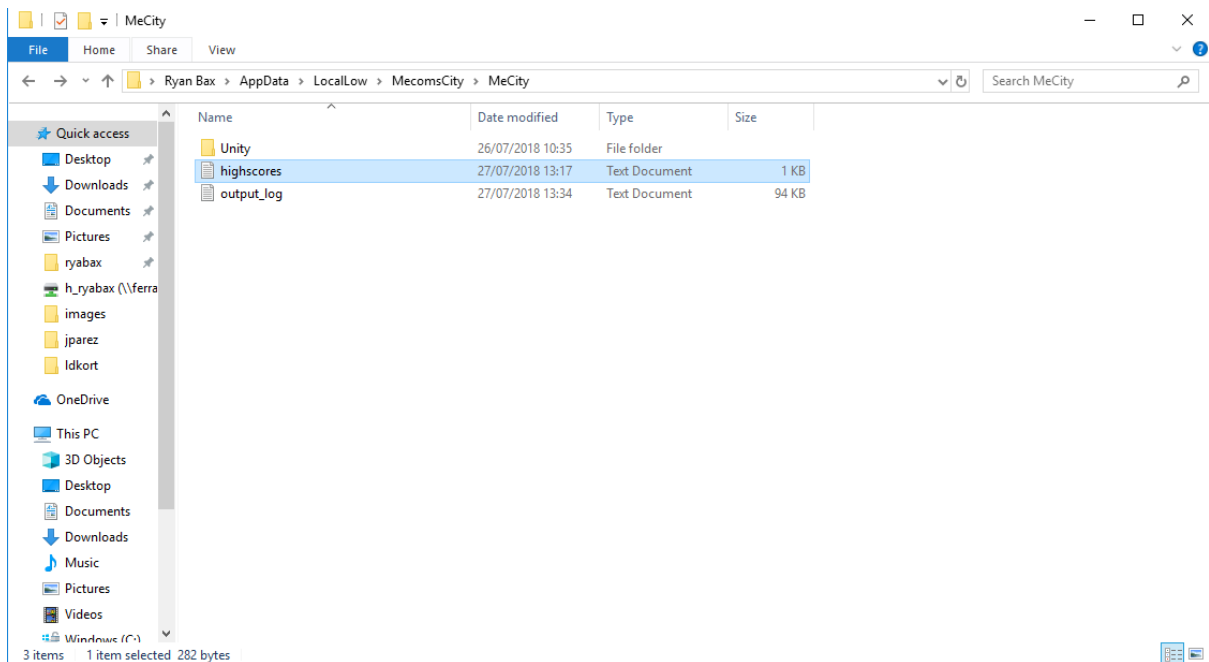
Folders

In the scripts you will find filePaths like:

Application.DataPath = C:/Users/[username, ryabax for example]/Documents/Unity/Github/test-project/Assets , When you play the game in the Unity Game Engine Editor this file will be saved in the Project/Assets folder like above. When you play the game with a build the file will be saved in the build/build_data folder.



Application.persistentDataPath = C:/Users/[username]/AppData/LocalLow/MecomCity/MeCity/ this will always be the same path.



Online highscores.txt file (encrypted) is saved on a server (Transfer disc for the game event).

Offline highscores.txt file (decrypted) is saved on the persistentDataPath.

The playing player name and score is saved on the static Datascript.

Tools we used during development

- Fraps <http://www.fraps.com/>
- Techpower GPU-Z <https://www.techpowerup.com/gpuz/>
- Github
- Umlet
- Sketchfab for unity (used for transferring Sketchfab assets to unity assets)

URL's

<https://assetstore.unity.com/search>

<https://unity3d.com/learn/tutorials>

<https://unity3d.com/learn/tutorials/s/tanks-tutorial>

<https://docs.unity3d.com/Manual/index.html>

<https://unity3d.com/learn/tutorials/topics/interface-essentials>

<https://unity3d.com/learn/tutorials/s/user-interface-ui>

<https://unity3d.com/learn/tutorials/s/graphics>

<https://docs.unity3d.com/Manual/OptimizingGraphicsPerformance.html>