Project assets

In your "Project" pane, you can find different types of assets:

# Prefabs

Reusable game objects (ex: Bullets, enemies, bonuses).

Prefabs can be seen as a class in a programming language, which can be instantiated into game objects. It's a mold you can duplicate and change at will in the scene or during the game execution.

# Scenes

A scene is basically a level or a menu.

Contrary to the other objects you create in the "Project" pane, scenes are created with the "File" menu. If you want to create a scene, click on the "New Scene" submenu, then do not forget to save it to the "Scenes" folder.

**Scenes need to be manually saved.** It's a classic mistake in Unity to make some changes to a scene and its elements and to forget to save it after. Your version control tool will not see any change until you scene is saved.

# Sounds

I guess it's pretty clear. See if you want to split music in another folder.

# Scripts

All the code goes here. We use this folder as the equivalent of a root folder in a C# project.

# Textures

Textures are sprites and images of your game. In a 2D project, you could rename this folder to "Sprites".

It doesn't really matter in a 2D project, but by keeping the name "Textures", Unity will recognize it and automate some tasks. If you want some informations about this topic, you can read [this](http://answers.unity3d.com/questions/172384/importing-models.html) or[this](http://docs.unity3d.com/Documentation/Components/class-Mesh.html).

Scripting

Default scripts come with the **Start** and **Update** methods. Here is a short list of the most used "Message" functions:

Awake() is called once when the object is created. See it as replacement of a classic constructor method.

Start() is executed after Awake(). The difference is that theStart() method is not called if the script is not enabled (remember the checkbox on a component in the "Inspector").

Update() is executed for each frame in the main game loop.

FixedUpdate() is called at every fixed framerate frame. You should use this method over Update() when dealing with physics ("RigidBody" and forces).

Destroy() is invoked when the object is destroyed. It's your last chance to clean or execute some code.

You also have some functions for the collisions :

OnCollisionEnter2D(CollisionInfo2D info) is invoked when another collider is touching this object collider.

OnCollisionExit2D(CollisionInfo2D info) is invoked when another collider is not touching this object collider anymore.

OnTriggerEnter2D(Collider2D otherCollider) is invoked when another collider marked as a "Trigger" is touching this object collider.

OnTriggerExit2D(Collider2D otherCollider) is invoked when another collider marked as a "Trigger" is not touching this object collider anymore.