

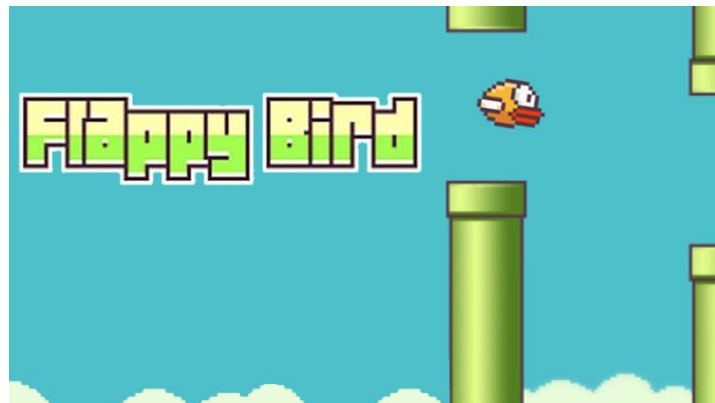


H1 - Workshop_Create_Game

H-HUB-100

Flappy Bird

Create your game in 2 hours !



1.0

Flappy Bird

binary name: Workshop_Create_Game
repository name: Workshop_Create_Game
repository rights: ramassage-tek
build tool: Unity



- The totality of your source files, except all useless files (binary, temp files, obj files,...), must be included in your delivery.
- All the bonus files (including a potential specific Makefile) should be in a directory named *bonus*.
- Error messages have to be written on the error output, and the program should then exit with the 84 error code (0 if there is no error).



Flappy Bird is an obstacle video game developed by Nguyen Ha Dong, a vietnamese independent game developer. His game was then published by GEARS Studios.

The goal of this workshop is to create a clone of Flappy Bird in several steps, each of these steps will have a timer to respect in order to stay on time and have an executable at the end of the session.

The steps are:

- The player
- Obstacles
- Collisions and interaction
- The menu
- The compilation
- Bonuses



For all your questions don't hesitate to look at the **Unity doc**, one of the best made docs in the world.

The estimated time for each step is:

- The player: 30min
- Obstacles: 30min
- Collisions and interaction: 20min
- The menu: 15min
- The compilation: 10min
- Bonuses: Time left...



THE PLAYER



To get started you need a good installation of unity

For this step you need:

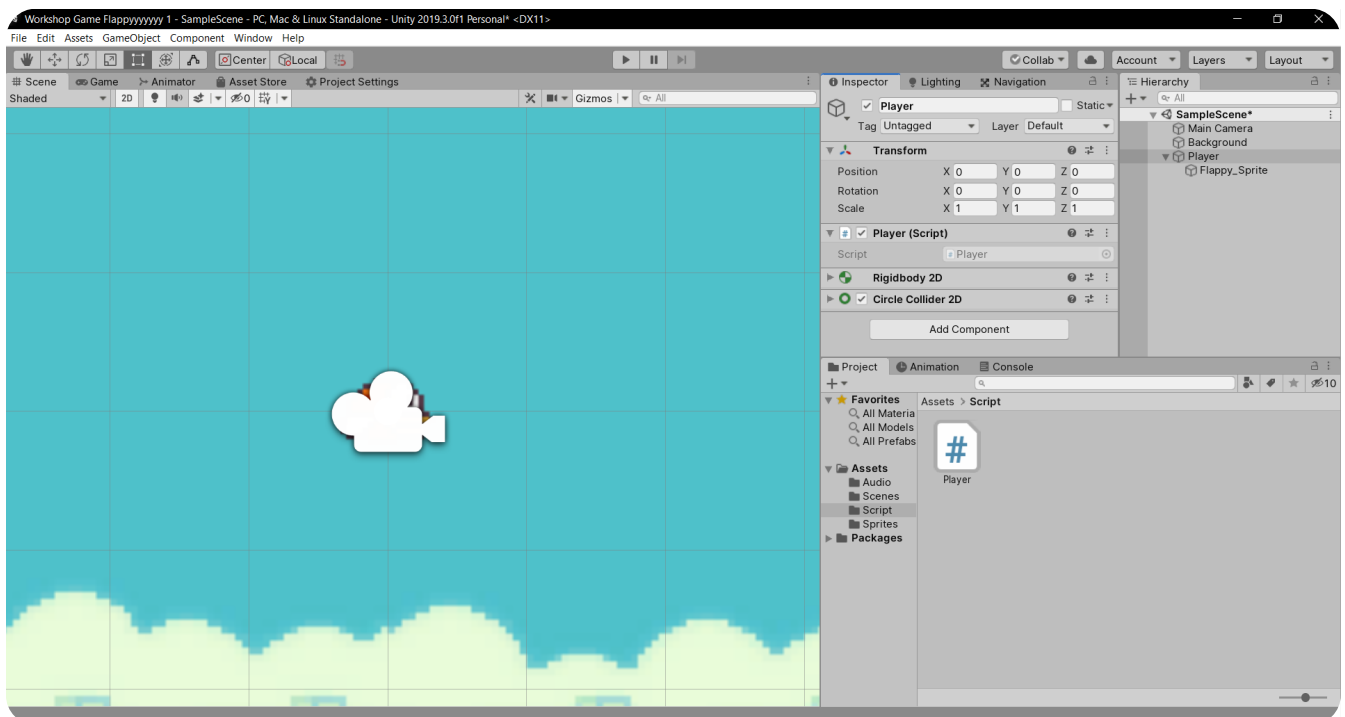
- * The component Vector2 and Rigidbody2D
- * Input.GetKeyUp



use a step-by-step process



use the **doc!**





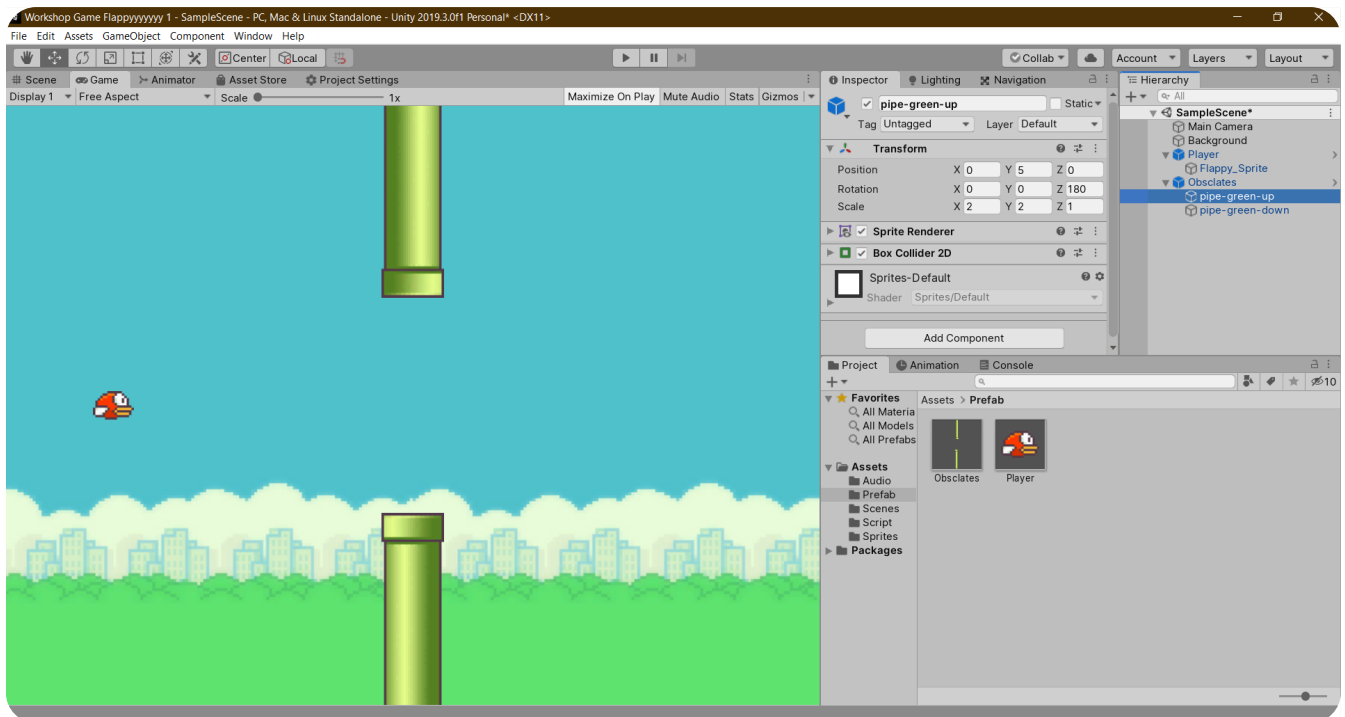
OBSTACLES



use a prefab

For this step you need:

- * `InvokeRepeating();`
- * `Instantiate();`
- * `transform.position`





COLLISIONS AND INTERACTION

For this step you need:

- * void OnTriggerEnter2D(Collider2D col)
- * using UnityEngine.UI;
- * Destroy() or scene management;

```
private void OnCollisionEnter2D(Collision2D collision)
{
    die();
}

void OnTriggerEnter2D(Collider2D col)
{
```



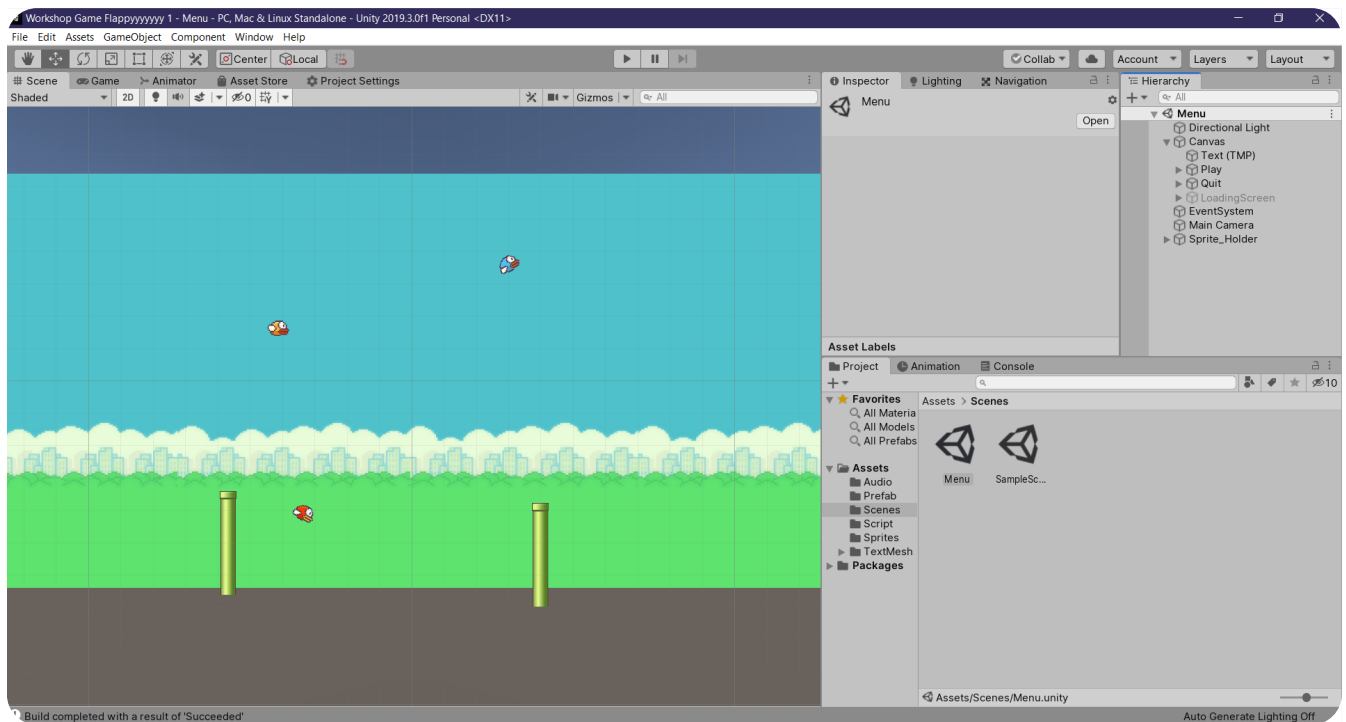
THE MENU

For this step you need:

- * using `UnityEngine.SceneManagement;`
- * `Application.Quit();`



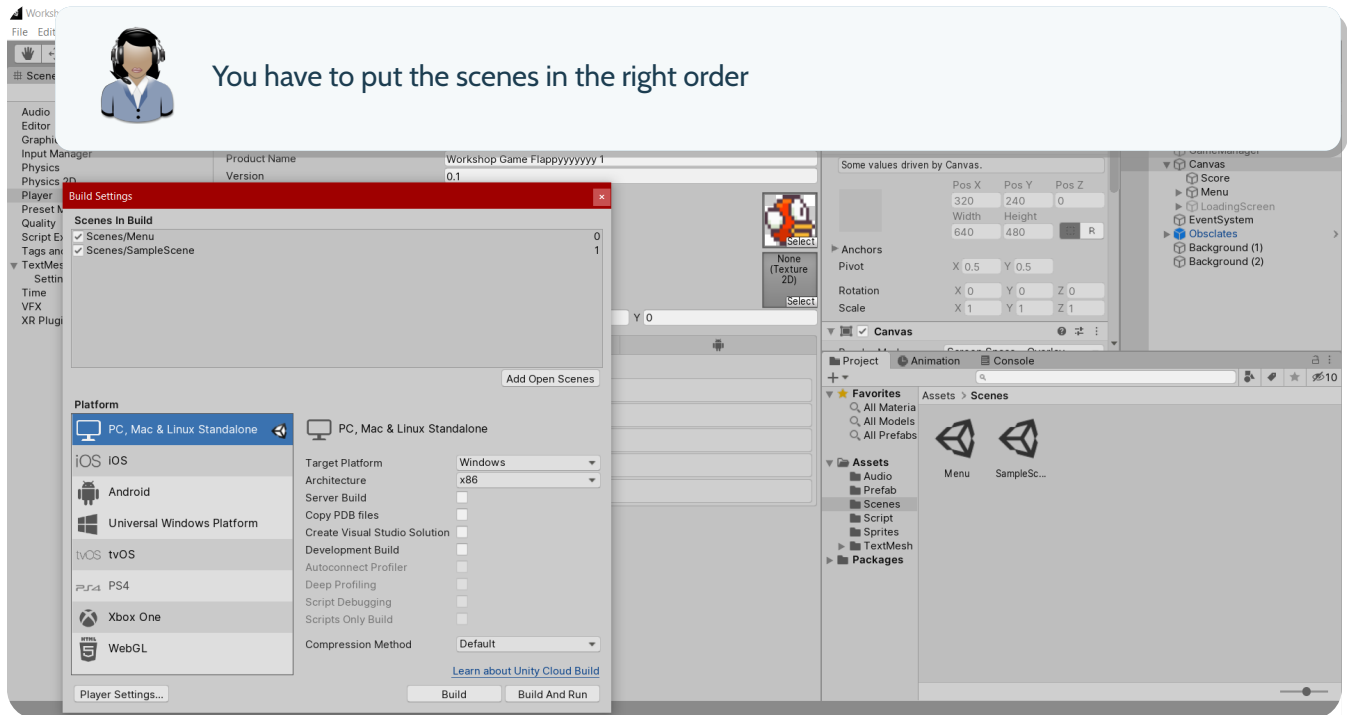
You need set your scene in management window



THE COMPILATION

For this step you need:

- * The build window





BONUSES

For this step you can do whatever you want with the time you have left

Examples:

- * Animation
- * Audio
- * Effect
- * Parameter
- * AsyncOperation
- * ...