



# Shape Race

**Thank you for purchasing the asset!**

The asset is mainly made for publishing and learning purposes, but you can use it as you like.

Please contact me if you have any questions/issues!

**If you like the asset, please leave a review!**

**If you don't like it, or you have any questions, don't hesitate to send me an email!**

You can use anything from the asset for any purpose.

**Email:** [ragendom@gmail.com](mailto:ragendom@gmail.com)

**Facebook:** <https://www.facebook.com/Ragendom-283436829168712/>

**Please use Unity 2019.1.1f1 to avoid errors**

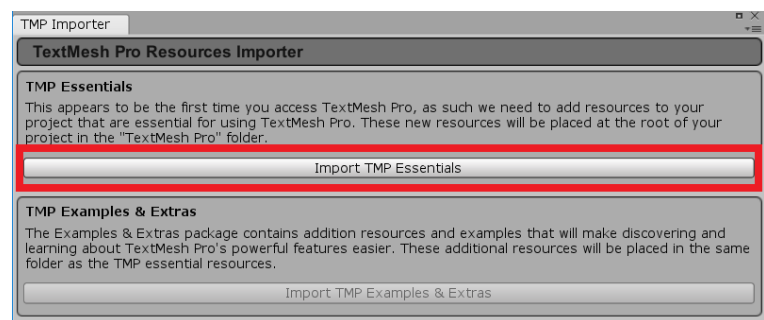
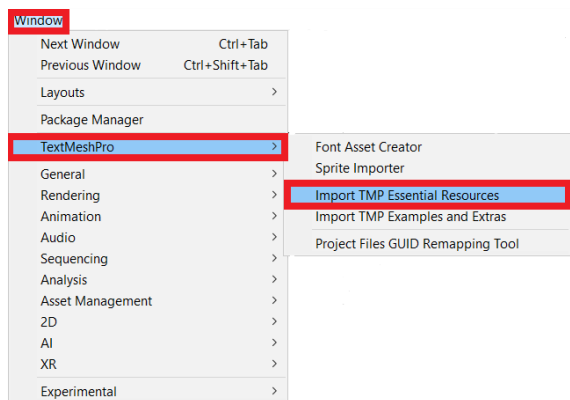
## Credits:

Background music by Eric Matyas: <http://www.soundimage.org>

Sound effects: <https://www.noiseforfun.com>

## IMPORTANT

1. In this asset I use **TextMesh Pro** for the text elements. First you have to import it.



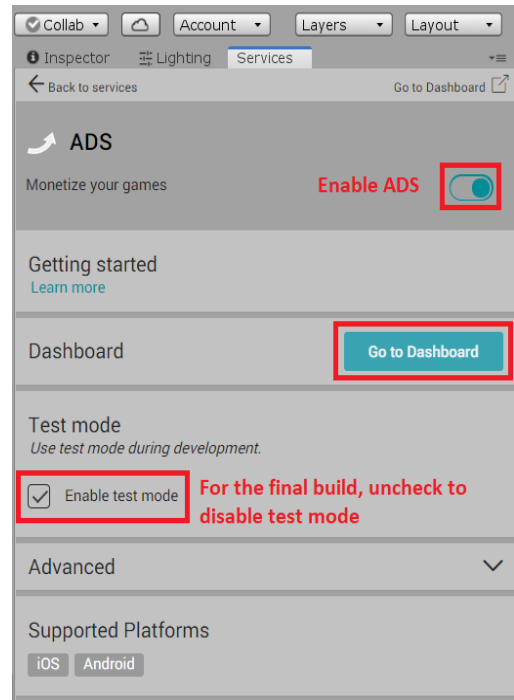
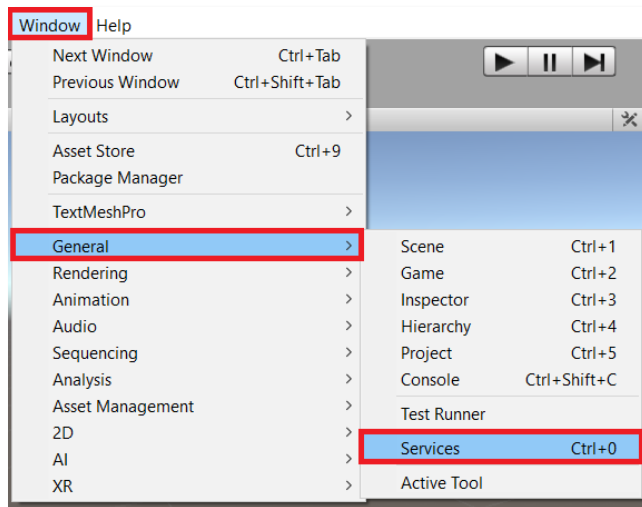
2. In this asset I use ***Bézier Path Creator*** for the path creation process. First you have to import it from the Asset Store.

***Here is the link for Bézier Path Creator:***

<https://assetstore.unity.com/packages/tools/utilities/b-zier-path-creator-136082>

# IMPLEMENTING ADS

## 1. Open *Services*



2. Enable Ads. If you want to test the ads, then enable test mode, but **you will need to disable test mode for the final build.**

## 3. Close and reopen project!!!

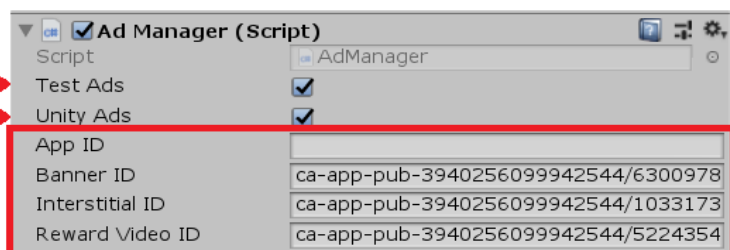
4. Find AdManager in the Hierarchy.

**You will need to disable test mode for the final build.** You can select the AD network here.

**You have to change App ID, and AD IDs for the final build.**

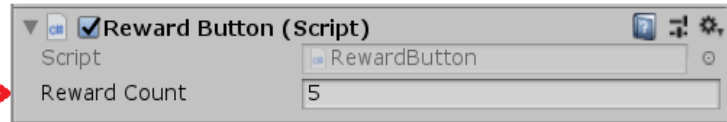
**Enable this for testing, disable it for final build**  
**Using Unity Ads when enabled, using Admob when disabled**

**Use the ad IDs you created in your Admob Panel**

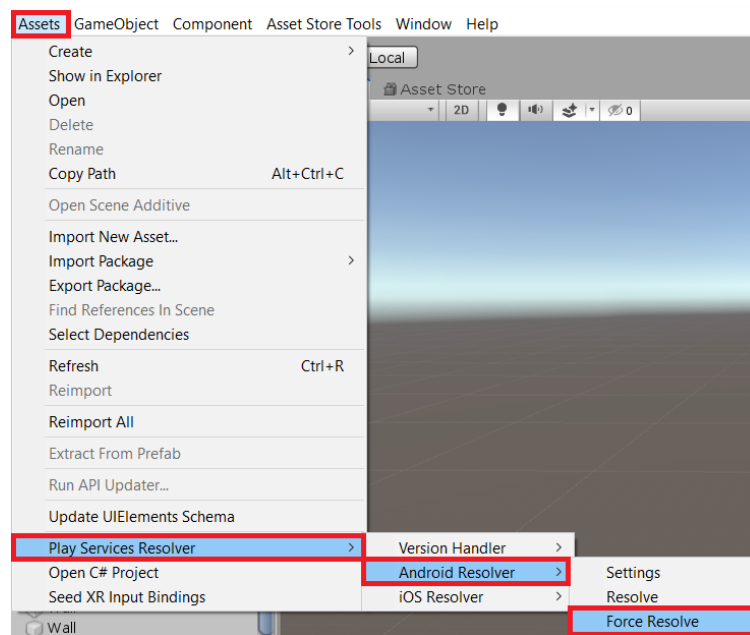


- Find RewardButton in the Hierarchy. You can change the amount of tokens the player gets after watching a Reward Video Ad.

Player gets this many tokens after watching a Reward Video Ad ➡



### 5. Resolve just before building the project.



## Calling ADS

### Admob

```
FindObjectOfType<AdManager>().ShowAdmobBanner();  
FindObjectOfType<AdManager>().ShowAdmobInterstitial();  
FindObjectOfType<AdManager>().ShowAdmobRewardVideo();
```

### Unity ADS

```
FindObjectOfType<AdManager>().ShowUnityVideoAd();  
FindObjectOfType<AdManager>().ShowUnityRewardVideoAd();
```

If you want to use Unity Ads, then after you have done the 1, 2, 3 steps, you have to **uncomment** some lines in the following scripts:

**GameManager.cs**      **Uncomment lines: 21-23, 32-37**

```
19 void Start () {
20     //UNCOMMENT THE FOLLOWING LINES IF YOU ENABLED UNITY ADS AT UNITY SERVICES AND REOPENED THE PROJECT!
21     //if (FindObjectOfType<AdManager>().unityAds)
22     //    CallUnityAds(); //Calls Unity Ads
23     //else
24     //    CallAdmobAds(); //Calls Admob Ads
25
26     StartPanelActivation();
27     HighScoreCheck();
28     AudioCheck();
29 }
30
31 //UNCOMMENT THE FOLLOWING LINES IF YOU ENABLED UNITY ADS AT UNITY SERVICES AND REOPENED THE PROJECT!
32 //public void CallUnityAds()
33 //{
34 //    if (Time.time != Time.timeSinceLevelLoad)
35 //        FindObjectOfType<AdManager>().ShowUnityVideoAd(); //Shows Interstitial Ad when game starts (except for the first time)
36 //    FindObjectOfType<AdManager>().HideAdmobBanner();
37 //}
38
```

**RewardButton.cs**      **Uncomment lines: 23-25**

```
20 public void ShowRewardVideo()
21 {
22     //UNCOMMENT THE FOLLOWING LINES IF YOU ENABLED UNITY ADS AT UNITY SERVICES AND REOPENED THE PROJECT!
23     //if (FindObjectOfType<AdManager>().unityAds)
24     //    FindObjectOfType<AdManager>().ShowUnityRewardVideoAd(); //Shows Unity Reward Video ad
25     //else
26     //    FindObjectOfType<AdManager>().ShowAdmobRewardVideo(); //Shows Admob Reward Video ad

```

**AdManager.cs**      **Uncomment lines: 253-259, 262-288**

```
251 /*
252 //Call this to show video ad
253 public void ShowUnityVideoAd()
254 {
255     Debug.Log("ShowUnityVideoAd");
256
257     if (Advertisement.IsReady("video"))
258         Advertisement.Show("video");
259 }
260
261 //Call this to show reward video ad
262 public void ShowUnityRewardVideoAd()
263 {
264     Debug.Log("ShowUnityRewardVideoAd");
265     if (Advertisement.IsReady("rewardedVideo"))
266     {
267         Debug.Log("Showing Advertisement");
268         var options = new ShowOptions { resultCallback = HandleShowResult };
269         Advertisement.Show("rewardedVideo", options);
270     }
271 }
272
273 private void HandleShowResult(ShowResult result)
274 {
275     switch (result)
276     {
277         case ShowResult.Finished:
278             Debug.Log("The Unity Reward ad was shown successfully");
279             FindObjectOfType<RewardButton>().AddReward(); //Adds reward to the player when rewardVideo is finished successfully
280             break;
281         case ShowResult.Skipped:
282             Debug.Log("Ad was skipped");
283             break;
284         case ShowResult.Failed:
285             Debug.LogError("The ad failed to be shown");
286             break;
287     }
288 }
289 */

```

## ABOUT THE GAME

### How to play

**On mobile :** Move your finger on the screen while holding it to make the player move.

**On desktop/web :** Use "A" and "D" or the arrow keys on your keyboard to move the player.

### Goal

Your goal is to hit the obstacles/enemies which have the same shape as the player.

However, if the player collides with an enemy or with an obstacle which has different shape, then you lose.

## FOLDER STRUCTURE

- **Assets**

Contains all of the used assets for the game

- ***Animations***

Contains all of the used animations

- ***Audio***

Contains all of the used sound effects and background music

- ***Materials***

Contains every used materials

- ***Meshes***

Contains every used meshes

- ***Particles***

Contains the used ParticleSystems

- ***Prefabs***

Contains the used prefabs (for example obstacles)

- ***Scripts***

Contains C# scripts

- ***Sprites***

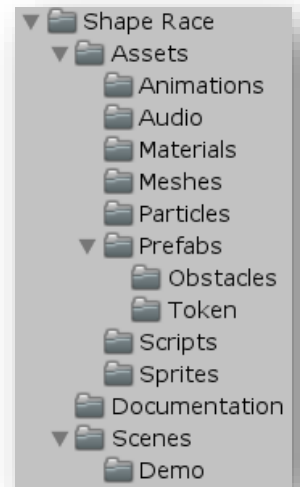
Contains all of the used 2D sprites (for example UI elements)

- **Documentation**

Contains documentation

- **Scenes**

Contains complete level(s) and lightmap of the level(s)

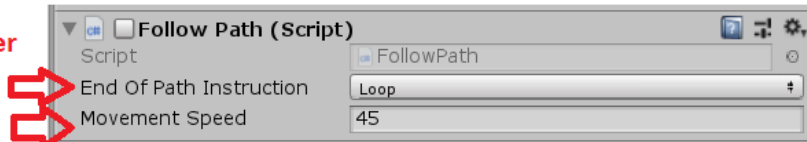


## CUSTOMIZATION

*PlayerHolder:*

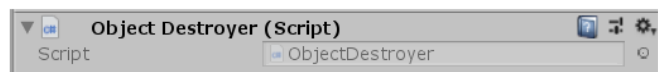
What happens when the player reaches the end of the path

Player's movement speed on the Z axis



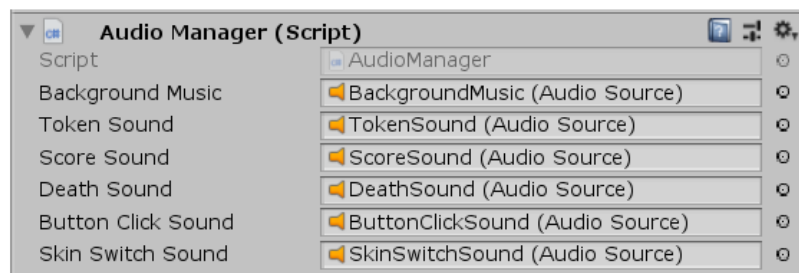
*ObjectDestroyer:*

ObjectDestroyer script is responsible for destroying every object that collides with it



*AudioManager:*

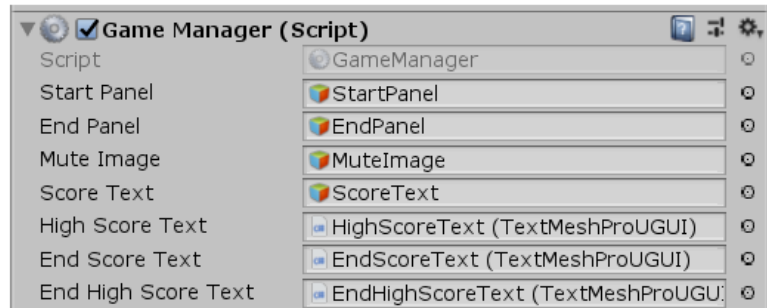
You can modify the sound effects here and under the AudioManager gameObject





## GameManager:

**GameManager script contains the main functions of the game. It manages scenes (starts, restarts the level), enables/disables scripts and gameObjects, switches between panels**



## Spawner:

**You can add new obstacles here** →

**Enemy Token** →

**The lower it is, the more chance to spawn a token** →

**What happens when spawner reaches the end of the path** →

**First enemy and obstacle spawn after x seconds** →

The screenshot shows the Unity Inspector for the 'Spawner (Script)' component. The 'Script' field is set to 'Spawner'. Below it, several fields are listed with their corresponding values: 'Obstacles' (EnemyHolder), 'Token' (Token), 'Token Spawn Frequency' (8), 'Time Between Obstacle Spawns' (1), 'Min Time Between Spawns' (1), 'Offset Y' (1), 'End Of Path Instruction' (Loop), 'Time Between Enemies' (3.2), 'First Enemy Spawn' (3), 'First Obstacle Spawn' (2), and 'Ahead Of Player' (100). Red arrows point to specific fields with annotations: 'Obstacle spawns again after this time' points to 'Time Between Obstacle Spawns', 'Enemies spawn again after this much time' points to 'Time Between Enemies', and 'This is how far the spawner is from the player' points to 'Ahead Of Player'.

**Obstacle spawns again after this time**

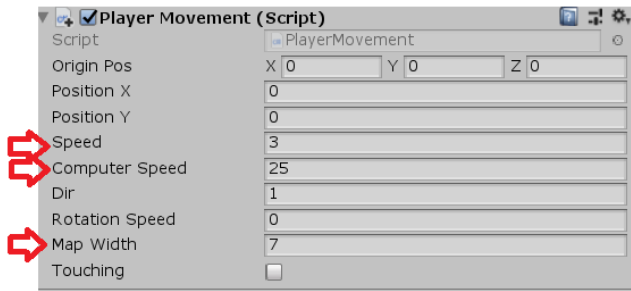
**Enemies spawn again after this much time**

**This is how far the spawner is from the player**

*Player:*

Speed is the speed of the player in the X axis while you use phone, computer speed is the same while you use computer

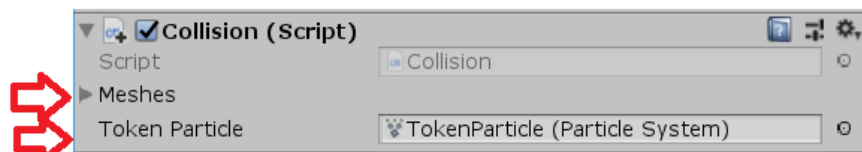
Player can move on the X axis between -7 and 7 in this case



*Player:*

You can add new meshes for the player

You can change the token's particle here



*Best regards,Ragendom*

*v 1.0*