
1 Solves the multiple deaths when hits spikes objects problem

An issue with the previous method of dying when you hit 'spikes' is that you may hit multiple spike objects, and therefore lose more than one life, before the character is repositioned back to its start position.

The solution – only have a single gameObject tagged 'Spikes' (containing lots of untagged images):

- So it looks like lots of spikes, but there will only ever be one collision

1.1 Follow these steps

Preparation:

- remove all spikes / spikes group gameObjects from the scene
- delete your Spikes prefab (so all you have is the spikes image in the Sprites folder)

Setup:

- position lots of Spikes sprite images along the bottom of the background
 - these are just 2D images – no collider, not tag
- create a new empty gameObject (named 'spikes-group')
- child all your spikes into that game object
 - i.e. drag each spikes image gameObject into spikes-group, so it becomes a child object of spikes-group
- tag your spikes-group 'Spikes'
- add to spikes-group a Box Collider 2D and tick its 'Is Trigger'
- size and position the collider in spikes-group, so that it stretches the whole width of the background, and that its top is in line with the spikes images

That's it – now you should only ever have a single collision with something tagged 'Spikes'

1.2 Alternative solution

An alternative would be to just have lots of untagged spikes sprites, and use the previous DEATH_Y y-position test to make the player lose a life and be repositioned when that Y position is hit.

There is no need for the empty gameObject containing the collider in this case, but it means the DEATH_Y value is 'hidden' in code, rather than an on-screen object that visually shows you where the action happens ...

1.3 If you don't like 'invisible' game objects ...

If you don't like invisible game objects with colliders, just use rectangular image as a sprite.

- Drag the sprite onto the scene, so it becomes a gameObject
- Rename it appropriately
- Resize it as required
- Add the collider
- Then un-tick it's renderer component, so the user cannot see the image

2 Stop guy jumping out of screen

2.1 The problem

Although the camera never moves to look past the edge of the background image, the hero-guy can still jump up/left/right off-screen.

2.2 The solution: objects with Box Colliders preventing move off screen

An object with a Box Collider 2D (that does NOT have 'Is Trigger' ticked), becomes a rectangle of the screen that the hero-guy is not allowed to enter.

So all we have to do is create some rectangles just off screen (LEFT, RIGHT, and if necessary TOP), so the hero-guy cannot move off screen.

Do the following

- create a new empty gameObject (e.g. named 'boundary')
 - or use any rectangular sprite image as the basis for your new gameObject
- add to the object a Box Collider 2D
 - do NOT tick its 'Is Trigger'
- size and position the gameObject / collider, so that the collider touches the edge of the game background
 - so the player can no longer move off screen because they'll hit the collider of this gameObject