Physics scripting project

Joe Place 991717262 January 27 2024

Bowling Alley top down

The scene will have a player able to move onto a booster arrow and be flung into objects mostly pins that the player can fling around

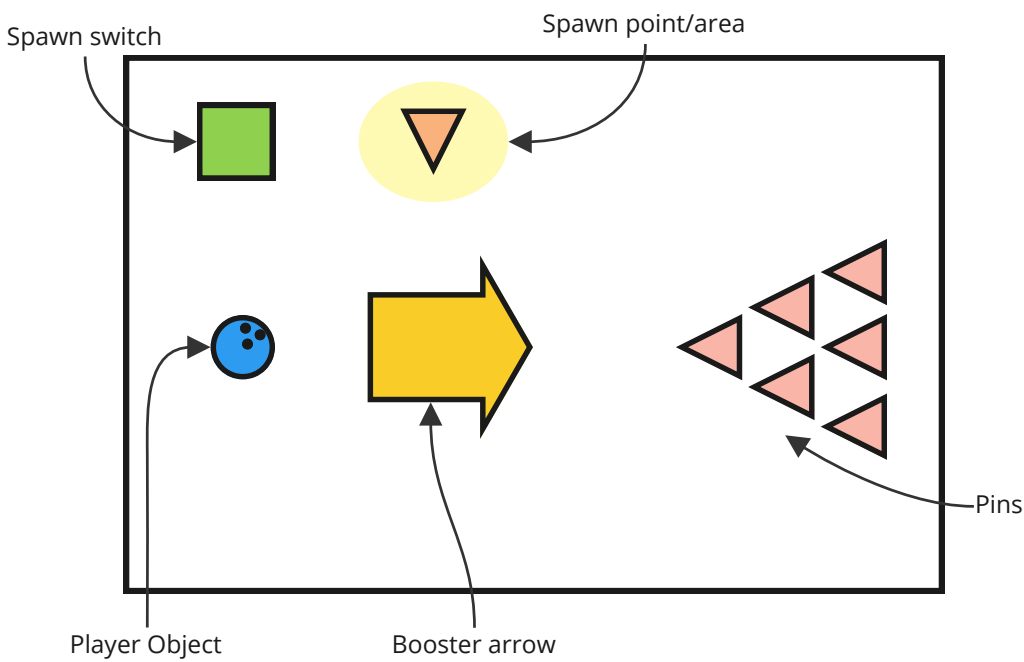
Main Functionality

Player can move in the physics engine (Topdown 2d)
The booster will send a player crashing into the objects
Pins that will change sprites once collided with
A reset switch that will reset the environment once the player collides with it

General Planning

All movement will be handled with the in engine physics engine with each object (Excluding booster and spawn point) having a RigidBody2D
Players will move with AddForce with inputs being Input.GetAxis for x and y movement
Booster area will be set to trigger and an area affector with OnTriggerEnter2D handling the increased speed of the object that enters it
Pins will be light objects that can be moved and flung around as well as be prefabs to be able to create multiple while the scene is running

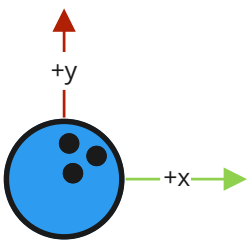
Sketch of the scene:



Player movement

The player will need to identify its rigidbody within its script (Hardcode to avoid forgetting manual declaring)
The player will need to have a variable that controls its direction: This will be a Vector to handle the two numbers within a contained variable
And a float variable to store the force that will be applied to the rigidBody so that it can move (the amount of force should be alterable as it is the players speed)

Each frame the computer should be checking for keyboard inputs for which direction the player wants to move the game object
To check for keyboard inputs is Input.GetAxis("<axis>");
This will be declared to the x and y of the vector controlling its direction in range of -1 and 1 values



The player should then move using the physics system in the direction they want
This can be done with AddForce applied to the rigidbody where an amount of force is added in the direction of the x and y vector for direction

The player will interact with the boxes using the in engine physics system requiring no scripting

Most of the other interactions the player will have with the game will be handled in the other game objects

The ball can spin as it moves with MoveRotation

Problem solving:
MoveRotation works but the direction is +/- dependant as well it will always rotate as it is in update with no trigger
Potential solution: have an if that checks if there is an input and which direction it will go
Actual solution: put direction. X in the MoveRotation statement and inverted the speed to rotate properly

Booster Arrow

The booster arrow will be a trigger that when the player is over it, propels them forward

This will be achieved with an area effector and a box collider set to trigger

another part of the arrow will be its ability to change color once the player is enters it giving a more satisfying experience.

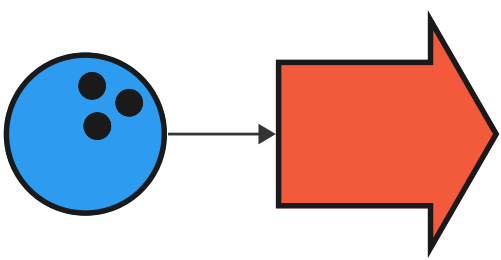
This can be done by getting the sprite rendered component in a script and calling the color function. Then it can be set to a color with Color._____.

the color change will be handled in OnTriggerEnter with OnTriggerExit reseting the color of the arrow.

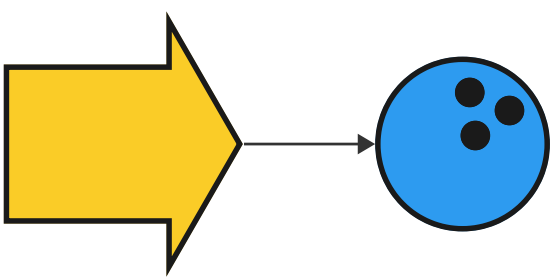
Problems discovered:
The color would not change when spriteR.color = Color.red was called
The problem was that the function was OnTriggerEnter not OnTriggerEnter2D

Booster Arrow demonstration:

Upon entering



Upon exiting



Spawn Block and Pin

Spawn Point

The spawn block will spawn a pin each time the player collides with the block the block will need a rigidBody2D with an empty game object to operate as the spawn point as a child to the block

The block will run a script that will require the spawn points position (its transform) and it will require the prefab of the pin

OnCollisionEnter2D the pin will instantiate at the position of spawn point

Pins

Pins will be simple rigidbodies with a circle collider

The pins will also be able to change states once collided with to emulate the pin being knocked over.
This will be handled in a script using the sprite renderer object that will change its color OnCollisionEnter but not reset once left.

they will then become prefabs so the spawn point can spawn as many as needed during run time

Spawning the pins

the block should spawn the pins in the traditional bowling layout on start as well as be able to spawn them on collision

So we could make 10 spawn points and orientate them as such and maybe make an arraylist to cycle through the object to spawn them 10 times and then on contact do the same but destroy them to prevent overlap

Problems

Making a list of transforms did not work as new transform doesn't work so it is now a list of game objects that we can still reference their transform during instantiate
How it would work initially would create new gameobjects instead of being able to search for existing ones
FindObjectsOfType<transform>(); (the list version) allows for it to find transforms and have them in an array
FindObjectsOfType moved into OnCollision

Additional problems

The loop was unable to detect or reference Pin as a class to reference an internal function So pin spawning loop was moved into the start and now the collision of the spawn box now resets the scene

During reset after initial start the pins are spawned in the bowling ball and spawn blocks transforms.
Giving each spawnpoint a tag and doing game object.findgameobjectwithtag("thetag") allowed it to find the game objects and then do .transform to spawn them at the specific places

However spawning now happens during start with the switch now resetting the scene itself instead of deleting and respawning