Pinball Game

4) At least one use of a trigger event to initiate or influence a scripted action, using:

`OnTriggerEnter2D` or **`OnTriggerExit2D`** or **`OnTriggerStay2D`**

Floor
(despawner)

5) At least one use of a collision event to initiate or influence a scripted action, using:

`OnCollisionEnter2D` or **`OnCollisionExit2D`** or **`OnCollisionStay2D`**

Paddles

1) At least one example of animation or kinematic motion using position or rotation:

translation using **`transform.Translate`** or **`rigidbody2D.MovePosition`**
and/or
rotation using **`transform.Rotate`** or **`rigidbody2D.MoveRotation`**

Moving ball spawn

2) At least one example of motion using forces:

`ridgidbody2D.AddForce` or **`.AddForceAtPosition`** or **`.AddRelativeForce`** or **`.AddTorque`**

Paddles

3) At least one of the uses of motion must be initiated or influenced by keyboard input using:

`Input.GetKey` or **`.GetKeyDown`** or **`.GetKeyUp`**
or
`Input.GetAxis("Horizontal")` or **`Input.GetAxis("Vertical")`**

Paddles

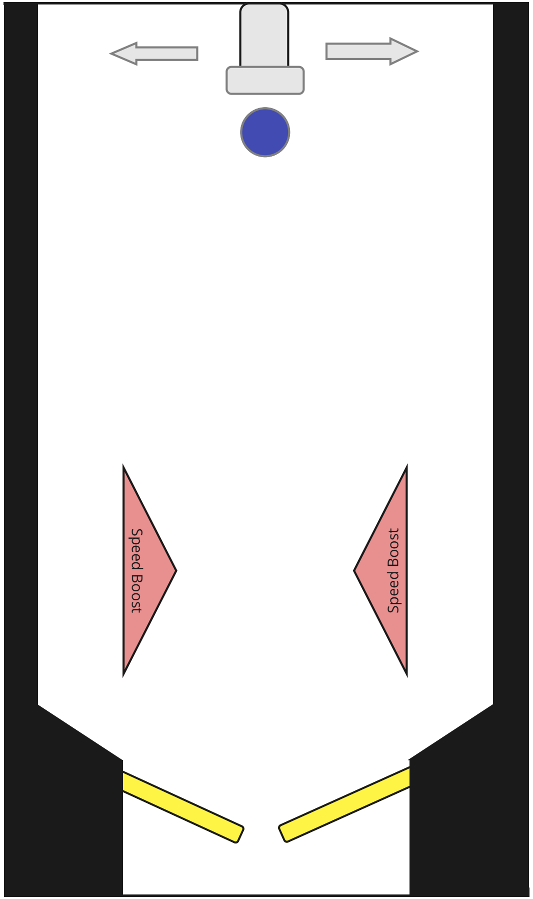
6) At least one Prefab must be instantiated at runtime

Ball

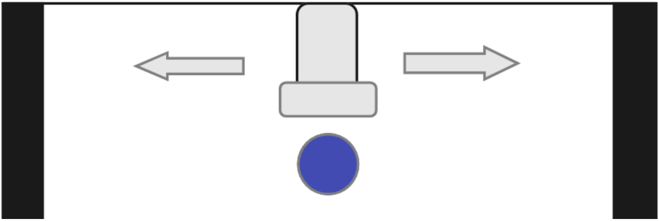
Moving ball spawn

Constraints:

- Do not use mouse input
- Do not use any Unity components not covered in Weeks 1-3 (see list of allowed components)
- Do not use any additional Unity packages except Cinemachine
- Do not import any Unity assets (eg from the Asset Store, GitHub, tutorials, etc) except images. That includes prefabs, scripts, scenes, projects
- Do not use Bolt, Visual Scripting, Playmaker, etc. Your code must be written in C#
- Do not use Copilot or any generative AI (GPT etc) to write, generate, or fix code



Ball Drop



Intended functionality:

- The spawner to move along horizontal input (#1)
- Instantiate ball on runtime (#6)
- Loop ball in game

Problem Solving:

Dropper shouldn't be part of physics but still move

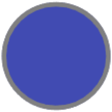
- Kinematic body

Dropper should move according to Horizontal input.

- *direction* variable = Input.GetAxis Horizontal
- In Update: Transform.translate(1*direction*Time.deltaTime, 0, 0)

Dropper should instantiate a ball at runtime, then only have one instance at a given moment.

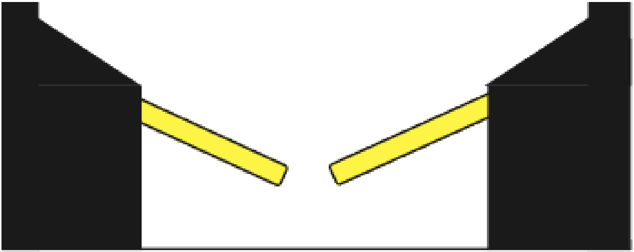
- Make empty object as spawn point for ball
- At Start: Instantiate ball prefab, then
- In Floor Update: check for collision (which should only be the ball), then instantiate a new one at spawn point. reference spawn position through the empty object.



Ball will have

- Dynamic body
- will use motion using forces

Paddles



Intended functionality:

- swivel on spacebar (#3)
- simulate launching (#2 & #5)
- collide with ball

Problem Solving:

Swivel on spacebar

- create empty objects on pivot points
- find (trial and error) on a good angle to receive ball.
- Input.GetKey on spacebar

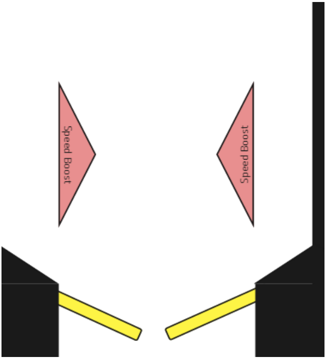
Simulate on launching

- use rigidbody2D physics methods (#2)
- only apply force boost while on the paddle using OnCollisionStay2D (#5)
- maybe effectors? see next component.

Collide with ball

- rigidbody2D kinematic type
- box collider
- have its own layer so script can discriminate between collisions.

Physics & Terrain



Intended functionality:

- Walls (keep in bounds)
- Floor deletes fallen balls.
- directional speed applied to ball
- improve paddle physics?

Problem Solving:

Walls

- Static rigidbody2D.

Floor

- Destroys the ball using OnTriggerExit2D (#4)
- Its own layer so ball can tell when to delete (so it doesn't delete on just any collision exit)

Directional Speed Boost

- Surface effector on polygon collider? experiment with this and point effect to find best one.

Paddles need to feel rubbery so ball bounces against it

- Maybe try having a buoyancy effector with really high density?
  - https://www.youtube.com/watch?v=p0n6EFR1M8c