**Game Name:** Docking

**Game Type:** 2D

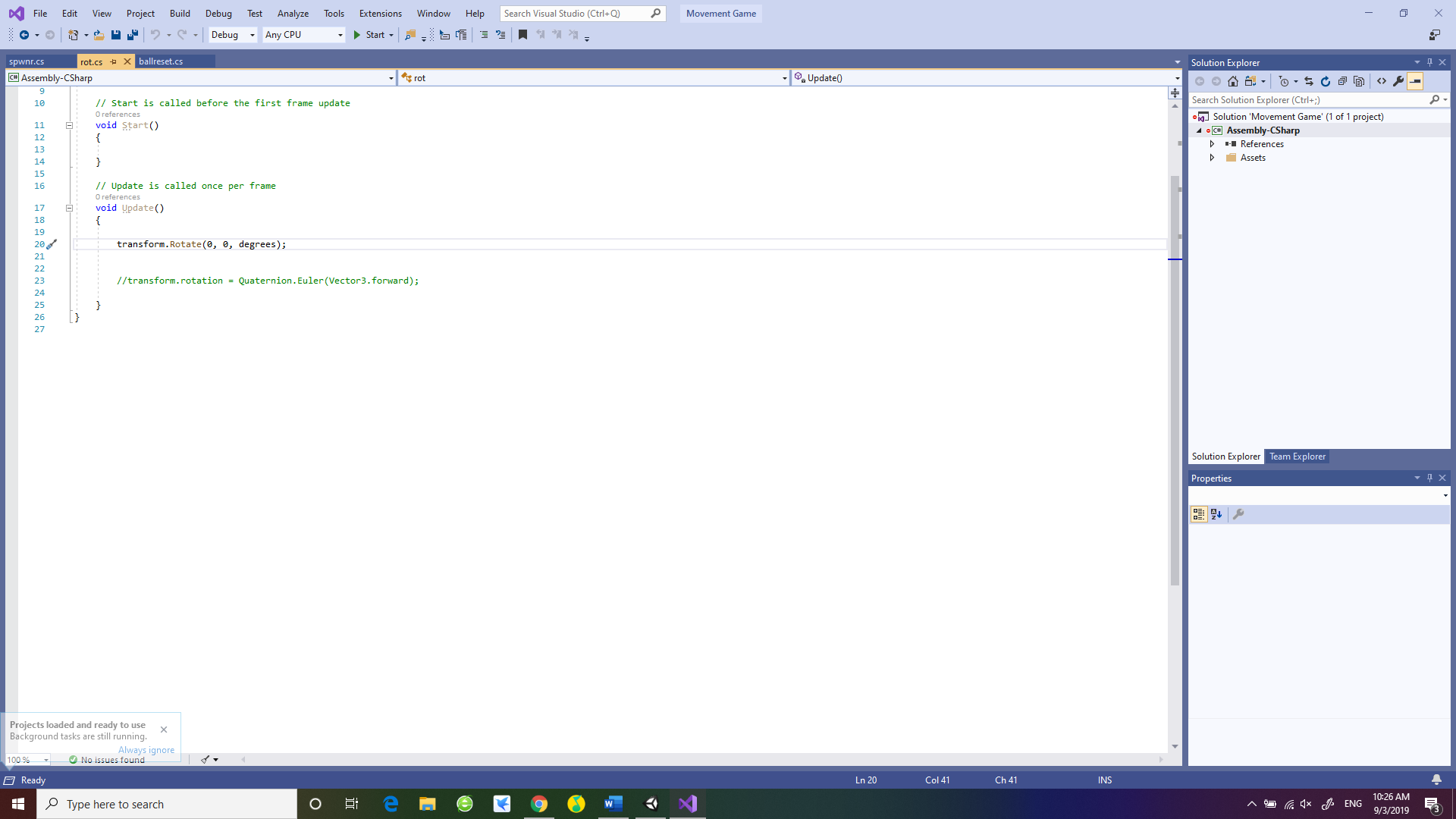
**2019/09/03**

**CONCEPT**

The initial idea was generated through the exploration of implementing the essence of non-interactivity into the very interactive (in many aspects) games. I aim to put emphasis on the key factors of game construction that manipulates the psychology of the players, and implementing barriers to the plyers while without jeopardizing user’s involvement. My key inspiration was derived from the DVD bouncing “game”, where participants awaits the moment where the DVD logos hit the corner of the display. Through my exploration on the mechanism of such play experience, I discovered that the core of the game is much simpler than expected. There are little differences compare to an interactive game. There is a goal, and there is a desire to accomplish the goal. The main difference lies on the preservation of the involvement of players in the absence of active control, and the very limit itself intensifies further desire in achieving, in another word, satisfaction. Therefore, my game has the core that evokes investment into the completion of the goal through anticipation and desire in the absence of active involvement. During my very first prototype, I designed the game with a central circle with for openings on the 4 directions of the axis. Projectiles were designed to launch from a random range from the edge of the viewing window, and the goal is to await the projectile to enter the circle through the openings. However, I soon changed my prototype due to a major flaw, where that the mechanism is too static. The attention spam of the player will son dissipate with a stationary target that contributes to “an easy game”, which is not my intention. As a result, I reduced the amount of entry point on the target to only one, and I made the central target spin; I created a spawner on the top of the window to periodically add a new ball into the scene to increase the possibilities of success/hope. In effect, the target is no longer static but necessary in relation to the position of the ball.

**Technical Implementations**

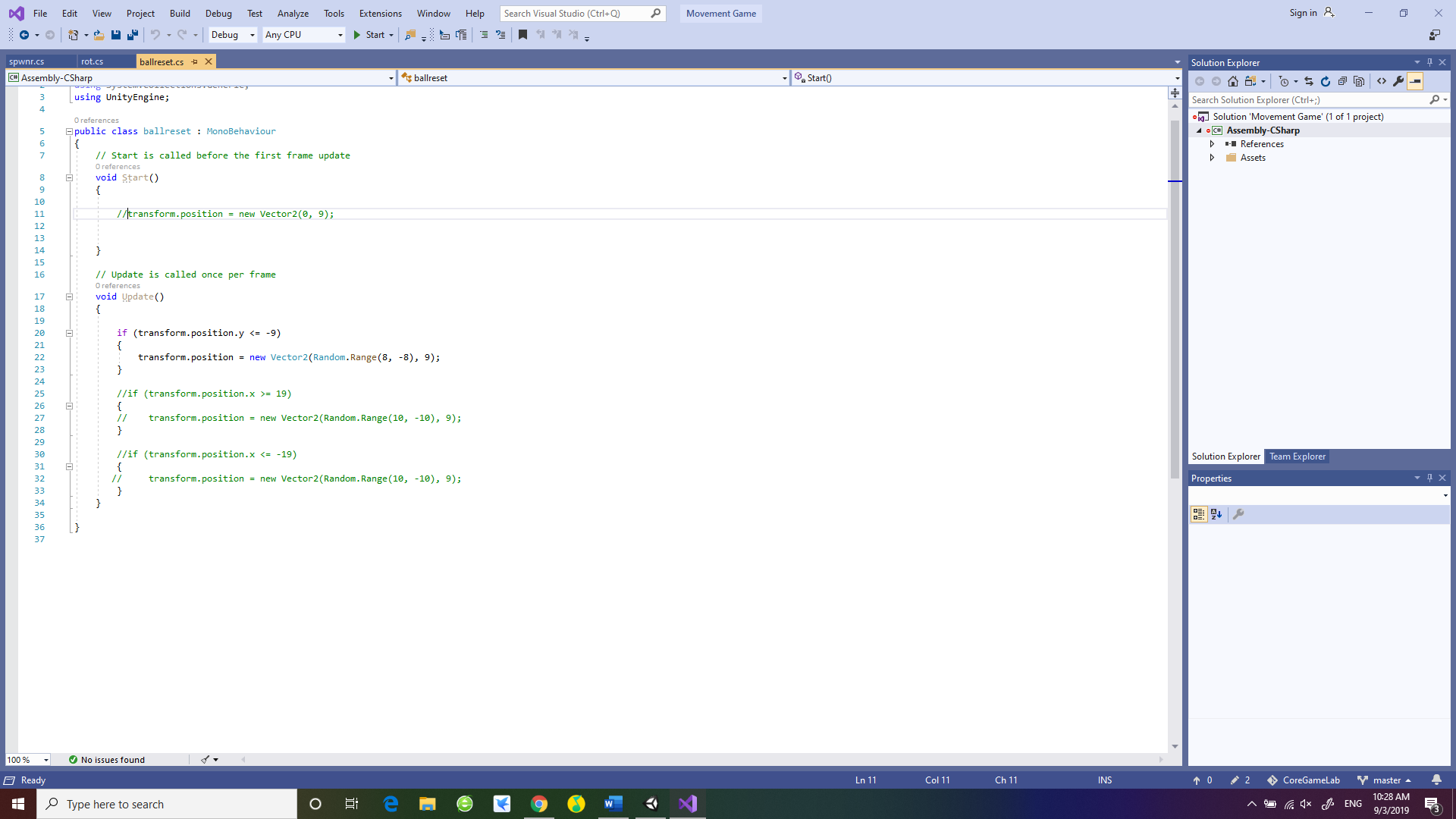
* Target Rotation



Rotate transformation function with a public degrees toggle that allows manual adjustments of the magnitude of rotational displacement per frame in degrees.

* Ball Reset

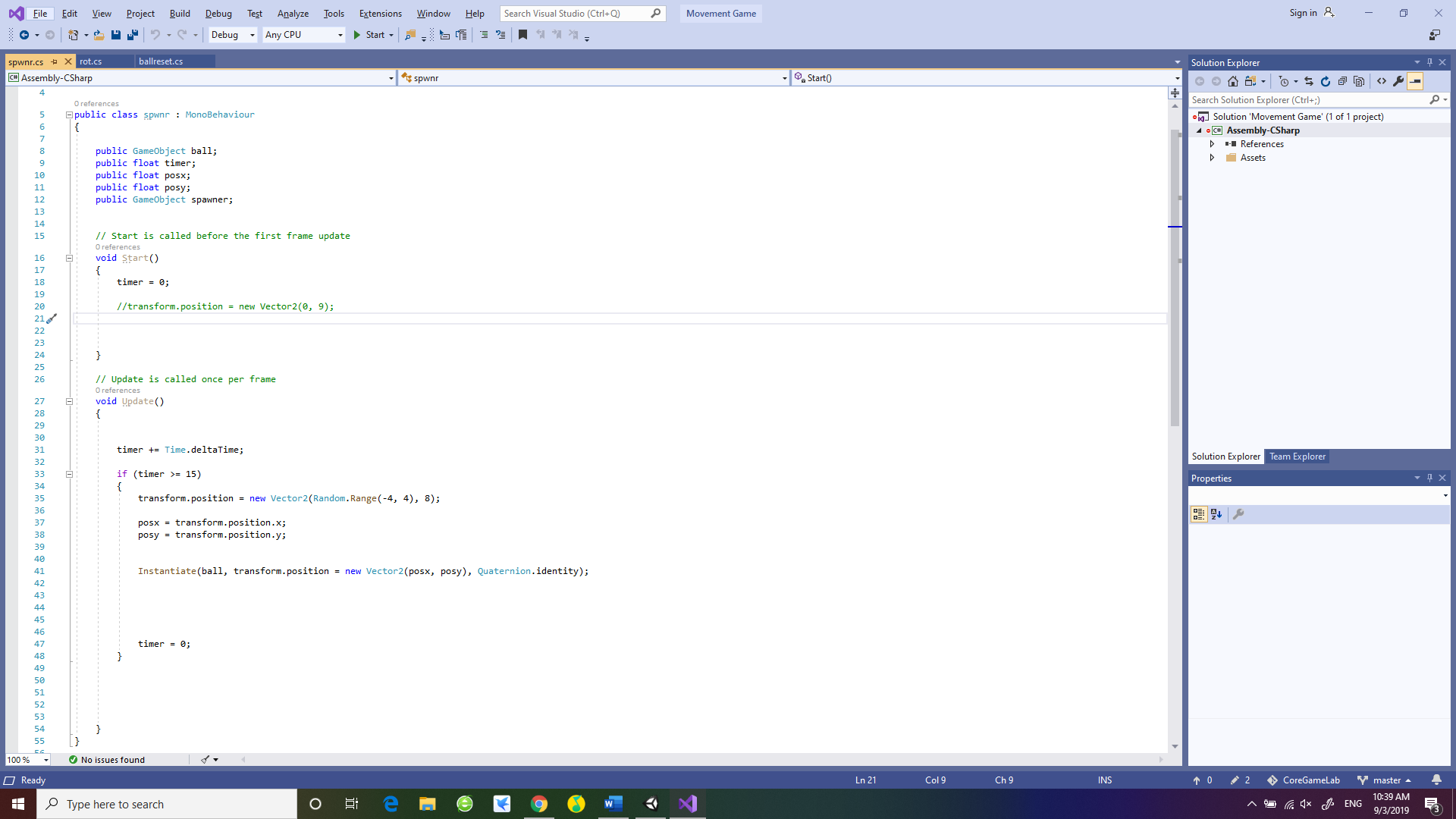
Using IF functions to establish boundaries to reset the displacement of the ball.



* Spawner

Using Instantiate to spawn and clone objects, and using timer to assign periodic actions.

Radom range displacements of the spawner and the ball, and tracking of the ball in relation to the displacement of the spawner.



**Software physics simulations**

* Collider, for the registration of hitboxes.
* Rigid Body, for the simulation of object in virtual space.
* 2D Physics, for the adjustments of material behaviors.