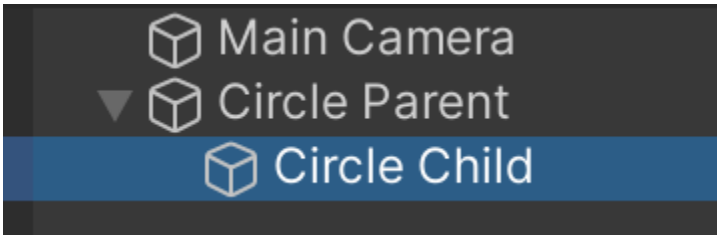
Problem #1

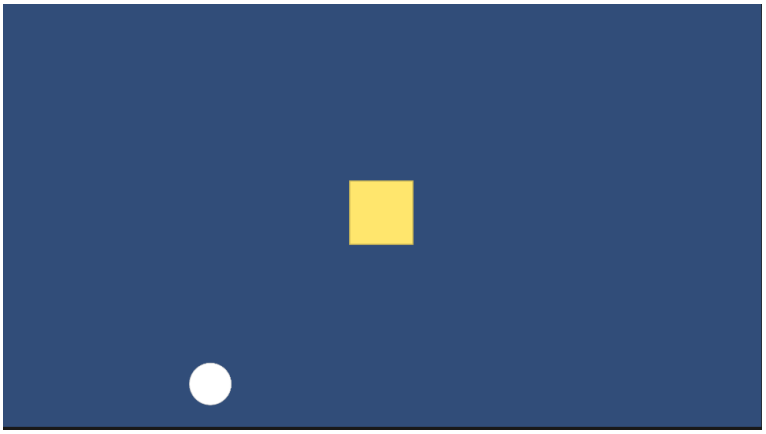
Description: Make the game object “Circle” blinks white and red. The circle should be white for 1 second and should become red for 0.5 (half a second) before switching back to white and so on.

Problem #2



Description: Do the same as Problem #1. But this time, place your script in the parent game object “Circle Parent” and leave the Sprite Renderer component inside “Circle Child”.

Problem #3



Description: Make “Circle Bullet” go diagonally up & down. When the circle goes over the “Square Target”, make the square disappear and when it goes away make the square appear again.

Rules:

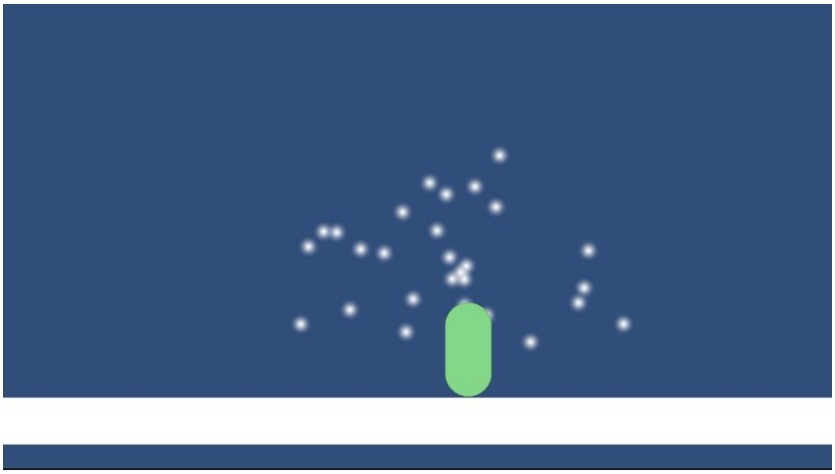
- Don’t let the square change its position.

- Make the variable that holds the speed of the circle visible and changeable from within the inspector.

Problem #4

Description: You have a game object named “Capsule” that holds an Animator component. The Animator has 2 animations: Capsule_Jump and Capsule_Damage. Let the capsule play the jump animation when the player presses SPACE and let it play the damage animation when the player presses TAB.

Problem #5

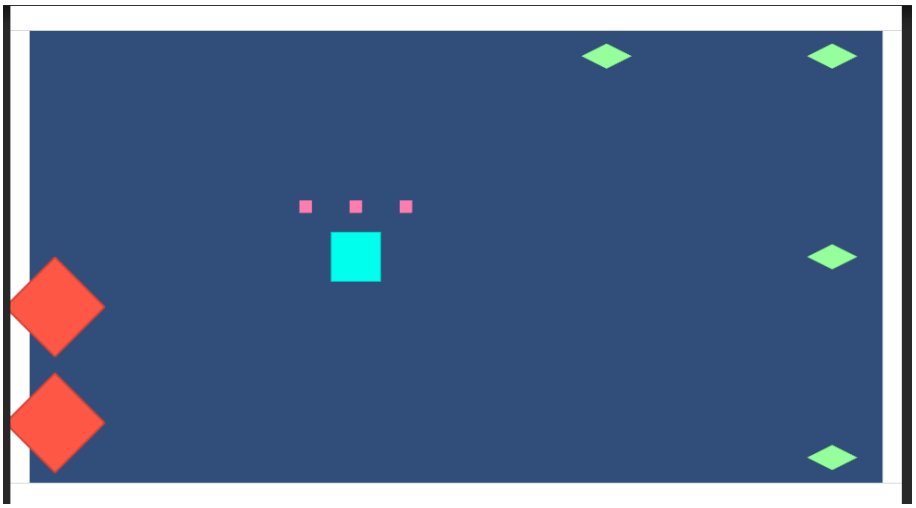


Description: Make the capsule moveable horizontally based on the received input. The capsule should have acceleration so it should not reach the maximum speed instantly. The capsule should have a particle system that should continue playing while the capsule is moving and should stop playing when the capsule stops moving. The particles inside the particle system should dynamically change their “start speed” based on how fast the capsule is moving (when maximum speed of capsule is reached, the speed of particles should also reach their maximum speed).

Rules:

- These variables should be made visible and changeable from the inspector: **Max Speed of Capsule**, **Acceleration of Capsule**, & **Max Speed of Particles in the particle system**.

Problem #6

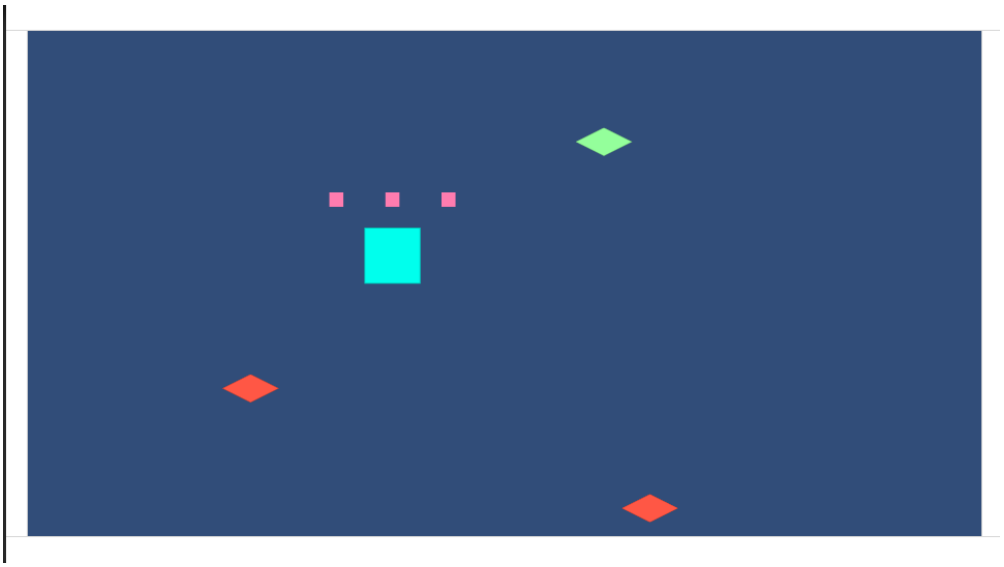


Description: The cyan-colored square is the player. The square should move both horizontally & vertically based on the input received from the player. There should be a health system consisting of 3 life points and it should be visible to the player as counters. Red diamonds are damaging game objects that will decrease the number of life points the player has when he/she collides with them by one (-1). Green diamonds are healing game objects that will increase the life points by one (+1). Each time the number of life points changes, a text should appear on the screen indicating either: “You have been hurt!” or “You have been healed!”. This text should appear for a limited amount of time.

Rules:

- The health points should be shown right above the cyan square and should follow the player’s movement. Also, they should always be updated to truly represent how much life points are left.
- Once colliding with any of the 2 types of diamonds, they should always disappear.
- This variable should be made visible and changeable from the inspector: **Duration of the indicating text**.

Problem #7



Description: Use the same script created for the player cyan square in the previous problem here, as we will need the same mechanic and health system.

This time we want to have two game objects to spawn (initiate) the diamonds. You can find and use the prefabs “Damaging Diamond” & “Healing Diamond” inside the folder “Prefabs”. Create one game object “Healing Spawner” for spawning green healing diamonds, and one game object “Damaging Spawner” for spawning red damaging diamonds. Each spawner should have the variable “Spawn Interval” that controls how often it should spawn. For example, if the Healing Spawner’s Spawn Interval is set to 7, this means it will spawn a healing diamond every 7 seconds.

All diamonds should be spawned from the right at the same point on the x-axis (use the x position **9**). However, they should be spawned at random points on the y-axis (use the range **-4.5** to **4.5**). Diamonds should also move to the left at a constant speed.

Also, there should be a game object in the scene that is responsible for destroying all diamonds when they reach the far left of the screen (we don’t need them anymore when they get out of the reach of player).

Rules:

- Once colliding with any of the 2 types of diamonds, they should always disappear (like the previous problem).
- This variable should be made visible and changeable from the inspector: **Spawn Interval**.