**Version Control**

October 25, 2016

Assignment v1.0

* + Started with the assignment.
  + Created Sprites, Scene, Prefab, Sounds and Scripts folders.
  + Created the background, background controller.
  + Created the Player, Player Controller Script, Player Collider Script
    - Added Rigidbody 2D and Polygon Collider 2D on the player.
  + Created the Enemy, Enemy Controller Script.
    - Added Polygon Collider 2D on the enemy.

*///----------------------------------------------------------------------------------*  
*///   Source File Name:* BackgroundController.cs  
*///   Author's Name:  Roderick Rodelas*  
*///   Student Number: 100978575*  
*///   Program Description: This controls the player's movement.*  
*///   Date: October 25, 2016*  
*///   Last Modified by: Roderick Rodelas*  
*///   Date last modified:*  
*///   Revision History:*  
*///----------------------------------------------------------------------------------*  
  
using UnityEngine;  
using System.Collections;  
  
public class BackgroundController : MonoBehaviour {  
  
        [SerializeField]  
        private float speed;  
  
  
        private Transform \_transform;  
        private Vector2 \_currentPosition;  
  
        *// Use this for initialization*  
        void Start () {  
              
                \_transform = gameObject.transform;  *//game.Object.GetComponent<Transform>;*  
                \_currentPosition = \_transform.position;  
                Reset ();  
            }  
          
        *// Update is called once per frame*  
        void FixedUpdate () {  
  
                \_currentPosition = \_transform.position;  
  
                \_currentPosition -= new Vector2 (speed, 0);  
                \_transform.position = \_currentPosition;  
  
                if (\_currentPosition.x <= -18f) {  
                        Reset ();  
                    }  
            }  
        private void Reset(){                          
                \_currentPosition = new Vector2 (0, 0);  
                \_transform.position = \_currentPosition;              
            }  
}

*///----------------------------------------------------------------------------------*  
*///   Source File Name:* PlayerController.cs  
*///   Author's Name:  Roderick Rodelas*  
*///   Student Number: 100978575*  
*///   Program Description: This controls the player's movement.*  
*///   Date: October 25, 2016*  
*///   Last Modified by: Roderick Rodelas*  
*///   Date last modified:*   
*///   Revision History:*   
*///----------------------------------------------------------------------------------*  
  
  
  
  
using UnityEngine;  
using System.Collections;  
  
public class PlayerController : MonoBehaviour {  
  
    [SerializeField]  
    private float speed;  
  
    [SerializeField]  
    private GameObject PlayerBullet;  
  
    [SerializeField]  
    private GameObject bulletPosition;  
  
    private Transform \_transform;  
    private Vector2 \_currentPosition;  
    private float \_playerInputX;  
    private float \_playerInputY;  
  
  
    *// Use this for initialization*  
    void Start () {  
        \_transform = gameObject.transform;  
        \_currentPosition = \_transform.position;  
    }  
          
    *// Update is called once per frame*  
    void FixedUpdate () {  
              
        \_playerInputX = Input.GetAxis ("Horizontal");  
        \_playerInputY = Input.GetAxis ("Vertical");  
        \_currentPosition = \_transform.position;  
  
        *//fire bullets*  
        if (Input.GetKeyDown ("space"))   
        {  
            GameObject bullet = (GameObject)Instantiate (PlayerBullet);  
            bullet.transform.position = bulletPosition.transform.position;  
  
        }  
  
        *//move right*  
        if (\_playerInputX > 0)   
        {  
            \_currentPosition += new Vector2 (speed, 0);  
        }  
  
        *//move left*  
        if (\_playerInputX < 0)   
        {  
            \_currentPosition -= new Vector2 (speed, 0);  
        }  
  
        if (\_playerInputY > 0)   
        {  
            \_currentPosition += new Vector2 (0, speed);  
        }  
  
        *//move left*  
        if (\_playerInputY < 0)   
        {  
            \_currentPosition -= new Vector2 (0, speed);  
        }  
        *//boundary movement for player*  
        *//check's player position on x Axis*  
        if (\_currentPosition.x < -18f) {  
            \_currentPosition.x = -18f; *//if less than the boundary mark, make it the boundary mark*  
        }  
        *//check's player position on x Axis*  
        if (\_currentPosition.x > -1f) {  
            \_currentPosition.x = -1f;  *//if greater than the boundary mark, make it the boundary mark*  
        }  
        *//check's player position on y Axis*  
        if (\_currentPosition.y < -6f) {  
            \_currentPosition.y = -6f;  *//if less than the boundary mark, make it the boundary mark*  
        }  
        *//check's player position on y Axis*  
        if (\_currentPosition.y > 6f) {  
            \_currentPosition.y = 6f;  *//if greater than the boundary mark, make it the boundary mark*  
        }  
  
        \_transform.position = \_currentPosition;  
      
    }  
          
}

*///----------------------------------------------------------------------------------*  
*///   Source File Name:* EnemyController.cs  
*///   Author's Name:  Roderick Rodelas*  
*///   Student Number: 100978575*  
*///   Program Description: This controls the enemy's action.*  
*///   Date: October 25, 2016*  
*///   Last Modified by: Roderick Rodelas*  
*///   Date last modified:*   
*///   Revision History:*   
*///----------------------------------------------------------------------------------*  
  
  
using UnityEngine;  
using System.Collections;  
  
public class EnemyController : MonoBehaviour {  
  
    [SerializeField]  
    private GameObject EnemyBullet;  
    [SerializeField]  
    private GameObject bulletPos;  
    [SerializeField]  
    private float speed;  
  
    private Vector2 \_currentPosition;  
    private Transform \_transform = null;  
  
    *//Constants*  
    private const float startPosition = 1.75f;  
    private const float resetPosition = -20f;  
    private const float minY = -6f;  
    private const float maxY = 6f;  
    private bool shoot = true; *// use for shooting*  
  
    *// Use this for initialization*  
    void Start () {  
        \_transform = gameObject.transform;  
        \_currentPosition = \_transform.position;  
        Reset ();  
    }  
  
    void FixedUpdate () {  
  
  
        \_currentPosition = \_transform.position;  
  
        *//move the enemy with the given speed*  
        \_currentPosition -= new Vector2(speed, 0);  
  
        *//shoot a bullet*  
        if (shoot == true) {  
            GameObject bullet = (GameObject)Instantiate (EnemyBullet);  
            bullet.transform.position = bulletPos.transform.position;  
            shoot = false;  
        }  
  
        \_transform.position = \_currentPosition;  
        *//if enemy goes outside the screen*  
        if (\_transform.position.x < resetPosition) {  
            Reset ();  
        }  
  
    }  
  
    *//reset the enemy.*  
    public void Reset(){  
           
        float startY = Random.Range (minY, maxY);  
        \_currentPosition = new Vector2 (startPosition, startY);  
        \_transform.position = \_currentPosition;  
        shoot = true;  
  
    }  
  
    public void OnTriggerEnter2D(Collider2D other){  
        Reset();  
    }      
}

*///----------------------------------------------------------------------------------*  
*///   Source File Name:* PlayerCollider.cs  
*///   Author's Name:  Roderick Rodelas*  
*///   Student Number: 100978575*  
*///   Program Description: This controls the enemy's action.*  
*///   Date: October 25, 2016*  
*///   Last Modified by: Roderick Rodelas*  
*///   Date last modified:*   
*///   Revision History:*   
*///----------------------------------------------------------------------------------*  
  
  
  
  
using UnityEngine;  
using System.Collections;  
  
public class PlayerCollider : MonoBehaviour {  
  
    [SerializeField]  
    GameObject explosion = null;  
  
  
    public void OnTriggerEnter2D(Collider2D other){  
  
        if (other.gameObject.tag == "enemyBullet")   
        {  
            Player.Instance.Lives -= 1;  
            Debug.Log ("Collision with " + other.gameObject.tag);  
        } else if (other.gameObject.tag=="enemy") {  
  
            Player.Instance.Lives -= 1;  
            Debug.Log ("Collision with " + other.gameObject.tag);  
  
            EnemyController sp = other.gameObject.GetComponent<EnemyController>();  
            if (sp != null) {  
                *//reset enemy*  
                sp.Reset();  
  
                *//show explosion*  
                GameObject ex = Instantiate (explosion);  
                ex.transform.position = sp.transform.position;  
  
            }  
        }  
    }  
}

October 26, 2016

Assignment v1.1

* + Added the bullet sprites (player and enemy). Bullet Scripts.
    - Added Rigidbody 2D and Circle Collider 2D on the player bullet.
    - Added Circle Collider 2D on enemy bullet.
  + Started adding UI canvas and added points and lives.
  + Created the Player class.
  + Added text and button on UI and created the HUDController Script

*///----------------------------------------------------------------------------------*  
*///   Source File Name:* Player.cs  
*///   Author's Name:  Roderick Rodelas*  
*///   Student Number: 100978575*  
*///   Program Description: This controls the enemy's action.*  
*///   Date: October 26, 2016*  
*///   Last Modified by: Roderick Rodelas*  
*///   Date last modified:*  
*///   Revision History:*  
*///----------------------------------------------------------------------------------*  
  
  
  
  
using UnityEngine;  
using System.Collections;  
  
public class Player {  
  
        private const string key = "HIGH\_SCORE";  
  
        private int \_points = 0;  
        private int \_lives = 3;  
        private int \_highScore = 0;  
        public HUDController hud;  
  
        private static Player \_instance = null;  
        public static Player Instance{  
              
                get {  
                        if (\_instance == null) {  
                                \_instance = new Player ();  
                            }          
                        return \_instance;  
                    }  
            }  
  
        private Player(){  
  
                if (PlayerPrefs.HasKey (key)) {  
                        \_highScore = PlayerPrefs.GetInt (key);  
                    }  
            }  
  
        public int Points{    
                get{ return \_points; }  
                set{ \_points = value;   
                        hud.updatePoints();  
                          
                        if (value > \_highScore) {  
                                PlayerPrefs.SetInt (key, value);  
                                \_highScore = value;  
                            }  
                    }  
            }  
  
        public int Lives{  
                get{ return \_lives; }  
                set{ \_lives = value;   
                        hud.updateLives();  
                        if(\_lives <= 0){  
                            *//GameOver*  
                                hud.GameOver();  
                            }  
                    }  
            }  
          
        public int HighScore{  
                get{ return \_highScore; }  
  
            }  
}

*///----------------------------------------------------------------------------------*  
*///   Source File Name:* BulletCollider.cs  
*///   Author's Name:  Roderick Rodelas*  
*///   Student Number: 100978575*  
*///   Program Description: This controls the bullet collision and detection.*  
*///   Date: October 26, 2016*  
*///   Last Modified by: Roderick Rodelas*  
*///   Date last modified: October 26, 2016*  
*///   Revision History: added the collision conditions*  
*///----------------------------------------------------------------------------------*  
  
  
  
using UnityEngine;  
using System.Collections;  
  
public class BulletCollider : MonoBehaviour {  
  
  
    public void OnTriggerEnter2D(Collider2D other){  
  
        if (other.gameObject.tag=="enemy") {  
            Player.Instance.Points += 10;  
            Debug.Log ("Collision with " + other.gameObject.tag);  
  
            EnemyController sp = other.gameObject.GetComponent<EnemyController>();  
                if (sp != null) {                      
                    *//reset enemy*  
                    sp.Reset();  
  
                    *//destroy missile*  
                    Destroy (gameObject);  
                      
                }  
        }  
    }  
}

*///----------------------------------------------------------------------------------*  
*///   Source File Name:EnemyBulletController.cs*  
*///   Author's Name:  Roderick Rodelas*  
*///   Student Number: 100978575*  
*///   Program Description: This controls the enemy's bullet.*  
*///   Date: October 26, 2016*  
*///   Last Modified by: Roderick Rodelas*  
*///   Date last modified: October 27, 2016*  
*///   Revision History: Update boundary - October 26, 2016*  
*///                     Added sounds - October 27, 2016*  
*///----------------------------------------------------------------------------------*  
  
using UnityEngine;  
using System.Collections;  
  
public class EnemyBulletController : MonoBehaviour {  
  
  
    *//bullet speed*  
    [SerializeField]  
    private float speed;  
  
    *// Use this for initialization*  
    void Start () {  
  
    }  
  
    *// Update is called once per frame*  
    void FixedUpdate () {  
            Vector2 currentPos = transform.position; *//bullet current position*  
  
        currentPos -= new Vector2(speed, 0) ; *//bullets new position*  
            transform.position = currentPos;*//update bullet position*  
  
            if (transform.position.x <= -20f) {  
                Destroy (gameObject);  
            }  
    }  
  
    public void OnTriggerEnter2D(Collider2D other){  
        Destroy (gameObject);  
    }      
}

*///----------------------------------------------------------------------------------*  
*///   Source File Name:HUDController.cs*  
*///   Author's Name:  Roderick Rodelas*  
*///   Student Number: 100978575*  
*///   Program Description: This controls the player's movement.*  
*///   Date: October 26, 2016*  
*///   Last Modified by: Roderick Rodelas*  
*///   Date last modified:*  
*///   Revision History:*   
*///----------------------------------------------------------------------------------*  
  
using UnityEngine;  
using System.Collections;  
using UnityEngine.UI;  
  
public class HUDController : MonoBehaviour {  
  
    [SerializeField]  
    Text points = null;  
  
    [SerializeField]  
    Text lives = null;  
  
    [SerializeField]  
    GameObject player = null;  
  
    [SerializeField]  
    GameObject enemy1 = null;  
  
    [SerializeField]  
    GameObject enemy2 = null;  
  
    [SerializeField]  
    GameObject enemy3 = null;  
  
    [SerializeField]  
    GameObject powerUp = null;  
  
    [SerializeField]  
    GameObject shield = null;  
  
    [SerializeField]  
    Text gameover = null;  
  
    [SerializeField]  
    Text highscore = null;  
  
    [SerializeField]  
    Text gametitle = null;  
  
    [SerializeField]  
    Text instruct = null;  
  
    [SerializeField]  
    Text playerScore = null;  
  
    [SerializeField]  
    Button resetButton = null;  
              
    private int pointsLife = 1000;  
    private int pointsShield = 500;  
  
    void initiliaze(){  
        *//initialize the game*  
        Player.Instance.hud = this;  
        points.gameObject.SetActive (false);  
        lives.gameObject.SetActive (false);  
        player.SetActive (false);  
        enemy1.SetActive (false);  
        enemy2.SetActive (false);  
        enemy3.SetActive (false);  
        playerScore.gameObject.SetActive (false);  
        instruct.gameObject.SetActive (true);  
        gametitle.gameObject.SetActive (true);  
        gameover.gameObject.SetActive (false);  
        shield.gameObject.SetActive (false);  
        powerUp.gameObject.SetActive (false);  
        highscore.text = "High Score: " + Player.Instance.HighScore;  
        highscore.gameObject.SetActive (true);  
        resetButton.gameObject.SetActive (true);  
    }  
  
    void Start () {  
        initiliaze ();  
    }  
  
          
    void Update () {  
    }  
  
    public void updateLives(){  
        *//this updates player lives and points*  
        lives.text = "Lives: " + Player.Instance.Lives;  
    }  
  
    public void updatePoints(){  
        *//this update player points*  
        points.text = "Points: " + Player.Instance.Points;  
        *//add 1 life for every 1000 score*  
        if (Player.Instance.Points == pointsLife) {  
            Player.Instance.Lives += 1;  
            pointsLife += 1000;  
        }  
        if (Player.Instance.Points == pointsShield) {  
            powerUp.gameObject.SetActive (true);  
            pointsShield += 500;  
        }  
    }  
  
    public void GameOver(){  
        points.gameObject.SetActive (false);  
        lives.gameObject.SetActive (false);  
        player.SetActive (false);  
        enemy1.SetActive (false);  
        enemy2.SetActive (false);  
        enemy3.SetActive (false);  
        gametitle.gameObject.SetActive (true);  
        gameover.gameObject.SetActive (true);  
        powerUp.gameObject.SetActive (false);  
        playerScore.text = "Score: " + Player.Instance.Points;  
        highscore.text = "High Score: " + Player.Instance.HighScore;  
        highscore.gameObject.SetActive (true);  
        playerScore.gameObject.SetActive (true);  
        resetButton.gameObject.SetActive (true);  
        Player.Instance.Lives = 3;  
        Player.Instance.Points = 0;  
  
    }  
  
    public void ResetGame(){  
        points.gameObject.SetActive (true);  
        lives.gameObject.SetActive (true);  
        player.SetActive (true);  
        enemy1.SetActive (true);  
        enemy2.SetActive (true);  
        enemy3.SetActive (true);  
        instruct.gameObject.SetActive (false);  
        gametitle.gameObject.SetActive (false);  
        gameover.gameObject.SetActive (false);  
        powerUp.gameObject.SetActive (false);  
        playerScore.gameObject.SetActive (false);  
        highscore.gameObject.SetActive (false);  
        resetButton.gameObject.SetActive (false);  
        Player.Instance.Lives = 3;  
        Player.Instance.Points = 0;  
    }  
}

October 27, 2016

Assignment v1.2

* + Added explosion sprite (animation), script.
  + Added the explosion sound

*///----------------------------------------------------------------------------------*  
*///   Source File Name: ExplosionController.cs*  
*///   Author's Name:  Roderick Rodelas*  
*///   Student Number: 100978575*  
*///   Program Description: This controls the player's movement.*  
*///   Date: October 25, 2016*  
*///   Last Modified by: Roderick Rodelas*  
*///   Date last modified:*  
*///   Revision History:*  
*///----------------------------------------------------------------------------------*  
  
using UnityEngine;  
using System.Collections;  
  
public class ExplosionController : MonoBehaviour {  
  
    *// Use this for initialization*  
    void Start () {  
      
    }  
      
    *// Update is called once per frame*  
    void Update () {  
      
    }  
  
    public void End(){  
        Destroy (gameObject);  
    }  
}

October 28, 2016

Assignment v1.3

* + Added powerup sprite and script.
  + Added shield sprite and script.
  + Modified the HUDController to show the powerup sprite whenever player reaches 500 points.
  + Modified the HUDController to add 1 life whenever player reaches 1000 points.
  + Added comments on codes to clarify certain methods.

*///----------------------------------------------------------------------------------*  
*///   Source File Name:HUDController.cs*  
*///   Author's Name:  Roderick Rodelas*  
*///   Student Number: 100978575*  
*///   Program Description: This controls the player's movement.*  
*///   Date: October 26, 2016*  
*///   Last Modified by: Roderick Rodelas*  
*///   Date last modified: October 27, 2016*  
*///   Revision History: Added enemy, player set active conditions.  - October 26, 2016*  
*///                     Added powerup and shield conditions. - October 27, 2016*  
*///----------------------------------------------------------------------------------*  
  
using UnityEngine;  
using System.Collections;  
using UnityEngine.UI;  
  
public class HUDController : MonoBehaviour {  
  
    [SerializeField]  
    Text points = null;  
  
    [SerializeField]  
    Text lives = null;  
  
    [SerializeField]  
    GameObject player = null;  
  
    [SerializeField]  
    GameObject enemy1 = null;  
  
    [SerializeField]  
    GameObject enemy2 = null;  
  
    [SerializeField]  
    GameObject enemy3 = null;  
  
    [SerializeField]  
    GameObject powerUp = null;  
  
    [SerializeField]  
    GameObject shield = null;  
  
    [SerializeField]  
    Text gameover = null;  
  
    [SerializeField]  
    Text highscore = null;  
  
    [SerializeField]  
    Text gametitle = null;  
  
    [SerializeField]  
    Text instruct = null;  
  
    [SerializeField]  
    Text playerScore = null;  
  
    [SerializeField]  
    Button resetButton = null;  
              
    private int pointsLife = 1000;  
    private int pointsShield = 500;  
  
    void initiliaze(){  
        *//initialize the game*  
        Player.Instance.hud = this;  
        points.gameObject.SetActive (false);  
        lives.gameObject.SetActive (false);  
        player.SetActive (false);  
        enemy1.SetActive (false);  
        enemy2.SetActive (false);  
        enemy3.SetActive (false);  
        playerScore.gameObject.SetActive (false);  
        instruct.gameObject.SetActive (true);  
        gametitle.gameObject.SetActive (true);  
        gameover.gameObject.SetActive (false);  
        shield.gameObject.SetActive (false);  
        powerUp.gameObject.SetActive (false);  
        highscore.text = "High Score: " + Player.Instance.HighScore;  
        highscore.gameObject.SetActive (true);  
        resetButton.gameObject.SetActive (true);  
    }  
  
    void Start () {  
        initiliaze ();  
    }  
  
          
    void Update () {  
    }  
  
    public void updateLives(){  
        *//this updates player lives and points*  
        lives.text = "Lives: " + Player.Instance.Lives;  
    }  
  
    public void updatePoints(){  
        *//this update player points*  
        points.text = "Points: " + Player.Instance.Points;  
        *//add 1 life for every 1000 score*  
        if (Player.Instance.Points == pointsLife) {  
            Player.Instance.Lives += 1;  
            pointsLife += 1000;  
        }  
        if (Player.Instance.Points == pointsShield) {  
            powerUp.gameObject.SetActive (true);  
            pointsShield += 500;  
        }  
    }  
  
    public void GameOver(){  
        points.gameObject.SetActive (false);  
        lives.gameObject.SetActive (false);  
        player.SetActive (false);  
        enemy1.SetActive (false);  
        enemy2.SetActive (false);  
        enemy3.SetActive (false);  
        gametitle.gameObject.SetActive (true);  
        gameover.gameObject.SetActive (true);  
        powerUp.gameObject.SetActive (false);  
        playerScore.text = "Score: " + Player.Instance.Points;  
        highscore.text = "High Score: " + Player.Instance.HighScore;  
        highscore.gameObject.SetActive (true);  
        playerScore.gameObject.SetActive (true);  
        resetButton.gameObject.SetActive (true);  
        Player.Instance.Lives = 3;  
        Player.Instance.Points = 0;  
  
    }  
  
    public void ResetGame(){  
        points.gameObject.SetActive (true);  
        lives.gameObject.SetActive (true);  
        player.SetActive (true);  
        enemy1.SetActive (true);  
        enemy2.SetActive (true);  
        enemy3.SetActive (true);  
        instruct.gameObject.SetActive (false);  
        gametitle.gameObject.SetActive (false);  
        gameover.gameObject.SetActive (false);  
        powerUp.gameObject.SetActive (false);  
        playerScore.gameObject.SetActive (false);  
        highscore.gameObject.SetActive (false);  
        resetButton.gameObject.SetActive (false);  
        Player.Instance.Lives = 3;  
        Player.Instance.Points = 0;  
    }  
}

*///----------------------------------------------------------------------------------*  
*///   Source File Name: ShieldCollideController.cs*  
*///   Author's Name:  Roderick Rodelas*  
*///   Student Number: 100978575*  
*///   Program Description: This controls the player's movement.*  
*///   Date: October 28, 2016*  
*///   Last Modified by: Roderick Rodelas*  
*///   Date last modified:*  
*///   Revision History:*  
*///----------------------------------------------------------------------------------*  
  
using UnityEngine;  
using System.Collections;  
  
public class ShieldCollideController : MonoBehaviour {  
  
    private int shieldLife = 2;  
  
    *// Use this for initialization*  
    void Start () {  
      
    }  
      
    *// Update is called once per frame*  
    void Update () {  
      
    }  
  
    public void OnTriggerEnter2D(Collider2D other){  
        if (other.gameObject.tag == "enemyBullet") {  
            shieldLife -= 1;  
            Debug.Log ("Collision with " + other.gameObject.tag);  
            if (shieldLife == 0) {  
                Destroy (gameObject);}  
        } else if (other.gameObject.tag == "enemy") {  
            shieldLife -= 1;  
            Debug.Log ("Collision with " + other.gameObject.tag);  
            if (shieldLife == 0) {  
                Destroy (gameObject);}  
        }              
    }  
}

*///----------------------------------------------------------------------------------*  
*///   Source File Name: PowerUpController.cs*  
*///   Author's Name:  Roderick Rodelas*  
*///   Student Number: 100978575*  
*///   Program Description: This controls the player's movement.*  
*///   Date: October 28, 2016*  
*///   Last Modified by: Roderick Rodelas*  
*///   Date last modified:*  
*///   Revision History:*  
*///----------------------------------------------------------------------------------*  
  
using UnityEngine;  
using System.Collections;  
  
public class PowerUpController : MonoBehaviour {  
  
  
    [SerializeField]  
    private GameObject shield;  
  
    [SerializeField]  
    private float speed;  
  
  
    private Vector2 \_currentPosition;  
    private Transform \_transform = null;  
  
    *//Constants*  
    private const float startPosition = 1.75f;  
    private const float resetPosition = -20f;  
    private const float minY = -6f;  
    private const float maxY = 6f;  
  
    *// Use this for initialization*  
    void Start () {  
        \_transform = gameObject.transform;  
        \_currentPosition = \_transform.position;  
        shield.gameObject.SetActive (false);  
    }  
  
    void FixedUpdate () {  
  
  
        \_currentPosition = \_transform.position;  
  
        *//move the powerup with the given speed*  
        \_currentPosition -= new Vector2(speed, 0);  
        \_transform.position = \_currentPosition;  
  
        *//if powerup goes outside the screen*  
        if (\_transform.position.x < resetPosition) {  
            Destroy (gameObject);  
        }  
  
    }  
          
  
    public void OnTriggerEnter2D(Collider2D other){  
        if (other.gameObject.tag == "player")   
        {  
            shield.gameObject.SetActive (true);  
            Debug.Log ("Collision with " + other.gameObject.tag);  
            Destroy (gameObject);  
        }  
  
    }          
}