|  |
| --- |
| using System.Collections; |
|  | using System.Collections.Generic; |
|  | using UnityEngine; |
|  | using UnityEngine.UI; |
|  |  |
|  | public class RadialLoadingBar : MonoBehaviour { |
|  |  |
|  | public Transform LoadingBar; |
|  | public Transform Percent; |
|  | public Transform ZonePath; |
|  | public Transform PathTotal; |
|  | public Zone Zone; |
|  |  |
|  | [SerializeField] float currentAmount; |
|  | [SerializeField] float speed; |
|  |  |
|  | // Use this for initialization |
|  | void Start () |
|  | { |
|  | //Set Zone to the zone object in the scene |
|  | Zone = GameObject.Find("Zone").GetComponent<ZoneHandler>().CurrentZone; |
|  | } |
|  |  |
|  | // Update is called once per frame |
|  | void Update () |
|  | { |
|  | //Update the path graphic to show progress, current zone and path. |
|  | currentAmount = Zone.findPercent(); |
|  | if (currentAmount < 100) |
|  | { |
|  | Percent.GetComponent<Text>().text = currentAmount.ToString("f1") + "%"; |
|  | ZonePath.GetComponent<Text>().text = Zone.ID + " - " + Zone.Path; |
|  | PathTotal.GetComponent<Text>().text = Zone.StepsUntilNextPath.ToString("f0"); |
|  | } |
|  | else |
|  | { |
|  | Percent.GetComponent<Text>().text = "100%"; |
|  | } |
|  |  |
|  | LoadingBar.GetComponent<Image>().fillAmount = currentAmount / 100; |
|  | } |
|  | } |

|  |
| --- |
| using System.Collections; |
|  | using System.Collections.Generic; |
|  | using UnityEngine; |
|  | using System; |
|  |  |
|  | public class Step : MonoBehaviour |
|  | { |
|  | public UnityEngine.UI.Text stepDisplay; |
|  | public UnityEngine.UI.Text currencyDisplay; |
|  | public static int steps = 0; |
|  | public static int currency = 0; |
|  | private bool encounterActive = false; |
|  | private PedometerPlugin pedometer; |
|  | private SensorDelay sensor; |
|  |  |
|  | void Start() |
|  | { |
|  | //Gets instance |
|  | pedometer = PedometerPlugin.GetInstance(); |
|  |  |
|  | //Hides debug messages when set to zero |
|  | pedometer.SetDebug(0); |
|  |  |
|  | //Initaialize |
|  | pedometer.Init(); |
|  |  |
|  | pedometer.StartPedometerService(sensor); |
|  | } |
|  |  |
|  | void Update() |
|  | { |
|  | //If the player didn't run into an encounter, update the step text |
|  | if (!encounterActive) |
|  | { |
|  | pedometer.LoadStepToday(); |
|  | int newSteps = pedometer.GetStepToday(); |
|  | if (steps < newSteps) |
|  | steps += (newSteps - steps); |
|  | stepDisplay.text = "" + steps; |
|  | currencyDisplay.text = "Currency: " + currency; |
|  |  |
|  | } |
|  | } |
|  |  |
|  | //Function for step button, to be replaced with actual walking at a later date. |
|  | public void Clicked() |
|  | { |
|  | encounterActive = Encounter(); |
|  |  |
|  | if (!encounterActive) |
|  | { |
|  | //Add step to the total step counter. |
|  | steps += 1; |
|  |  |
|  | //Add step to the Zone and Path counter that's stored in another object |
|  | GameObject.Find("Zone").GetComponent<ZoneHandler>().CurrentZone.addStep(); |
|  |  |
|  | //Every 50 steps will add one currency. |
|  | if (steps % 50 == 0) |
|  | currency += 1; |
|  | } |
|  | else |
|  | //Temporary. To be replaced with a function that goes to encounter screen |
|  | stepDisplay.text = "ENCOUNTER!"; |
|  |  |
|  | } |
|  |  |
|  | public bool Encounter() |
|  | { |
|  | //The player as a 1/100 chance of encountering an enemy every step they take. |
|  | if (Mathf.Ceil(UnityEngine.Random.Range(0.0f, 100.0f)) == 1) |
|  | return true; |
|  | else |
|  | return false; |
|  | } |
|  |  |
|  | } |

|  |
| --- |
| using System.Collections; |
|  | using System.Collections.Generic; |
|  | using UnityEngine; |
|  |  |
|  | public class StepScene : MonoBehaviour |
|  | { |
|  | private CanvasGroup fadeGroup; |
|  | private float fadeInSpeed = 0.75f; |
|  |  |
|  | private void Start() |
|  | { |
|  | //get the fading graphic |
|  | fadeGroup = FindObjectOfType<CanvasGroup>(); |
|  | fadeGroup.alpha = 1.0f; |
|  | } |
|  |  |
|  | private void Update() |
|  | { |
|  | //fade in |
|  | fadeGroup.alpha = 1 - Time.timeSinceLevelLoad \* fadeInSpeed; |
|  | } |
|  | } |

|  |
| --- |
| using System.Collections; |
|  | using System.Collections.Generic; |
|  | using UnityEngine; |
|  | using UnityEngine.UI; |
|  | using UnityEngine.EventSystems; |
|  | using System; |
|  |  |
|  | public class ZoneDisplay : MonoBehaviour { |
|  |  |
|  | public List<Zone> zones; |
|  |  |
|  | public GameObject ZoneIcon; |
|  | GameObject mainPanel; |
|  | GameObject zonePanel; |
|  |  |
|  | public List<GameObject> zoneSlots = new List<GameObject>(); |
|  |  |
|  | // Use this for initialization |
|  | void Start () |
|  | { |
|  | //grab the zone object in the scene |
|  | zones = GameObject.Find("Zone").GetComponent<ZoneHandler>().Zones; |
|  |  |
|  | //grab the panel objects in the scene |
|  | mainPanel = GameObject.Find("MainPanel"); |
|  | zonePanel = mainPanel.transform.Find("ZonePanel").gameObject; |
|  |  |
|  | for (int i = 0; i < zones.Count; i++) |
|  | { |
|  | zoneSlots.Add(Instantiate(ZoneIcon)); |
|  |  |
|  | //Get the gameobjects to adjust values based on the zone. |
|  | GameObject center = zoneSlots[i].transform.Find("Center").gameObject; |
|  | GameObject percentText = center.transform.Find("txt\_percent").gameObject; |
|  | GameObject zonePathText = zoneSlots[i].transform.Find("ZonePath").gameObject; |
|  | GameObject untilNextPathText = center.transform.Find("txt\_pathTotal").gameObject; |
|  | GameObject loadingBar = zoneSlots[i].transform.Find("LoadingBar").gameObject; |
|  | GameObject selector = zoneSlots[i].transform.Find("Selector").gameObject; |
|  |  |
|  | zoneSlots[i].transform.SetParent(zonePanel.transform); |
|  | zoneSlots[i].GetComponent<ZoneSlotInfo>().id = i; |
|  | //Get the center picture right |
|  | center.GetComponent<Image>().sprite = zones[i].Sprite; |
|  |  |
|  | //Get the progress bar |
|  | loadingBar.GetComponent<Image>().fillAmount = zones[i].findPercent() / 100.0f; |
|  |  |
|  | //Get Percent text |
|  | percentText.GetComponent<Text>().text = zones[i].findPercent().ToString("F1") + "%"; |
|  |  |
|  | //Get the steps until next path text |
|  | untilNextPathText.GetComponent<Text>().text = zones[i].StepsUntilNextPath.ToString(); |
|  |  |
|  | //Get text for the Zone - Path |
|  | zonePathText.GetComponent<Text>().text = zones[i].ID + " - " + zones[i].Path; |
|  |  |
|  | //Is object selected? Activate Selector |
|  | if (zones[i] == GameObject.Find("Zone").GetComponent<ZoneHandler>().CurrentZone) |
|  | selector.SetActive(true); |
|  | else |
|  | selector.SetActive(false); |
|  |  |
|  | } |
|  | } |
|  |  |
|  | // Update is called once per frame |
|  | void Update () |
|  | { |
|  | //Search through all the zones |
|  | for (int i = 0; i < zones.Count; i++) |
|  | { |
|  | //Find the selector, selector highlights the zone you're currently in. |
|  | GameObject selector = zoneSlots[i].transform.Find("Selector").gameObject; |
|  |  |
|  | //Is zone selected? Activate Selector |
|  | if (zones[i] == GameObject.Find("Zone").GetComponent<ZoneHandler>().CurrentZone) |
|  | selector.SetActive(true); |
|  | else |
|  | selector.SetActive(false); |
|  | } |
|  |  |
|  | } |
|  | } |

|  |
| --- |
| /\* Inventory.cs |
|  | \* Phillip Buckreis 11/4/2017 |
|  | \* |
|  | \* This creates the inventory system. |
|  | \*/ |
|  |  |
|  | using System.Collections; |
|  | using System.Collections.Generic; |
|  | using UnityEngine; |
|  | using UnityEngine.UI; |
|  |  |
|  | public class ZoneHandler : MonoBehaviour |
|  | { |
|  | public Zone CurrentZone; |
|  | public List<Zone> Zones = new List<Zone>(); |
|  |  |
|  | private void Awake() |
|  | { |
|  | //When we create the object, make sure it isn't destroyed when we switch scenes. |
|  | DontDestroyOnLoad(gameObject); |
|  | } |
|  |  |
|  | void Start() |
|  | { |
|  | //Create zones, to be replaced with a database and a loop |
|  | Zone tempZone; |
|  |  |
|  | tempZone= new Zone(1, 100.0f, 1.15f, "forest\_zone", true); |
|  | Zones.Add(tempZone); |
|  |  |
|  | tempZone = new Zone(2, 150.0f, 1.14f, "graveyard\_zone", false); |
|  | Zones.Add(tempZone); |
|  |  |
|  | //Starting zone |
|  | changeZone(0); |
|  | } |
|  |  |
|  | public void changeZone(int changeTo) |
|  | { |
|  | CurrentZone = Zones[changeTo]; |
|  | } |
|  | } |
|  |  |
|  | public class Zone |
|  | { |
|  | //simple set/get functions |
|  | public int ID { get; set; } |
|  | public int Path { get; set; } |
|  | public float InitialStepRequirement { get; set; } |
|  | public float StepsInCurrentPath { get; set; } |
|  | public float StepsUntilNextPath { get; set; } |
|  | public float Growth { get; set; } |
|  | public string Slug { get; set; } |
|  | public bool Unlocked { get; set; } |
|  | public Sprite Sprite { get; set; } |
|  |  |
|  | //default constructor, give strange id |
|  | public Zone() |
|  | { |
|  | this.ID = -1; |
|  | this.Path = 1; |
|  | this.StepsInCurrentPath = 0; |
|  | } |
|  |  |
|  | //constructor |
|  | public Zone(int id, float initialStepRequirement, float growth, string slug, bool unlocked) |
|  | { |
|  | this.ID = id; |
|  | this.Path = 1; |
|  | this.InitialStepRequirement = initialStepRequirement; |
|  | this.StepsInCurrentPath = 0; |
|  | this.StepsUntilNextPath = initialStepRequirement; |
|  | this.Growth = growth; |
|  | this.Slug = slug; |
|  | this.Unlocked = unlocked; |
|  | this.Sprite = Resources.Load<Sprite>("Sprites/Zone Icons/" + slug); |
|  | } |
|  |  |
|  | public void addStep() |
|  | { |
|  | StepsInCurrentPath++; |
|  | //If you reach the end of the path, increase the path by one, reset current steps in path. |
|  | if (StepsInCurrentPath > StepsUntilNextPath) |
|  | { |
|  | StepsInCurrentPath = 0; |
|  | Path++; |
|  |  |
|  | //Growth is expotential, beginning zones have a low initial step requirement, but faster growth, |
|  | //while later zones have high initial step requirements, but slower growth |
|  | StepsUntilNextPath = Mathf.Ceil(InitialStepRequirement \* Mathf.Pow(Growth, Path - 1)); |
|  | } |
|  | } |
|  |  |
|  | public float findPercent() |
|  | { |
|  | return (StepsInCurrentPath / StepsUntilNextPath) \* 100; |
|  | } |
|  |  |
|  | } |