smartAInnovators: Προωθώντας τον ψηφιακό Μετασχηματισμό και την καινοτομία με γνώμονα την Τεχνητή Νοημοσύνη σε Κέντρα Επαγγελματικής Εκπαίδευσης (ΚΕΕ) για κοινωνική αλλάγη και καλύτερη σύνδεση Δεξιοτήτων και Αγοράς Εργασίας

Αριθμός Έργου: 2021-1-PL01-KA220-VET-000033011

smart AInnovators

ΕΚΠΑΙΔΕΥΤΙΚΟ ΠΑΚΕΤΟ: ΔΙΔΑΚΤΙΚΟ ΥΛΙΚΟ ΚΑΙ ΑΞΙΟΛΟΓΗΣΗ

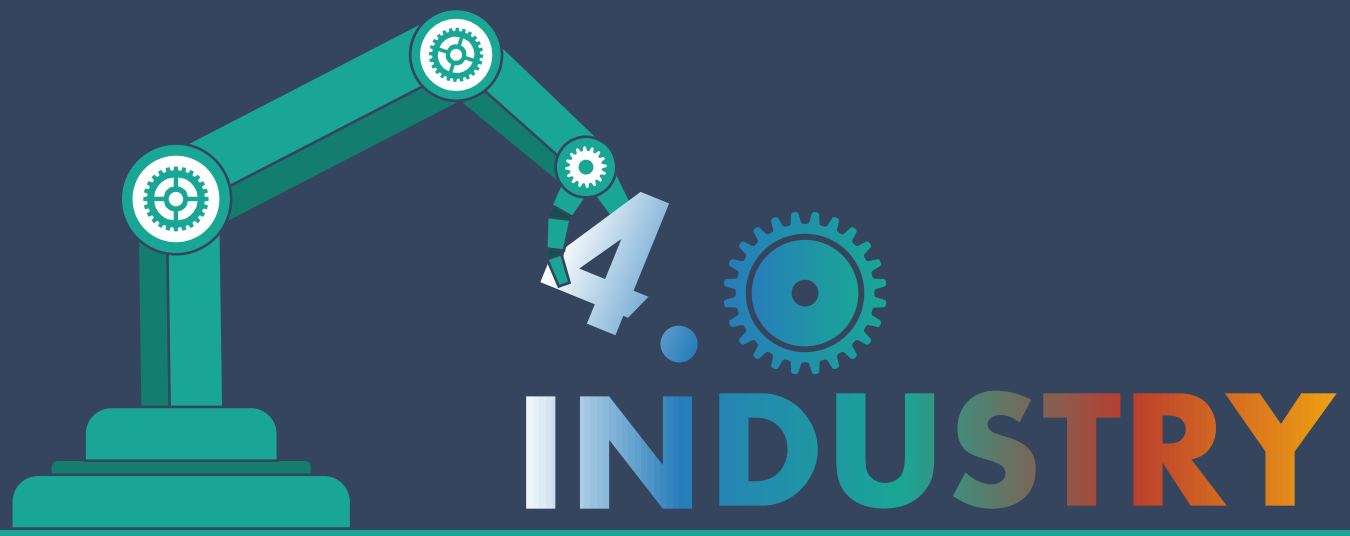
Co-funded by the European Union

Το έργο αυτό χρηματοδοτήθηκε με την υποστήριξη της Ευρωπαϊκής Επιτροπής. Το παρόν αντικατοπτρίζει αποκλειστικά τις απόψεις του συντάκτη και η Επιτροπή δεν μπορεί να θεωρηθεί υπεύθυνη για τυχόν χρήση των πληροφοριών που περιέχονται σε αυτό.

Αυτό το έγγραφο μπορεί να αντιγραφεί, να αναπαραχθεί ή να τροποποιηθεί σύμφωνα με τους παραπάνω κανόνες. Επιπλέον, πρέπει να αναφέρεται σαφώς η αναγνώριση των συντακτών του εγγράφου και όλων των ισχυόντων τμημάτων της ειδοποίησης πνευματικών δικαιωμάτων. Ολα τα δικαιώματα διατηρούνται. © Πνευματικά δικαιώματα 2021 smartAInnovators

CC BY NC SA

smart AInnovators



UNITY Game Development – ‘Τυχερό Παιχνίδι’

Οργανισμός: ΔΤΕΡΜΟΝ EDUCATIONAL PLAY

Κοινοπραξία

smart AInnovators

Coordinator

Społeczna Akademia Nauk

Emphasys Centre

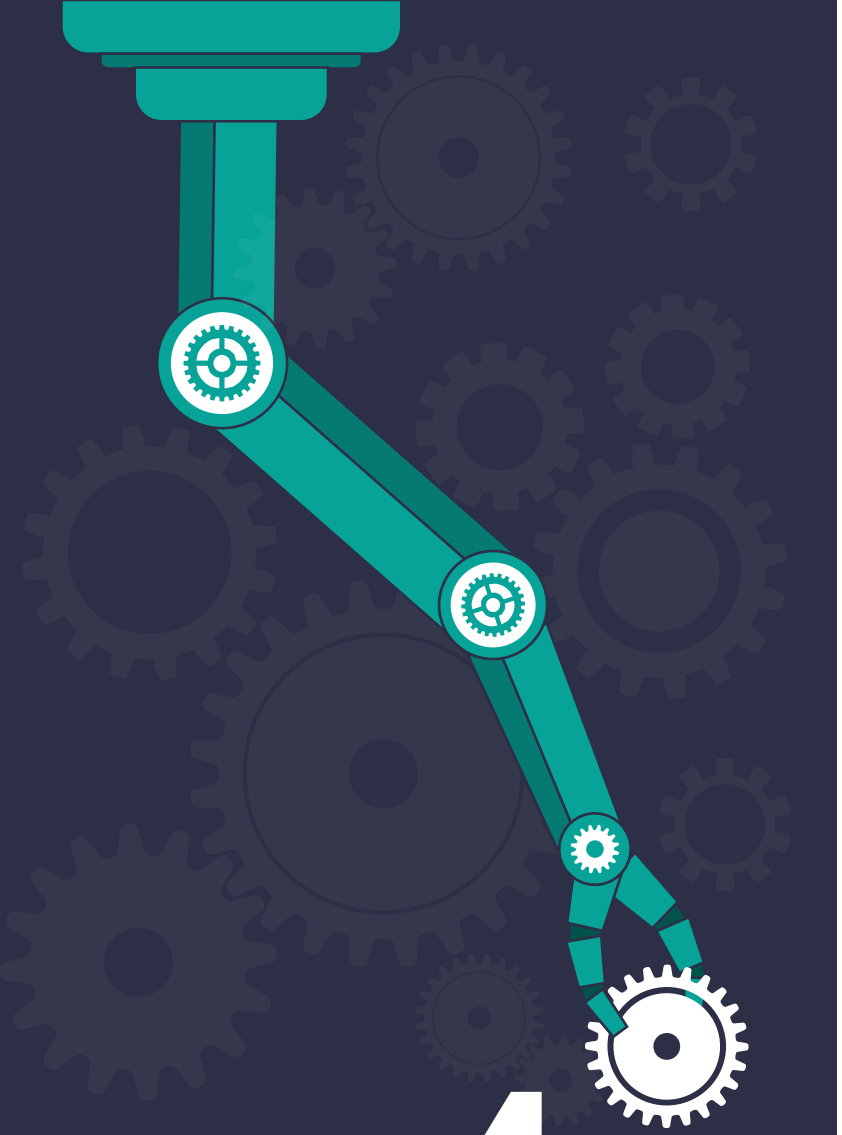
Atlantis Engineering

Politeknika Txorierri

PCG Polska
A Public Consulting Group, Inc. Company

ΔTERMON Educational Play

Information Technology School



4.
INDUSTRY

smart
AInnovators

Περιεχόμενα

01 **Θέμα 1**

Ρυθμίστε το περιβάλλον Unity δημιουργώντας το περιβάλλον εργασίας χρήστη και προσθέτοντας κινούμενα σχέδια

02 **Θέμα 2**

Αναπτύξτε ένα «τυχερό παιχνίδι» στο Unity δημιουργώντας τη λειτουργικότητα του τροχού και δημιουργώντας τη λογική του παιχνιδιού

03 **Θέμα 3**

Δοκιμάστε το «τυχερό παιχνίδι» και δείτε τα ποσοστά των πιθανοτήτων σε κάθε αντικείμενο

04 **Θέμα 4**

Αλλάξτε το ποσοστό των πιθανοτήτων τυχαία ή χειροκίνητα

Εισαγωγή

UNITY Game Development – «Lucky Game»

Τι αφορά αυτή η ενότητα

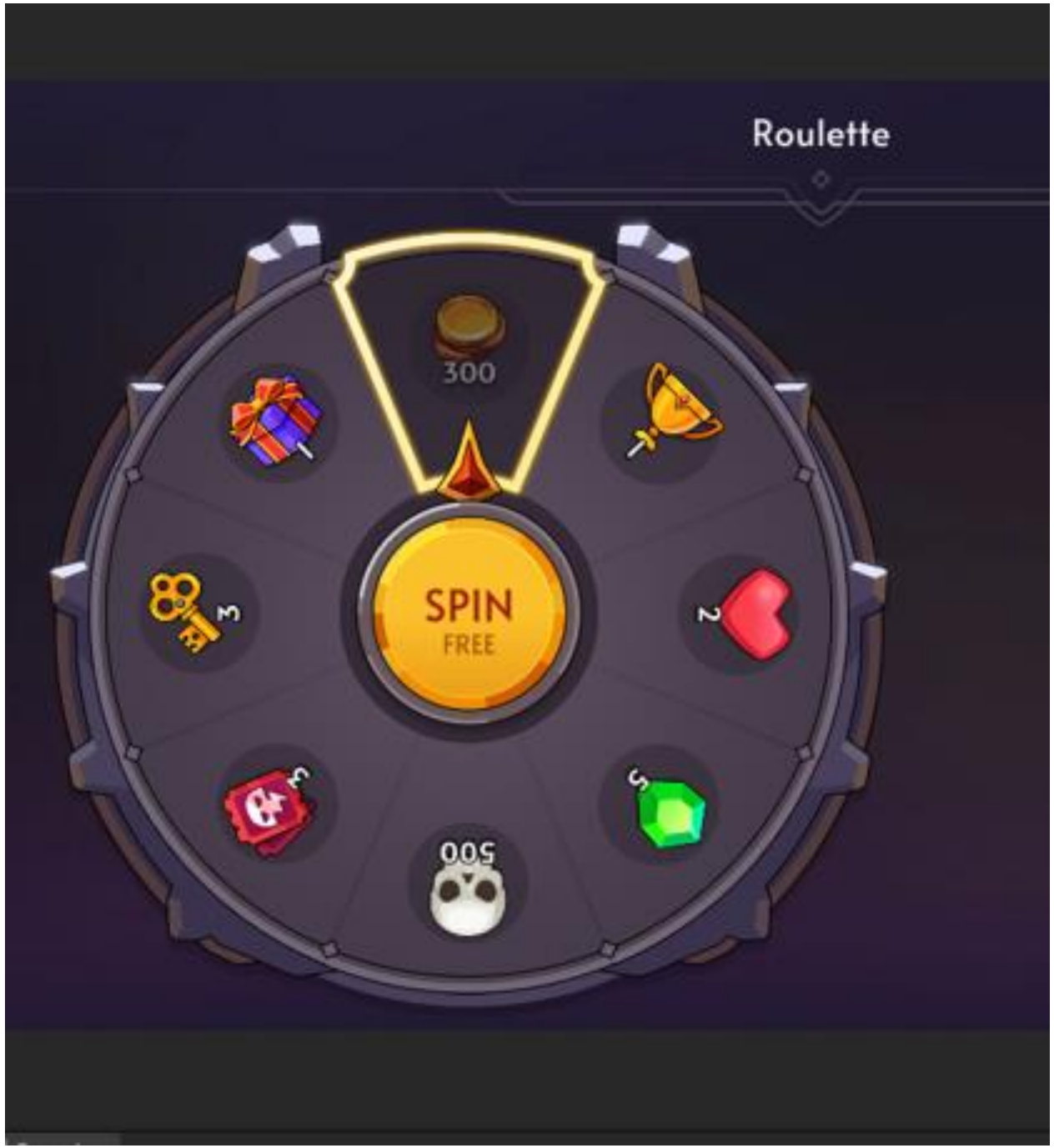
Αυτή η ενότητα περιλαμβάνει 4 βασικά θέματα:

1. Ρυθμίστε το περιβάλλον Unity δημιουργώντας τη διεπαφή χρήστη και προσθέτοντας κινούμενα σχέδια

2. Αναπτύξτε ένα «τυχερό παιχνίδι» στο Unity δημιουργώντας τη λειτουργικότητα του τροχού και δημιουργώντας τη λογική του παιχνιδιού

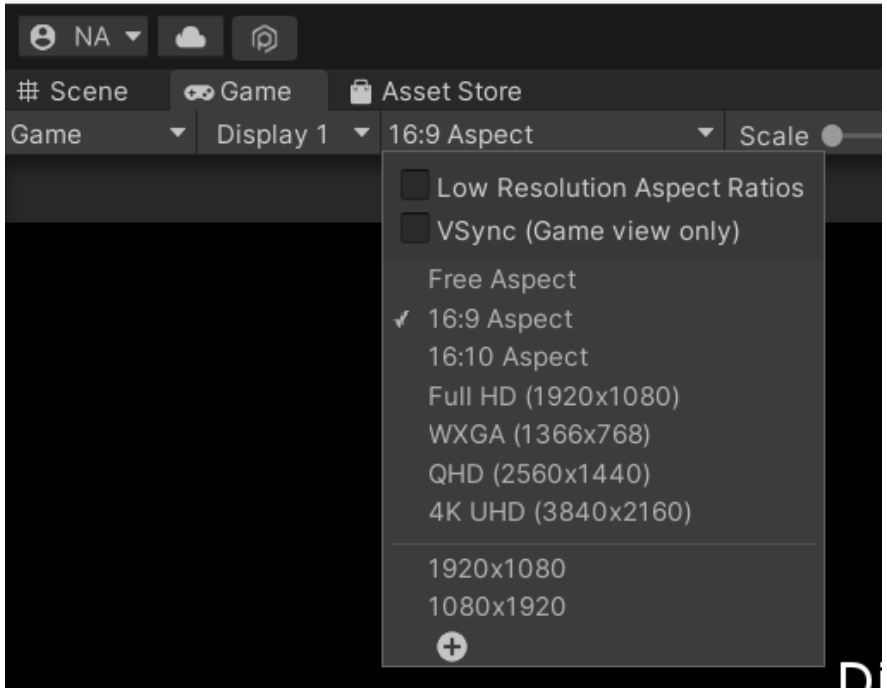
3. Δοκιμάστε το «τυχερό παιχνίδι» και δείτε τα ποσοστά των πιθανοτήτων σε κάθε αντικείμενο

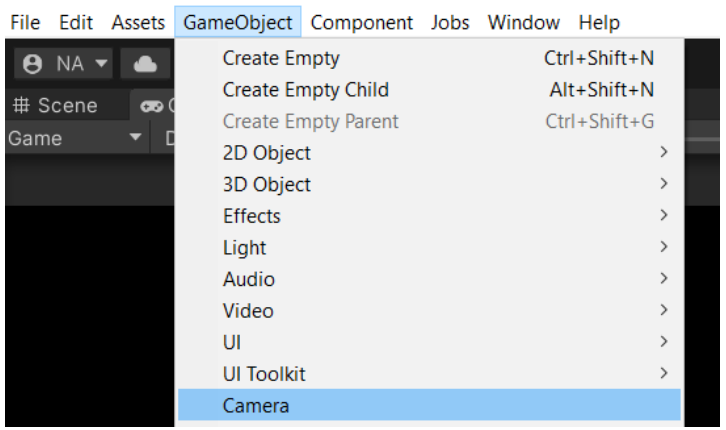
4. Αλλάξτε το ποσοστό των πιθανοτήτων τυχαία ή χειροκίνητα



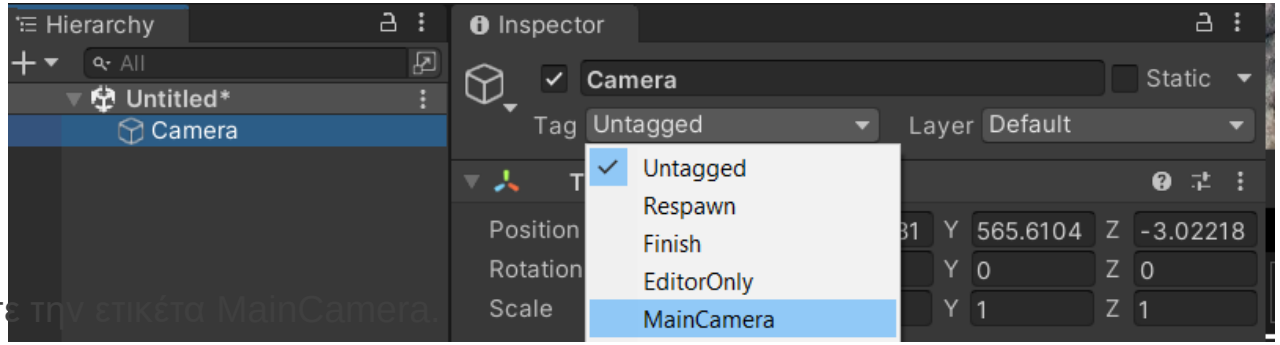
Setup Unity Project

- Ανοίξτε το Unity Hub σας.
- Επιλέξτε νέο για να δημιουργήσετε ένα νέο έργο. Σε αυτό το σημείο, μπορείτε να επιλέξετε από μια σειρά διαφορετικών προτύπων για το έργο Unity σας.
- Επιλέξτε Πρότυπο 2D, ονομάστε αυτό το έργο "LuckyWheel_{το όνομά σας}" και επιλέξτε δημιουργία.
- Στην επάνω δεξιά γωνία του παραθύρου Unity, θα βρείτε ένα αναπτυσσόμενο μενού με την ένδειξη Layout. Επιλέξτε τη διάταξη που επιθυμείτε.
- Δημιουργία νέας σκηνής Αρχείο > Νέα σκηνή. Πριν το αποθηκεύσετε, δημιουργήστε έναν νέο κατάλογο μέσα στο φάκελο Assets με το όνομα "Scene".
- Αποθηκεύστε τη νέα σκηνή Αρχείο > Αποθηκεύστε το με το όνομα "Παιχνίδι".
- Εντοπίστε και εισαγάγετε το ProjectLuckyWheelAssets.unitypackage στο Assets-> Import Package -> CustomPackage
- Ρύθμιση σκηνής
- Ρυθμίστε την ανάλυση παιχνιδιού του Editor σε 1920x1080 ή ρυθμίστε την σε αναλογία διαστάσεων 16:9



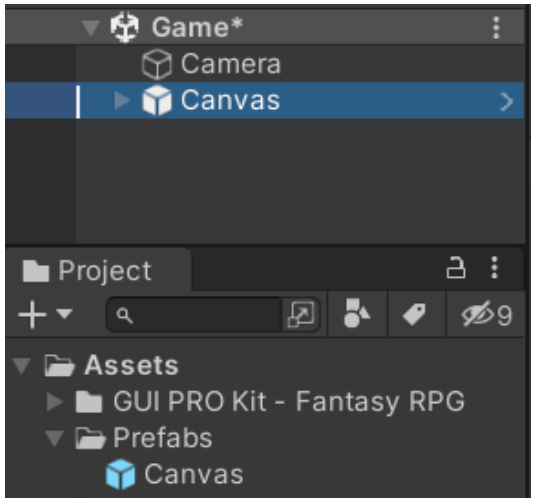


- Μεταβείτε στο μενού Gameobject->Camera για να δημιουργήσετε μια κάμερα.



- Ορίστε την ετικέτα Camera σε MainCamera.
- Επιλέξτε Camera GameObject.
- Μεταβείτε στο παράθυρο του επιθεωρητή.
- Κάντε κλικ στο αναπτυσσόμενο μενού ετικέτας και επιλέξτε την ετικέτα MainCamera.

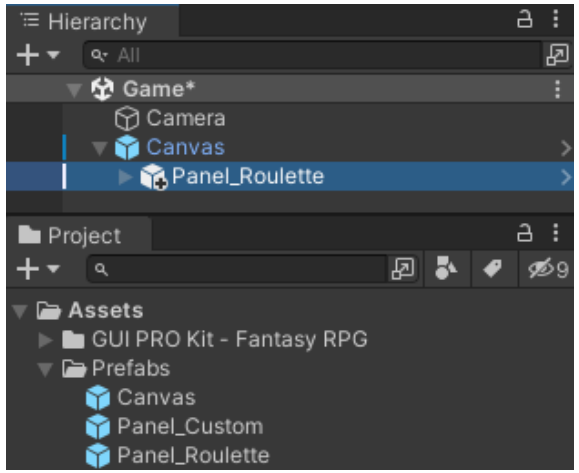
- Ελέγξτε το παράθυρο Project και αναπτύξτε το φάκελο Prefabs.

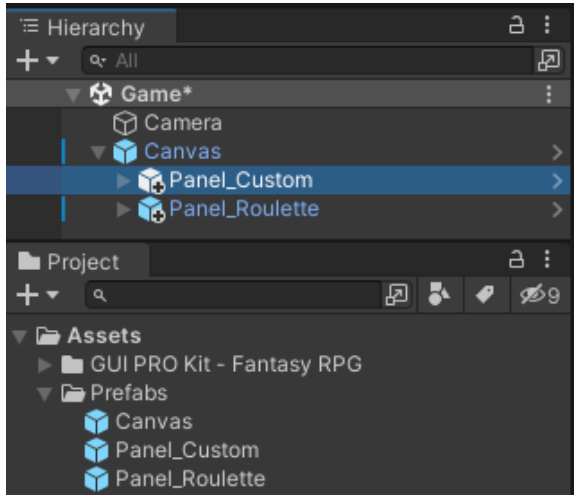


- Σύρετε και αποθέστε το προκατασκευασμένο Canvas στο παράθυρο Ιεραρχία.
- Πατήστε Ctrl + S για να αποθηκεύσετε τη σκηνή.

Ρύθμιση προκατασκευασμένου καμβά

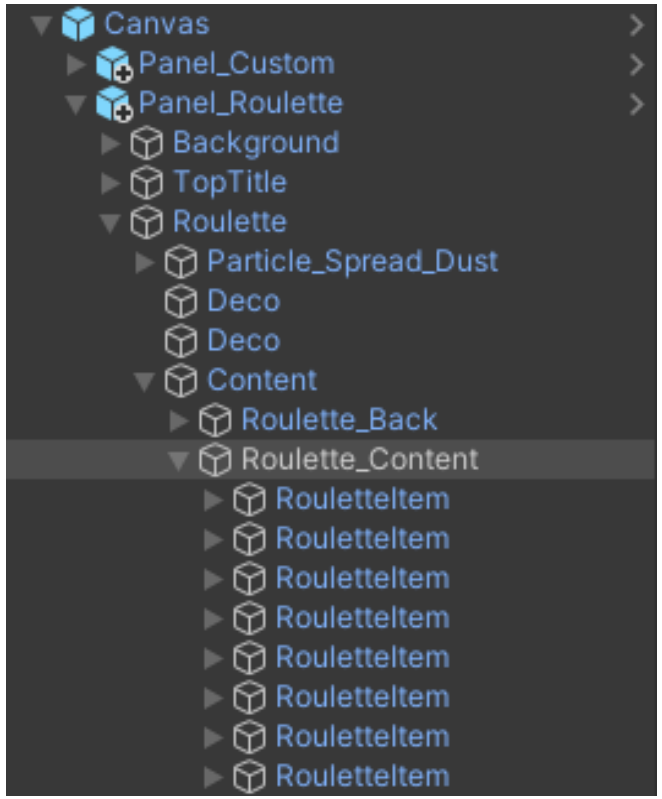
- Σύρετε και αποθέστε το προκατασκευασμένο Canvas στο παράθυρο Ιεραρχία.
- Σύρετε και αποθέστε το προκατασκευασμένο Panel_Roulette στο παράθυρο Ιεραρχίας μέσα στο Προκατασκευασμένο Καμβά για να το ορίσετε ως το παιδί του Canvas

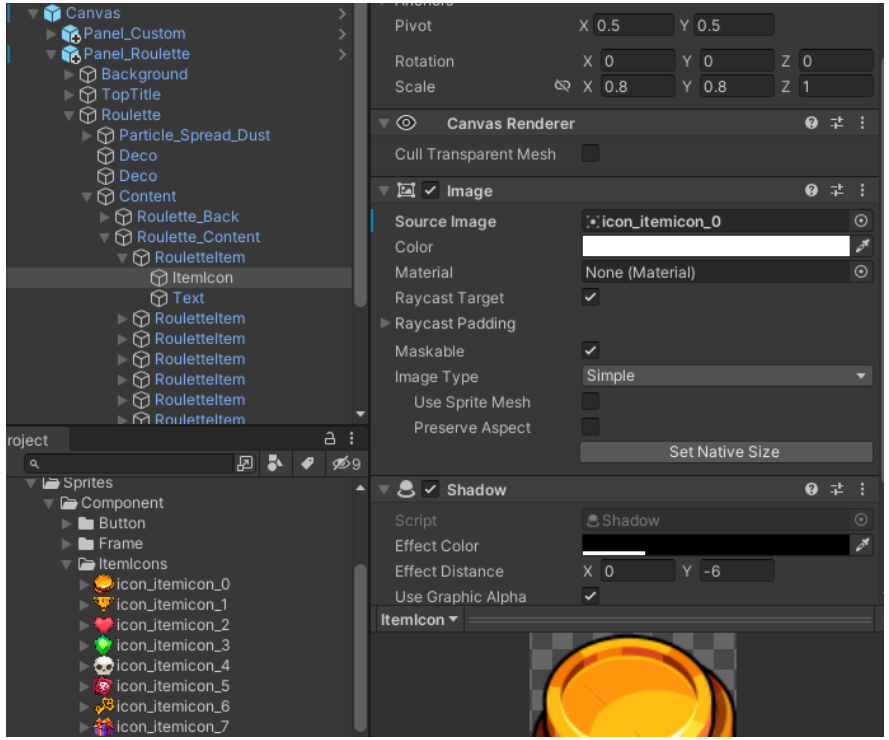




- Σύρετε και αποθέστε το Panel_Custom prefab στο παράθυρο Hierarchy μέσα στο Canvas Prefab για να το ορίσετε ως το παιδί του Canvas

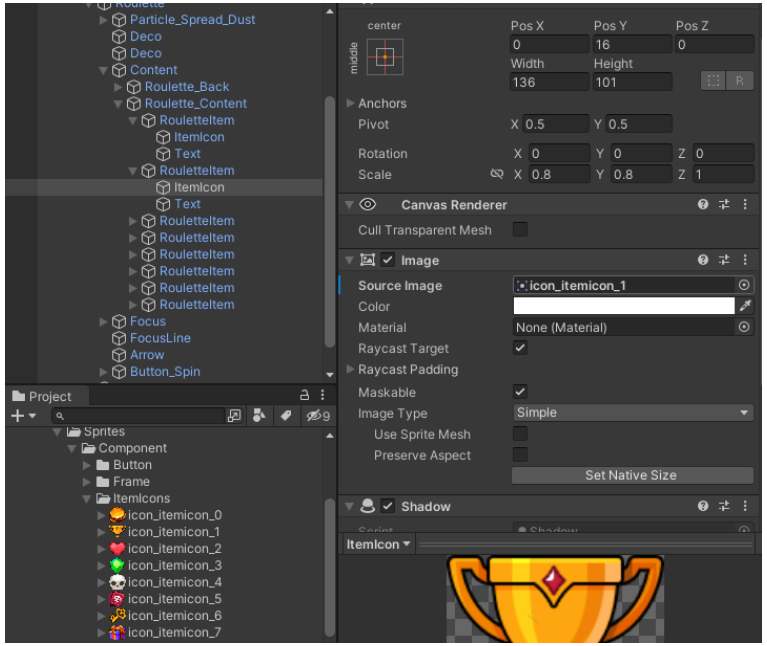
- Αναπτύξτε το Panel_Roulette κάνοντας κλικ στο αριστερό βέλος του GameObject.
- Αναπτύξτε το GameObject μέχρι το prefab να φτάσει στο Roulette_Content GameObject. Δείτε την παρακάτω εικόνα:

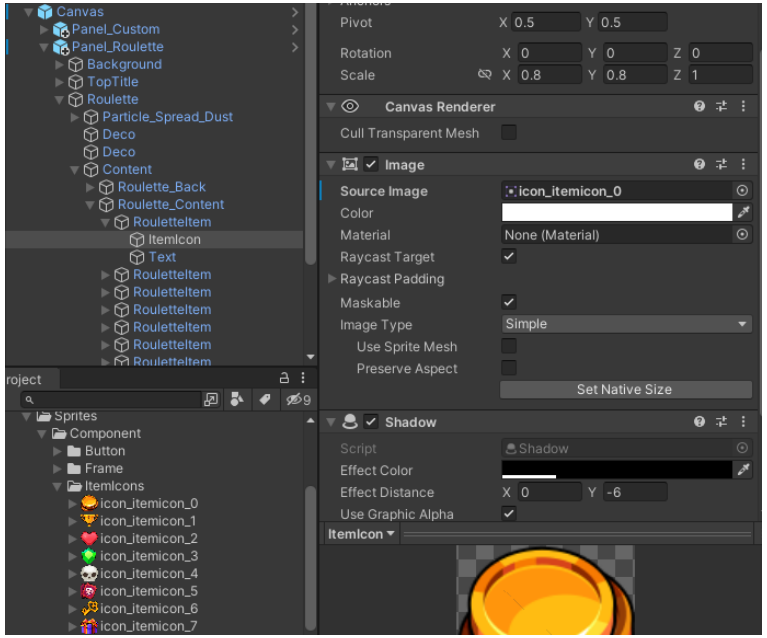




- Σύρετε και αποθέστε τα σωστά εικονίδια για κάθε sprite ItemIcon. Το Sprite μπορεί να βρεθεί στο GUI PRO Kit - Fantasy RPG->ResourcesData->Sprites->Component->ItemIcons.

- Σύρετε και αποθέστε τα σωστά εικονίδια για κάθε sprite ItemIcon. Το Sprite μπορεί να βρεθεί στο GUI PRO Kit - Fantasy RPG->ResourcesData->Sprites->Component->ItemIcons.

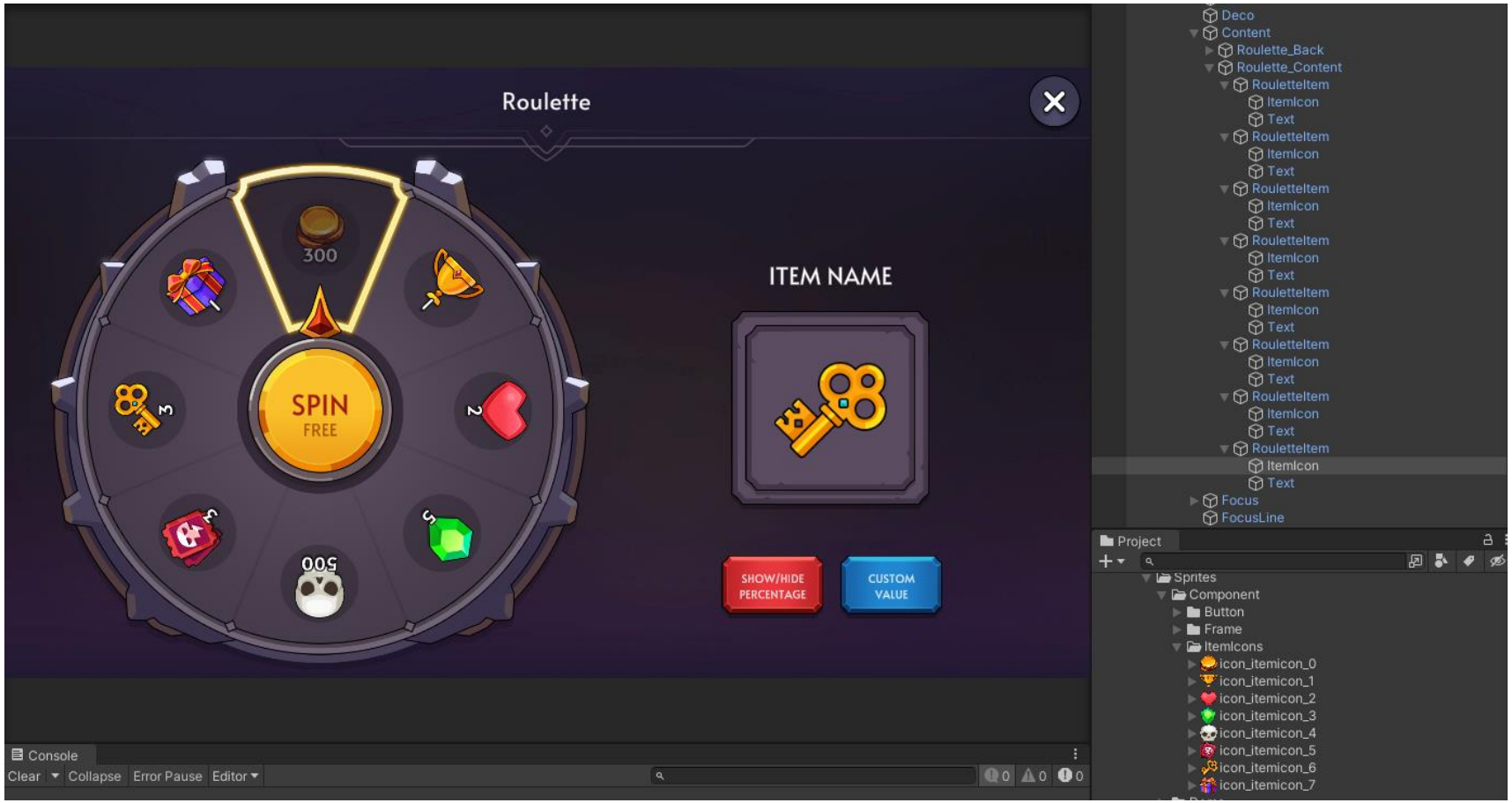




- Αναπτύξτε το RouletteItem GambObject και επιλέξτε το ItemIcon και μεταφέρετε και αποθέστε το sprite στο sourceImage.

smart
AInnovators

Το αποτέλεσμα θα πρέπει να είναι η παρακάτω εικόνα:



Ρύθμιση του αρχικού σεναρίου UIController

1. Κάντε δεξί κλικ στο Φάκελο Script Create->C# Script.
2. Ονομασία δέσμης ενεργειών στο UIController.
3. Επιλέξτε Canvas GameObject και Προσθήκη στοιχείου σεναρίου UIController.
4. Κάντε διπλό κλικ στη δέσμη ενεργειών UIController στο παράθυρο του έργου.
5. Ας γράψουμε το Σενάριο μας:
   A)Προσθέστε βιβλιοθήκες Unity και TextMeshPro
      i. Προσθέστε τον ακόλουθο κώδικα κάτω από τον κωδικό UnityEngine που χρησιμοποιεί:
         χρησιμοποιώντας το σύστημα
         χρησιμοποιώντας UnityEngine.Events
         χρησιμοποιώντας UnityEngine.UI
         χρησιμοποιώντας TMPro

smart AInnovators

b. Προσθήκη μεταβλητών σεναρίου

Προσθέστε τον ακόλουθο κώδικα πάνω από τη συνάρτηση Start():

```
[SerializeField] RectTransform _rectPanelRoulette;
[SerializeField] RectTransform _rectPanelCustom;

[SerializeField] RectTransform rectSpinner;
[SerializeField] RectTransform rectArrow;
[SerializeField] Button buttonSpin;
[SerializeField] Button buttonClose;
[SerializeField] Button buttonCustom;
[SerializeField] Button buttonShowHide;
[SerializeField] TextMeshProUGUI txtResult;
[SerializeField] Image imageIcon;
```

```
[SerializeField] GameObject particle;
[SerializeField] GameObject particleResult;
[SerializeField] TextMeshProUGUI[] textPercentages;

[SerializeField] TextMeshProUGUI textButtonShowHide;

[SerializeField] RectTransform _contentItem;
[SerializeField] RectTransform _contentItemInput;

[SerializeField] GameObject item;
[SerializeField] GameObject itemInput;

[SerializeField] Button buttonRandom;
[SerializeField] Button buttonSetValue;
[SerializeField] Button buttonBack;

bool toggle = true;
```

c. Προσθήκη συναρτήσεων σεναρίου

Προσθέστε τον ακόλουθο κώδικα κάτω από τη συνάρτηση Start():

```
public void AddButtonSetValueListener(UnityAction
callbackSet)
{
}
public void AddButtonRandomListener(UnityAction
callbackRandom)
{
 }
 public void AddButtonSpinListener(UnityAction
callbackSpin)
 {
 }
 public void ShowParticleResult(bool show)
 {
 }
 public void ShowParticleSelected(bool show)
 {
 }
 public void ShowTextResult(bool show)
 {
 }
 public void ShowUI(UIState state)
 {
 }
 public void SetSpinnerRotation(float rot)
 {
 }
 public void SetArrowRotation(float rot)
 {
 }
 public void SetIcon(Sprite sprite)
 {
 }
 public void SetResultText(string result)
 {
 }
 public void ShowItemPercentage(PrizeData[] data, bool
updateToggle = true)
 {
 }
 public void ShowContentItem(PrizeData[] data)
 {
 }
 public void ShowContentItemInput(PrizeData[] data)
 {
 }
 public void SetInputValueToData(PrizeData[] data)
 {
 }
 private void ToggleTextPercentage()
 {
 }
```

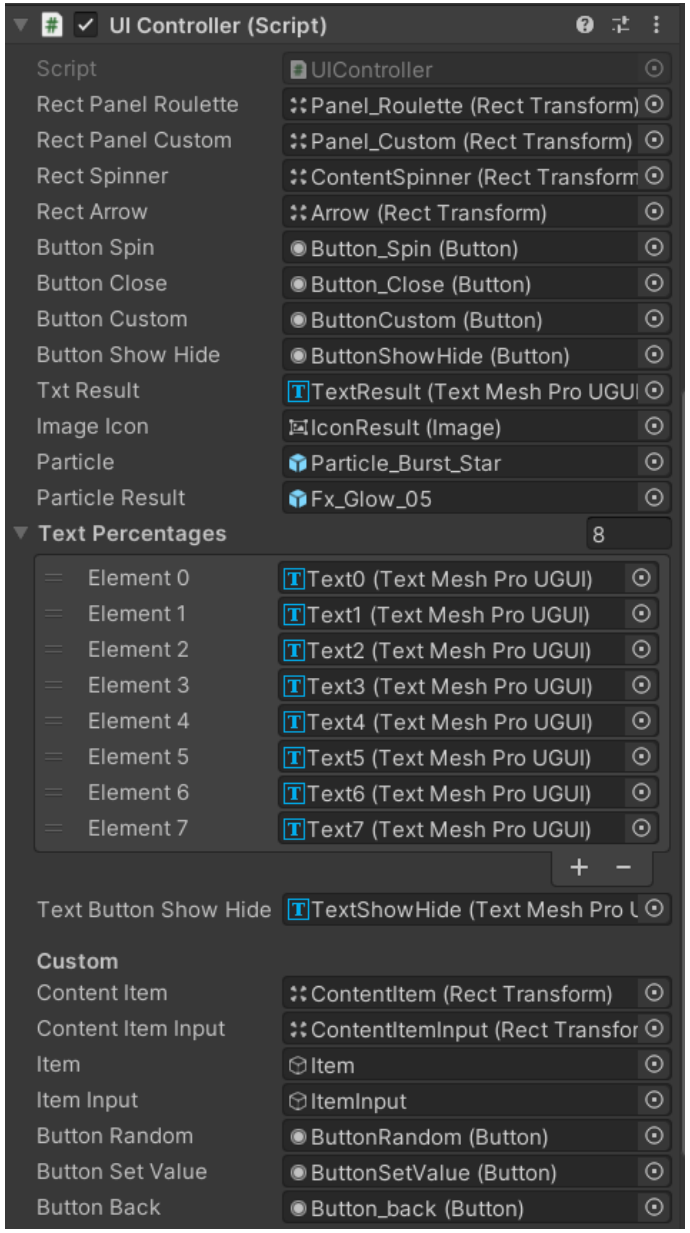
smart AInnovators

d. Προσθήκη νέας τάξης και Enum

Προσθέστε τον ακόλουθο κώδικα κάτω από την κατηγορία UIController:

```
[Serializable]
public class PrizeData
{
        public string name;
        public float percentage;
        public Sprite sprite;
}


public enum UIState
{
   ROULETTE,
   CUSTOM
}
```

Επιστρέψτε στο Unity Editor.
Επιλέξτε Canvas GameObject.
Σύρετε και αποθέστε στοιχεία Canvas στις αντίστοιχες μεταβλητές του Ελεγκτή διεπαφής χρήστη. Δείτε την παρακάτω εικόνα:



Ρύθμιση σεναρίου UIController (συνέχεια)

1. Κάντε διπλό κλικ στη δέσμη ενεργειών UIController στο παράθυρο του έργου.
2. Ας γράψουμε το Σενάριο μας:

Α)Γράψτε τον ακόλουθο κώδικα μέσα στη συνάρτηση Start():

```
buttonClose.onClick.AddListener(() => { Application.Quit(); });
buttonBack.onClick.AddListener(()=>
{
    ShowUI(UIState.ROULETTE);
});

buttonCustom.onClick.AddListener(()=>
{
    ShowUI(UIState.CUSTOM);
});

buttonShowHide.onClick.AddListener(()=>
{
    ToggleTextPercentage();
});
```

b. **Γράψτε τον ακόλουθο κώδικα μέσα στη συνάρτηση AddButtonSetValueListener:**

```
buttonSetValue.onClick.AddListener(callbackSet);
```

c. **Γράψτε τον ακόλουθο κώδικα μέσα στη συνάρτηση AddButtonRandomListener:**

```
buttonRandom.onClick.AddListener(callbackRandom);
```

d. **Γράψτε τον ακόλουθο κώδικα μέσα στη συνάρτηση AddButtonSpinListener:**

```
buttonSpin.onClick.AddListener(callbackSpin);
```

e. **Γράψτε τον ακόλουθο κώδικα μέσα στη συνάρτηση ShowParticleResult:**

```
particleResult.gameObject.SetActive(show);
```

f. **Γράψτε τον ακόλουθο κώδικα μέσα στη συνάρτηση ShowParticleSelected:**

```
particle.gameObject.SetActive(show);
```

g. **Γράψτε τον ακόλουθο κώδικα μέσα στη συνάρτηση ShowTextResult:**

```
txtResult.gameObject.SetActive(show);
```

h. **Γράψτε τον ακόλουθο κώδικα μέσα στη συνάρτηση ShowUI:**

```
_rectPanelRoulette.gameObject.SetActive(state == UIState.ROULETTE);
        _rectPanelCustom.gameObject.SetActive(state == UIState.CUSTOM);
```

i. **Γράψτε τον ακόλουθο κώδικα μέσα στη συνάρτηση SetSpinnerRotation:**

```
rectSpinner.eulerAngles = new Vector3(0, 0, rot);
```

j. **Γράψτε τον ακόλουθο κώδικα μέσα στη συνάρτηση SetArrowRotation:**

```
rectArrow.eulerAngles = new Vector3(0, 0, rot);
```

k. **Γράψτε τον ακόλουθο κώδικα μέσα στη συνάρτηση SetIcon:**

```
imageIcon.sprite = sprite;
```

l. **Γράψτε τον ακόλουθο κώδικα μέσα στη συνάρτηση SetResultText:**

```
txtResult.text = string.Format("You got a {0}", result);
```

m. **Γράψτε τον ακόλουθο κώδικα μέσα στη συνάρτηση ShowItemPercentage:**

```
if(updateToggle)
            ToggleTextPercentage();

        for (int i = 0; i < data.Length; i++)
            textPercentages[i].text = string.Format("{0}%", data[i].percentage);
```

n. **Γράψτε τον ακόλουθο κώδικα μέσα στη συνάρτηση ShowContentItem:**

```
foreach(Transform t in _contentItem.transform)
        {
            if (t != _contentItem.transform)
                Destroy(t.gameObject);
        }

        for (int i = 0; i < data.Length; i++)
        {
            GameObject view = Instantiate(item);
            view.SetActive(true);
            view.transform.SetParent(_contentItem);

            ItemView viewInput = view.GetComponent<ItemView>();
            string percentage = string.Format("{0} %", data[i].percentage);
            viewInput.SetValue(data[i].sprite, data[i].name, percentage);

            view.transform.localScale = Vector3.one;
        }
```

o. **Γράψτε τον ακόλουθο κώδικα μέσα στη συνάρτηση ShowContentItemInput:**

```
foreach (Transform t in _contentItemInput.transform)
        {
            if (t != _contentItemInput.transform)
                Destroy(t.gameObject);
        }

        for (int i = 0; i < data.Length; i++)
        {
            GameObject view = Instantiate(itemInput);
            view.SetActive(true);
            view.transform.SetParent(_contentItemInput);

            ItemViewInput viewInput = view.GetComponent<ItemViewInput>();
            string percentage = string.Format("{0}", data[i].percentage);
            viewInput.SetValue(data[i].sprite, data[i].name, percentage);

            view.transform.localScale = Vector3.one;
        }
```

p. **Γράψτε τον ακόλουθο κώδικα μέσα στη συνάρτηση SetInputValueToData:**

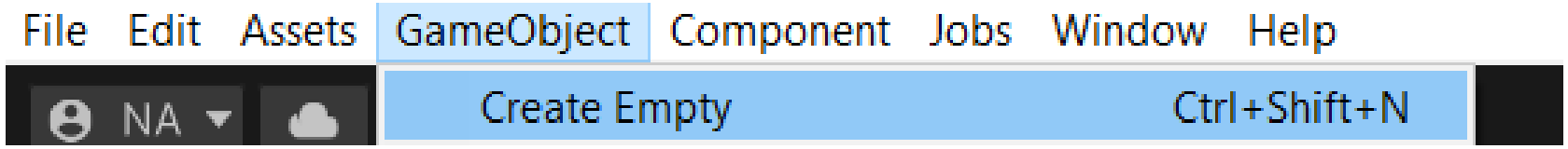
```
ItemViewInput[] inputs =
_contentItemInput.GetComponentsInChildren<ItemViewInput>();

int total = 0;
int maxValue = 100;
for(int i = 0; i < inputs.Length; i++)
{
    int val = int.Parse(inputs[i].GetInputValue);
    int valTotal = val + total;

    if(valTotal > maxValue)
    {
        val = maxValue - total;
    }

    if (total > maxValue)
    {
        int diff = total - maxValue;
        data[i].percentage = diff;
        total = maxValue;
    }
    else if(total == maxValue)
    {
        data[i].percentage = 0;
    }
    else
    {
        total += val;
        data[i].percentage = val;
    }
}

if (total < maxValue)
{
    int count = 0;

    for (int i = 0; i < inputs.Length; i++)
    {
        int val = int.Parse(inputs[i].GetInputValue);
        if (val == 0)
            count++;
    }


    int sum = maxValue - total;
    int[] tempValue = Utils.GetRandomDistribution(sum, count);

    count = 0;
    for(int i = 0; i < data.Length; i++)
    {
        if(data[i].percentage == 0)
        {
            data[i].percentage = tempValue[count];
            count++;
        }
    }
}
```

q. **Γράψτε τον ακόλουθο κώδικα μέσα στη συνάρτηση ToggleTextPercentage:**

```
toggle = !toggle;

for (int i = 0; i < textPercentages.Length; i++)
    textPercentages[i].gameObject.SetActive(toggle);

textButtonShowHide.text = toggle ? "HIDE \n PERCENTAGE" :
"SHOW \n PERCENTAGE";
```

Ρύθμιση αντικειμένου SpinController

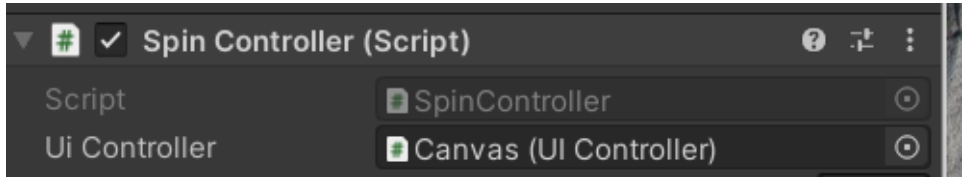
Δημιουργήστε ένα κενό αντικείμενο παιχνιδιού.
Ορίστε το όνομα GameObject σε "SpinController".



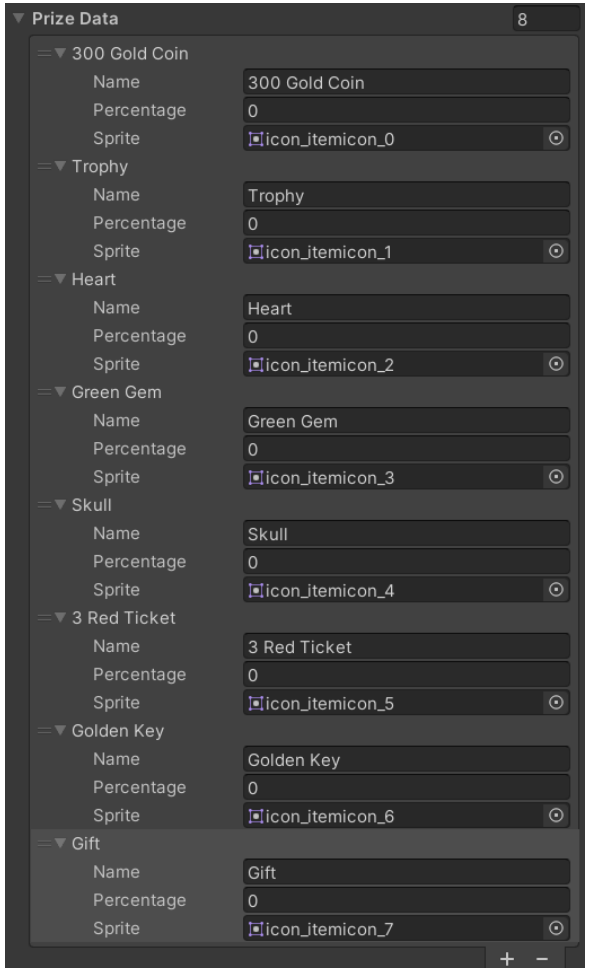
Προσθήκη συναρτήσεων σεναρίου
Προσθέστε τον ακόλουθο κώδικα κάτω από τη συνάρτηση Start().:

```
private void OnClickSpin()
{
}

private void OnClickClose()
{
}

private void RandomizePercentage()
{
}

private IEnumerator Spin()
{
    yield return null;
}
```

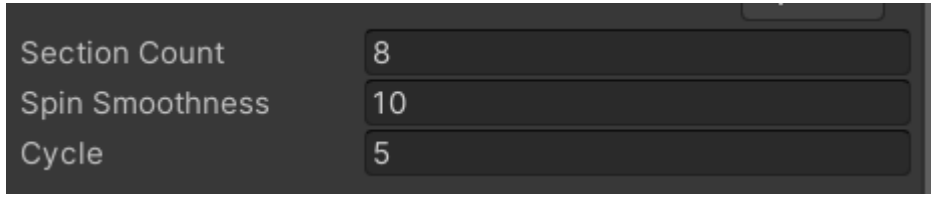
- Επιστρέψτε στο Unity Editor.
- Επιλέξτε SpinController GameObject.
- Σύρετε και αποθέστε το Canvas GameObject στη μεταβλητή UIController. Δείτε την παρακάτω εικόνα:



- Ορίστε το μήκος PrizeData σε 8. Εφαρμόστε πληροφορίες δεδομένων Prize για κάθε πίνακα δεδομένων Prize. Δείτε την επόμενη εικόνα:



- Ορίστε την ενότητα Count σε 8. spinSmoothness = 10; κύκλος = 5;



Ρύθμιση σεναρίου SpinController (συνέχεια)

1. Κάντε διπλό κλικ στη δέσμη ενεργειών UIController στο παράθυρο του έργου.
2. **Ας γράψουμε το Σενάριο μας:**

A)Γράψτε τον ακόλουθο κώδικα μέσα στη συνάρτηση Start():

```
uiController.ShowUI(UIState.ROULETTE);

_isSpinning = false;
_totalAngle = 360 / sectionCount;

uiController.ShowParticleSelected(false);
uiController.ShowParticleResult(false);
uiController.ShowTextResult(false);

RandomizePercentage();

uiController.AddButtonSpinListener(OnClickSpin);
uiController.ShowItemPercentage(prizeData);

uiController.ShowContentItem(prizeData);
uiController.ShowContentItemInput(prizeData);

uiController.AddButtonRandomListener(()=>
{
        RandomizePercentage();

        uiController.ShowContentItem(prizeData);
        uiController.ShowContentItemInput(prizeData);
        uiController.ShowItemPercentage(prizeData, false);
});

uiController.AddButtonSetValueListener(()=>
{
        uiController.SetInputValueToData(prizeData);

        uiController.ShowContentItem(prizeData);
        uiController.ShowContentItemInput(prizeData);
        uiController.ShowItemPercentage(prizeData, false);
});
```

b. **Γράψτε τον ακόλουθο κώδικα μέσα στη συνάρτηση**

```
if(!_isSpinning)
        StartCoroutine(Spin());
```

c. **Γράψτε τον ακόλουθο κώδικα μέσα στη συνάρτηση OnClickClose():**

```
Application.Quit();
```

d. **Γράψτε τον ακόλουθο κώδικα μέσα στη συνάρτηση RandomizePercentage():**

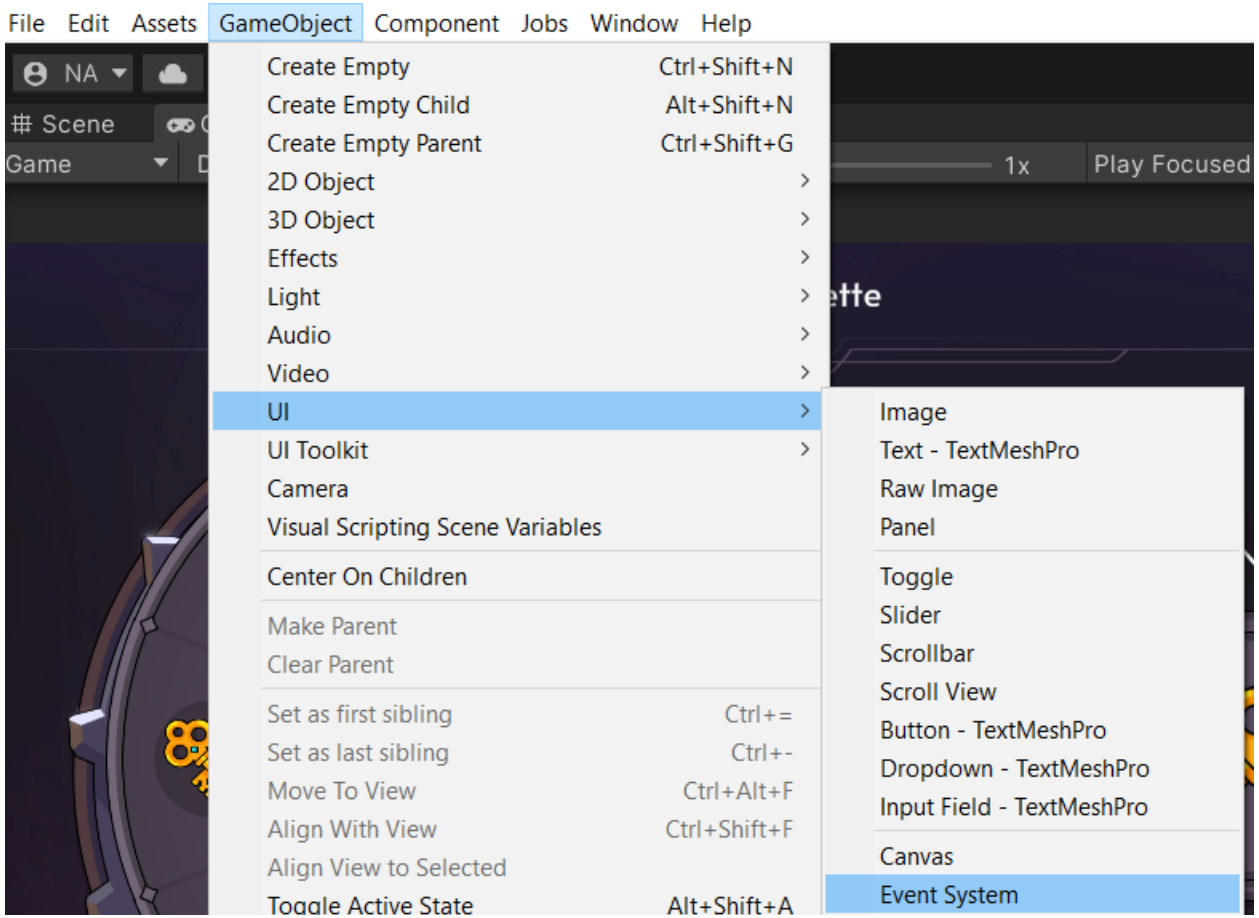
```
int[] percentage = Utils.GetRandomDistribution(100, sectionCount);
        for (int i = 0; i < percentage.Length; i++)
                prizeData[i].percentage = percentage[i];
```

e. **Γράψτε τον ακόλουθο κώδικα μέσα στη συνάρτηση Spin():**

```
uiController.ShowParticleSelected(false);
uiController.ShowParticleResult(false);
uiController.ShowTextResult(false);

_isSpinning = αληθινό;

_prizePercentage.Clear();
για (int x = 0; x < prizeData.Length; x++)
_prizePercentage.Add(prizeData[x].percentage);

int itemIndex = Utils.GetPrizeByProbability(_prizePercentage);
_randomValue = ((sectionCount * cycle) * spinSmoothness) +
((itemIndex * spinSmoothness) - (_prevIndex * spinSmoothness));
float increment = 0;
για (int i = 0; i < _randomValue; i++)
{
float angle = _totalAngle / spinSmoothness;

_zRotation += angle;
uiController.SetSpinnerRotation(_zRotation);
float time = (float)i / (float)_randomValue;
float interval = Mathf.Lerp(0.01f, 0.1f, time * time);
float angleSpeed = Mathf.Lerp(1f, 0.5f, time);

increment += angleSpeed ;
float sin = Mathf.Sin(increment);

uiController.SetArrowRotation(sin * 15f);

uiController.SetIcon(prizeData[UnityEngine.Random.Range(0,
prizeData.Length)].sprite);
yield return new WaitForSeconds(interval);
}

uiController.SetArrowRotation(0);
_prevIndex = itemIndex;
_isSpinning = false;

uiController.SetIcon(prizeData[itemIndex].sprite);
uiController.SetResultText(prizeData[itemIndex].name);

uiController.ShowParticleSelected(true);
uiController.ShowParticleResult(true);
uiController.ShowTextResult(true);
```

Ρύθμιση του EventSystem

Προσθέστε EventSystem μεταβαίνοντας στο GameObject->UI->EventSystem



smart AInnovators

Ευχαριστούμε πολύ

smartainnovators.eu

Co-funded by the European Union

This project has been co-funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.Project number: 2021-1-PL01-KA220-VET-000033011

www.smartainnovators.eu