INFORMATIKPROJEKT

PROJEKTMAPPE



Gliederung

|  |  |
| --- | --- |
| 3 | Ideen-Entwicklung |
| 3 | Vorbereitung |
| 4 | Entwicklung |
| 5 | Versionsverlauf |
| 6 | Skripts |
|  | Anmerkungen |

Ideen-Entwicklung

Ich hatte am Anfang die Idee, ein 3D Spiel zu schreiben, das hat sich aber wenden den überaus schlechten Schulcomputern als unmöglich rausgestellt.

Da ich in meiner Freizeit gerne das Spiel EGG.INC spiele, hatte ich mich bei meinem weiteren Versuch, ein Spiel zu programmieren etwas davon inspirieren lassen. Des weitern wollte ich die Handy-App in dem Prinzip von Cookieclicker halten und eine Art Idle-game daraus machen.

Die Idee von EggFarmers war geboren.

Das Spiel sollte aus einem Huhn als Clicker und einem Shop mit Upgrades bestehen.

Vorbereitung

Zuerst habe ich das Design des Hauptbildschirmes in Procreate erstellt:

Dann habe ich das Shop Design in Procreate erstellt:



Ich habe das Projekt als ein Unity-projekt gestartet, da dies mir am einfachsten zu designen schien.

Entwicklung

Zuerst habe ich die, in meine vorläufigen Designs verwendeten Grafiken, als Unity-Assets importiert und alle nötigen Grundeinstellungen geändert. Da ich ein Handyspiel programmieren wollte, habe ich die handytypische Ausrichtung 1080x1920 gewählt. Im Folgenden habe ich nun den Hintergrund und alle nötigen UI Elemente eingefügt, den entsprechenden Assets zugewiesen sowie beschriftet und größenmäßig angepasst.

Dann habe ich angefangen, mich mit den jeweiligen Mechanismen in C# auseinanderzusetzen. Da mir jeweilige Programmierkenntnisse als auch Erfahrungen im Zusammenhang mit C# fehlten, habe ich mir einige Anfängerskripts als Beispiel angeschaut um mir Grundkenntnisse zu schaffen.

Insgesamt habe 5 separate Skripts erstellt:

* GameManager
  + Accespoint für die Eieranzahl
  + Accespoint für den Multipler
* Spiel (Mainscript)
  + Festlegen der Zahlen
  + Alle Berechnungen für weitere Käufe
  + Alle buy Funktionen
  + Alle Shop Aktualisierungen
* ShopOpen
  + Öffnet den Shop, wenn man auf den „Shop“ Button kommt
  + Schließt den Shop, wenn man auf das Kreuz kommt
* SettingsOpen
  + Öffnet die Einstellungen, wenn man auf das Zahnrad kommt
  + Schließt die Einstellungen, wenn man auf das Kreuz kommt
* Exitscript
  + Schießt das Spiel komplett

Allem in allem war es schon recht kompliziert, da man immer Umwege um Fehlerlastige Befehle finden musste.

Versionsverlauf

Version: 1.0.0:

-projekt erstellt

-hintergrund und clicker/counter hinzugefügt

Version: 1.0.1:

-Shop UI erstellt

-design vorbereitet

Version 1.0.2:

-bug fix beim starten der App

Version 1.0.3:

-Shop UI design komplett

Version 1.0.4:

-Variablen erstellen

-preise und multipler erstellen

-bug fixes

Version 1.0.5:

-shop Skripts komplett

-bug fixes

Version 1.0.6:

-restlichen Skripte hinzugefügt

-Bug fixes

Version 1.0.7:

-Bug fixes

Release 1.0.8:

-Blue Screen beim Starten des Spieles gefixt

Release 1.0.9:

-anpassen von Skalierungen

Release 1.1.0:

-rechnungen hinzugefüge

Release 1.1.1:

-Restlichen Bugs gefixed

**Release 2.0**

Skripts

GameManager:

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class GameManager : MonoBehaviour

{

    public static int Eier;

    public static int multi;

    void Start()

    {

        Eier = PlayerPrefs.GetInt("Eier", 0);

        multi = PlayerPrefs.GetInt("multi", 1);

    }

}

Spiel:

using System.Collections;

using UnityEngine.UI;

using UnityEngine;

public class spiel : MonoBehaviour

{

    public Text eieranzahl;

    public Text Kosten01;

    public Text Kosten02;

    public Text Kosten03;

    public Text Kosten04;

    public Text Kosten05;

    public Text Multi01;

    public Text Multi02;

    public Text Multi03;

    public Text Multi04;

    public Text Multi05;

    public static int Basis\_Kosten1; ///////////////////////////// Base Cost

    public static int Basis\_Kosten2;

    public static int Basis\_Kosten3;

    public static int Basis\_Kosten4;

    public static int Basis\_Kosten5;

    public static int Basis\_Multi1; ////////////////////////////// base Revenue

    public static int Basis\_Multi2;

    public static int Basis\_Multi3;

    public static int Basis\_Multi4;

    public static int Basis\_Multi5;

    public static int Neu\_Kosten1; //////////////////////////// Cost next

    public static int Neu\_Kosten2;

    public static int Neu\_Kosten3;

    public static int Neu\_Kosten4;

    public static int Neu\_Kosten5;

    public static int Neu\_Multi1; ////////////////////Multi next

    public static int Neu\_Multi2;

    public static int Neu\_Multi3;

    public static int Neu\_Multi4;

    public static int Neu\_Multi5;

    public static int Multi001; //////////////////////////////// Cost Factor

    public static int Multi002;

    public static int Multi003;

    public static int Multi004;

    public static int Multi005;

    public static int Anzahl1; //////////////////////////// Owned

    public static int Anzahl2;

    public static int Anzahl3;

    public static int Anzahl4;

    public static int Anzahl5;

    public static int rechenmulti01; //////////////////////////// rechenmulti für potenz

    public static int rechenmulti02;

    public static int rechenmulti03;

    public static int rechenmulti04;

    public static int rechenmulti05;

    public void eier\_legen()

    {

        GameManager.Eier += GameManager.multi;

        PlayerPrefs.SetInt("Eier", GameManager.Eier);

    }

    void Start()

    {

        Basis\_Kosten1 = 10;  ////////////////////////Base cost

        Basis\_Kosten2 = 100;

        Basis\_Kosten3 = 5000;

        Basis\_Kosten4 = 47000;

        Basis\_Kosten5 = 230000;

        Basis\_Multi1 = 1; ///////////////////////////////Base Revenue

        Basis\_Multi2 = 10;

        Basis\_Multi3 = 20;

        Basis\_Multi4 = 50;

        Basis\_Multi5 = 100;

        Anzahl1 = 0; /////////////////////////////////Owned

        Anzahl2 = 0;

        Anzahl3 = 0;

        Anzahl4 = 0;

        Anzahl5 = 0;

        Neu\_Kosten1 = 0; //////////////////////////Cost next

        Neu\_Kosten2 = 0;

        Neu\_Kosten3 = 0;

        Neu\_Kosten4 = 0;

        Neu\_Kosten5 = 0;

        Multi001 = 11 / 10; ///////////////////////Cost Factor

        Multi002 = 13 / 10;

        Multi003 = 15 / 10;

        Multi004 = 17 / 10;

        Multi005 = 19 / 10;

        Neu\_Multi1 = 0; /////////////////////Multi next

        Neu\_Multi2 = 0;

        Neu\_Multi3 = 0;

        Neu\_Multi4 = 0;

        Neu\_Multi5 = 0;

        //rechenmulti01 = 0;

        //rechenmulti02 = 0;

        //rechenmulti03 = 0;

        //rechenmulti04 = 0;

        //rechenmulti05 = 0;

        ///////////////////////////////////////////////////

        ///////////////////////////////////////////////////

        ///////////////////////////////////////////////////

        ////////////Rechnungen/////////////////////////////

        //rechenmulti01 = Math.Pow(Multi001, Anzahl1);

        //rechenmulti02 = Math.Pow(Multi002, Anzahl2);

        //rechenmulti03 = Math.Pow(Multi003, Anzahl3);

        //rechenmulti04 = Math.Pow(Multi004, Anzahl4);

        //rechenmulti05 = Math.Pow(Multi005, Anzahl5);

        //Neu\_Kosten1 = Basis\_Kosten1 \* Multi001 \* Anzahl1;

        //Neu\_Kosten2 = Basis\_Kosten2 \* Multi002 \* Anzahl2;

        //Neu\_Kosten3 = Basis\_Kosten3 \* Multi003 \* Anzahl3;

        //Neu\_Kosten4 = Basis\_Kosten4 \* Multi004 \* Anzahl4;

        //Neu\_Kosten5 = Basis\_Kosten5 \* Multi005 \* Anzahl5;

        //Neu\_Multi1 = Basis\_Multi1 \* Multi001 \* Anzahl1;

        //Neu\_Multi2 = Basis\_Multi2 \* Multi002 \* Anzahl2;

        //Neu\_Multi3 = Basis\_Multi3 \* Multi003 \* Anzahl3;

        //Neu\_Multi4 = Basis\_Multi4 \* Multi004 \* Anzahl4;

        //Neu\_Multi5 = Basis\_Multi5 \* Multi005 \* Anzahl5;

    }

    public void shop\_kauf(int shop)

    {

        if(shop == 1 && GameManager.Eier >= Neu\_Kosten1)

        {

            GameManager.multi += Neu\_Multi1;

            GameManager.Eier -= Neu\_Kosten1;

            spiel.Anzahl1 += 1;

            PlayerPrefs.SetInt("Eier", GameManager.Eier);

            PlayerPrefs.SetInt("multi", GameManager.multi);

        }

        if(shop == 2 && GameManager.Eier >= Neu\_Kosten2)

        {

            GameManager.multi += Neu\_Multi2;

            GameManager.Eier -= Neu\_Kosten2;

            spiel.Anzahl2 += 1;

            PlayerPrefs.SetInt("Eier", GameManager.Eier);

            PlayerPrefs.SetInt("multi", GameManager.multi);

        }

        if(shop == 3 && GameManager.Eier >= Neu\_Kosten3)

        {

            GameManager.multi += Neu\_Multi3;

            GameManager.Eier -= Neu\_Kosten3;

            spiel.Anzahl3 += 1;

            PlayerPrefs.SetInt("Eier", GameManager.Eier);

            PlayerPrefs.SetInt("multi", GameManager.multi);

        }

        if(shop == 4 && GameManager.Eier >= Neu\_Kosten4)

        {

            GameManager.multi += Neu\_Multi4;

            GameManager.Eier -= Neu\_Kosten4;

            spiel.Anzahl4 += 1;

            PlayerPrefs.SetInt("Eier", GameManager.Eier);

            PlayerPrefs.SetInt("multi", GameManager.multi);

        }

        if(shop == 5 && GameManager.Eier >= Neu\_Kosten5)

        {

            GameManager.multi += Neu\_Multi5;

            GameManager.Eier -= Neu\_Kosten5;

            spiel.Anzahl5 += 1;

            PlayerPrefs.SetInt("Eier", GameManager.Eier);

            PlayerPrefs.SetInt("multi", GameManager.multi);

        }

    }

    void Update()

    {

        Neu\_Kosten1 = Basis\_Kosten1 \* Multi001 \* Anzahl1 + Basis\_Kosten1;

        Neu\_Kosten2 = Basis\_Kosten2 \* Multi002 \* Anzahl2 + Basis\_Kosten2;

        Neu\_Kosten3 = Basis\_Kosten3 \* Multi003 \* Anzahl3 + Basis\_Kosten3;

        Neu\_Kosten4 = Basis\_Kosten4 \* Multi004 \* Anzahl4 + Basis\_Kosten4;

        Neu\_Kosten5 = Basis\_Kosten5 \* Multi005 \* Anzahl5 + Basis\_Kosten5;

        Neu\_Multi1 = Basis\_Multi1 \* Multi001 \* Anzahl1 + Basis\_Multi1;

        Neu\_Multi2 = Basis\_Multi2 \* Multi002 \* Anzahl2 + Basis\_Multi2;

        Neu\_Multi3 = Basis\_Multi3 \* Multi003 \* Anzahl3 + Basis\_Multi3;

        Neu\_Multi4 = Basis\_Multi4 \* Multi004 \* Anzahl4 + Basis\_Multi4;

        Neu\_Multi5 = Basis\_Multi5 \* Multi005 \* Anzahl5 + Basis\_Multi5;

        eieranzahl.text = "Eier: " + GameManager.Eier;

        Kosten01.text = "Eier: " + spiel.Neu\_Kosten1;

        Kosten02.text = "Eier: " + spiel.Neu\_Kosten2;

        Kosten03.text = "Eier: " + spiel.Neu\_Kosten3;

        Kosten04.text = "Eier: " + spiel.Neu\_Kosten4;

        Kosten05.text = "Eier: " + spiel.Neu\_Kosten5;

        Multi01.text = "+ " + spiel.Neu\_Multi1;

        Multi02.text = "+ " + spiel.Neu\_Multi2;

        Multi03.text = "+ " + spiel.Neu\_Multi3;

        Multi04.text = "+ " + spiel.Neu\_Multi4;

        Multi05.text = "+ " + spiel.Neu\_Multi5;

    }

}

ShopOpen / SettingsOpen:

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class ShopOpen01 : MonoBehaviour

{

    public GameObject ShopUI;

    public void OpenShopPannel()

    {

        if(ShopUI != null)

        {

            bool isActive = ShopUI.activeSelf;

            ShopUI.SetActive(!isActive);

        }

    }

}

Exitscript:

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class Exitscript : MonoBehaviour

{

    public void Exit()

    {

        Application.Quit();

    }

}

Anmerkungen:

Am kompliziertesten war es die Grenze zu finden, ab wann die Upgrades exponentiell teurer werden, und diese Funktion mit in das Programm einzubringen.

* Veröffentlichung auf <https://github.com/IzzLenn/EggFarmers>