Sweet Clicker - Сладкий Кликер

Deadline – 8 working days

Idea

Пользователь кликает на кубик сахара, получая за каждый клик 1 ед. сахара. Сахар конвертируется в монеты для покупки других сладостей. Другие сладости приносят больше сахара.

Design

В игре две валюты – сахар и деньги.

Существует 7 видов сладостей, которые можно кликать – сахар, конфета, кекс, шоколадка, чизкейк, желе, торт.

Сахар приносит 1 ед. сахара, конфета – 3 ед. сахара, кекс – 5 ед. сахара, шоколадка – 10 ед. сахара, чизкейк - 30 ед. сахара, желе – 50 ед. сахара, торт – 100 ед. сахара.

1 ед. сахара продается за 2 монеты. Сахар конвертируется в деньги одним из 5-ти способов.

1. 100 монет за 50 сахара
2. 500 монет за 250 сахара
3. 1000 монет за 500 сахара
4. 5000 монет за 2500 сахара
5. 10000 монет за 5000 сахара

*Механизм открытия новых сладостей*

Конфета – 50 кликов сахара

Кекс – 100 кликов конфеты

Шоколадка – 150 кликов кекса

Чизкейк – 200 кликов шоколадки

Желе – 250 кликов чизкейка

Торт – 300 кликов желе

Изначально открыт только сахар.

Конфета – 100 монет

Кекс – 600 монет,

Шоколадка – 1500 монет,

Чизкейк – 4000 монет,

Желе - 15000 монет,

Торт – 30000 монет

Gameplay

В игре присутствует одна главная сцена – Gameplay.

Сладость-кликер находится по центру экрана. При нажатии размер увеличивается, активируется эффект – пузыри и проигрывается звук с зачислением сахара на счет.

Задний фон и кликер находятся в движении.

Сверху по центру расположены два счетчика монет и сахара. Снизу по середине по порядку расположены кнопки панелей обменника, кликеров и настроек.

В сцене проигрывается единственный soundtrack – Main\_Soundtrack.

*Панели*

1) Панель обменника содержит 5 кнопок обмена. При нажатии проигрывается эффект и звук.

2) Панель кликеров содержит 7 кнопок каждой сладости. Некупленные сладости имеют кнопку покупки с обозначением цены. При покупке проигрывается звук и эффект.

3) Панель настроек регулирует два компонента – звук и музыка, а также содержит кнопку сброса прогресса.

Tools and Approach

Game Engine – Unity 2018

Language – C#

IDE – Visual Studio 2017

Image Editor - GIMP

Sprites and Soundtrack & Sound FX – Downloaded

Scripting

*public class Gameplay : MonoBehaviour*

*public static int sugar, money;*

*public Text sugarT, moneyT;*

*public AudioSource musicSource, soundSource;*

*public AudioClip moneyClip, buttonClip, bubbleClip;*

*public Slider musicSlider, soundSlider;*

*public GameObject ExchangerPanel, ClickersPanel, SettingsPanel;*

*public Button ExchangerBtn, ClickersBtn, SettingsBtn;*

*public int[] sugarTrade = new int[5] {50, 250, 500, 2500, 5000};*

*public Button Exchange1, Exchange2, Exchange3, Exchange4, Exchange5;*

*public int[] clickersPrice = new int[6] {100, 600, 1500, 4000, 15000, 30000};*

*public Button BuyCandyBtn, BuyCupcakeBtn, BuyChocolateBtn, BuyCheesecakeBtn, BuyJellyBtn, BuyCakeBtn;*

*public Button ChooseSugarBtn, ChooseCandyBtn, ChooseCupcakeBtn, ChooseChocolateBtn, ChooseCheesecakeBtn, ChooseJellyBtn, ChooseCakeBtn;*

*public ParticleSystem bubblesEffect;*

*public int[] sugarProfit = new int[7] {1, 3, 5, 10, 30, 50, 100};*

*public Sprite[] clickerSprites = new Sprite[7];*

*public string[] tags = new string[7]{ “Sugar”, “Candy”, “Cupcake”, “Chocolate”, “Cheesecake”, “Jelly”, “Cake” };*

*private int chosenClicker;*

*private List<int> ownClickers;*

*public Button ClickerBtn;*

*private int currentProfit*

*public Animator animator;*

*public Button ResetBtn;*

Methods

*- private void Awake()*

LoadGame(), Listeners1(), Listeners2()

- private void Listeners2()

ResetProgress(), AdjustMusic(), AdjustSound(), Exchanger(), Clickers(), Settings(), OnClicker() listeners

- private void *Listeners1()*

BuyClicker(), ChooseClicker(), Exchange() listeners

*- private void Update()*

update sugarT, moneyT

*- private void ExchangerPanel(), ClickersPanel(), SettingsPanel()*

invoke panel, disable other panels, play sound

*- private void AdjustMusic(), AdjustSound()*

adjust music, sound source volumes

*- private void Exchange(int sugarAmount)*

check sugar, subtract sugar, add twice more money, play sound of money and click

*- private void BuyClicker(Button buyBtn, Button chooseBtn, int clickerPrice)*

check money, subtract value, deactivate buyBtn, make interactable chooseBtn, play two sounds, ChooseClicker(), ownClickers.Add()

- *private void ChooseClicker(Button chosenBtn)*

color all buttons white and chosen as green, set chosenClicker, apply sprite to ClickerBtn, set currentProfit, play sound

- *private int GetClickerIndex(Button btn)*

“for” loop with tags[] compares btn to get index

- *private Button GetClickerButton(int index)*

switch statement returns appropriate to index button

*- private SaveLoad CreateSaveGameObject()*

creates Save object and apply 7 main variables to it, return the object

*- private void SaveGame()*

serialize Save object

*- private void LoadGame()*

If finds the file: deserialize Save object, apply save values to 7 main variables and sliders, ChooseClicker(), SetClickers(); else ResetProgress()

- *private void ResetProgress()*

Default values to 9 variables: sugar, money, musicSource, soundSource, musicSlider, soundSlider, chosenClicker, ownClickers, currentProfit. ChooseClicker(), SetClickers.

- *private void SetClickers()*

check ownClickers for containing indexes; deactivate buy-buttons, make interactable choose-buttons and vice versa

*- private void OnClicker()*

sugar += currentProfit, play bubble sound, play effect, play animation

*- private void OnDisable()*

SaveGame()

*[Serializable]*

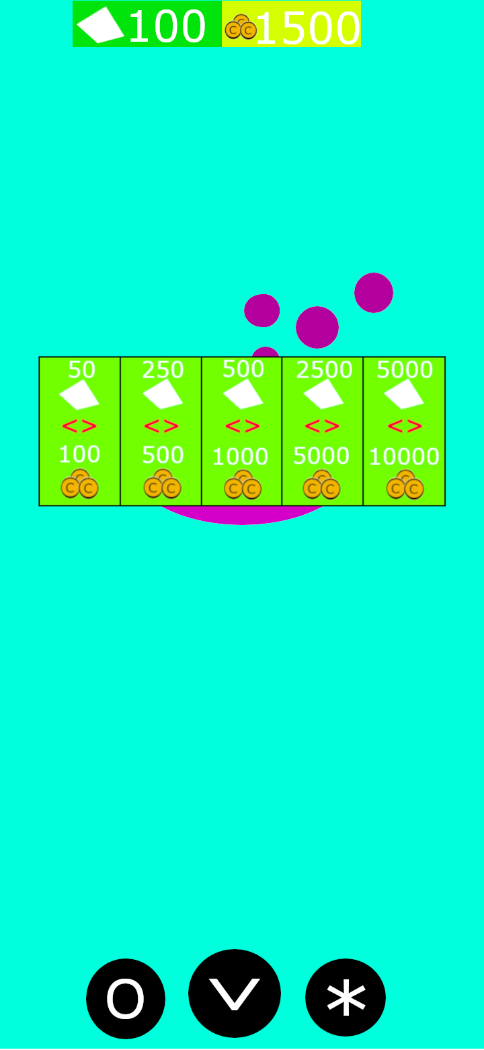
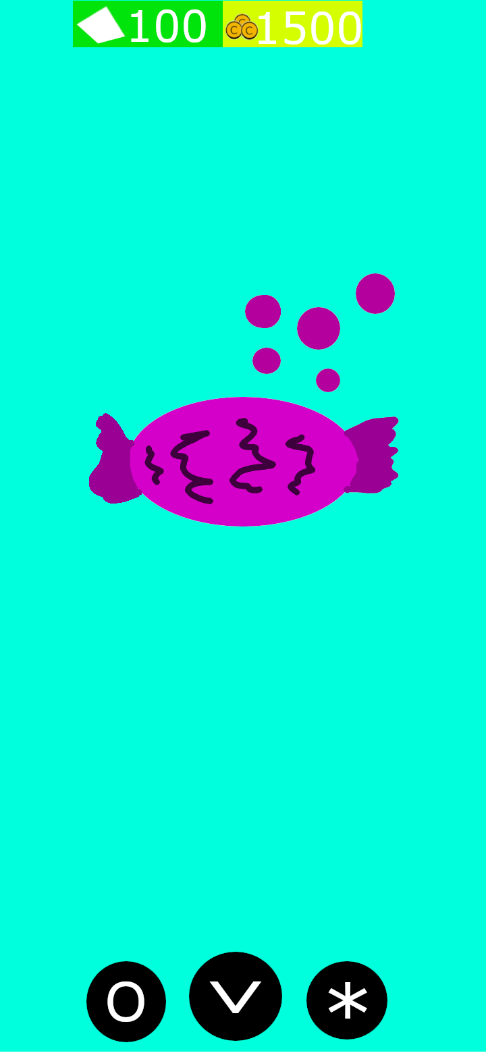
*public class SaveLoad*

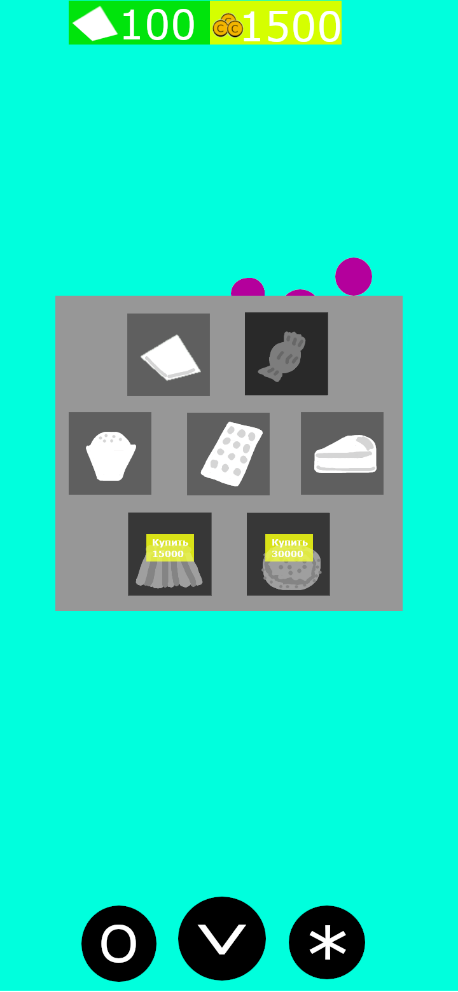
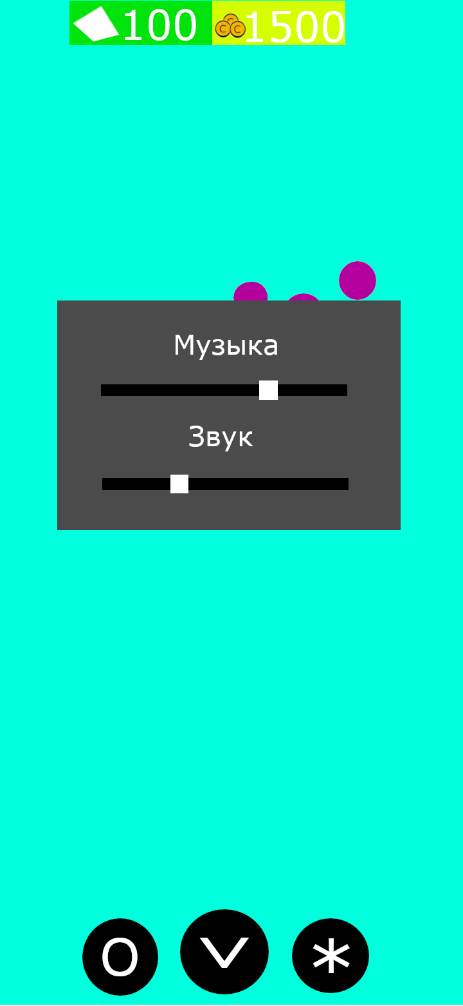
*public int sugar, money, chosenClicker, currentProfit;*

*public float soundVolume, musicVolume;*

*public List<int> ownClickers;*

Sketches



*Gameplay, Exchanger Panel, Clickers Panel, Settings Panel*

Editor Components

*Game Objects*

gameplay, camera, event system

*GUI*

all the rest

Target platforms

*Android, PC*

Assets

*Sprites*

Coin, Sugar, Candy, Cupcake, Chocolate, Cheesecake, Jelly, Cake, Game Icon, bubble, exchange icon, background; buttons of settings, clickers, exchange.

*Soundtrack & SoundFX*

Soundtrack; sounds of buttons click, clickers bubble, money

*Font*

Charge-Vector

*Scripts*

Gameplay

Testing

Game play time 5 - 10 minutes

Steps of Development

1. Prepare assets
2. Create visual prototype
3. Implement code
4. Complete game
5. Compile for Android, PC
6. Testing, debugging
7. Update doc

Steps of Coding

1. Declare classes, variables, methods
2. Describe methods
3. Apply code to game
4. Debug code
5. Optimize code
6. Update doc