VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belgaum-590018

A Database Management System Mini Project Report
on

"Gamification of Digital Education"

Submitted in Partial fulfillment of the Requirements for the V Semester of the Degree of

Bachelor of Engineering
In
Computer Science & Engineering
By

ABHIGYAN
(1CR19CS004)

AAYUSH DUBEY
(1CR19CS003)

Under the Guidance of

Dr. R. Kesavamoorthy
Assoc Professor, Dept. of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR INSTITUTE OF TECHNOLOGY

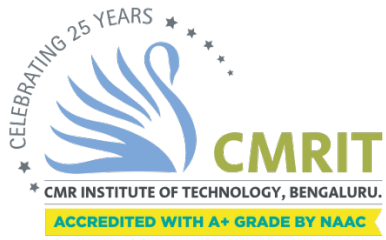
#132, AECS LAYOUT, IT PARK ROAD, KUNDALAHALLI,
BANGALORE-560037

CMR INSTITUTE OF TECHNOLOGY

#132, AECS LAYOUT, IT PARK ROAD, KUNDALAHALLI,

BANGALORE-560037

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



CERTIFICATE

This is to certify that the Database Management System Project work entitled **"Gamification of Digital Education"** has been carried out by **Abhigyan (1CR19CS004)** and **Aayush Dubey (1CR19CS003)** bonafide students of CMR Institute of Technology in partial fulfillment for the award of **Bachelor of Engineering** in **Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year **2021-2022**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. This DBMS Project Report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

----------	**----------**
Signature of Guide	**Signature of HOD**
Dr. R. Kesavamoorthy **Assoc Professor** **Dept. of CSE, CMRIT**	**Dr. Shreekanth M.Prabhu** **Professor, Head** **Dept. of CSE, CMRIT**

External Viva

Name of the examiners — Signature with date

1.

2.

ABSTRACT

The proposed database management system project is a Gamification of Digital Education. This is designed and developed for an interactive gaming-based learning experience through fun puzzles and quizzes. This provides a unique and rewarding method of learning concepts and proves to be a better method of grasping the topics and traditional method of learning.

The Technologies used are Unity, SQLite and SQLiteUnityKit and the languages used are C# and SQL (Structured Query Language).

The primary purpose of this Gaming Application is to provide an alternative, fun and interactive way of learning with scores, coins and rewards to boost the motivation of the learner. The learner can use the coins to purchase various in-game items for a more personalized experience and can complete with other players.

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of project would be incomplete without mentioning the people who made it possible, whose constant guidance and encouragement crowned our effort with success.

First and foremost, we would like to express our sincere words of gratitude and respect to our university, Visvesvaraya Technological University, Belgaum and our college CMR Institute of Technology, Bangalore for providing us an opportunity to carry out our project work report.

We express our sincere thanks to Dr. Shreekanth M.Prabhu, HOD, Lecturer, Dept of CSE, and our guides Mr. Kartheek G C R, Assistant Professor, Dept of CSE and Dr. R. Kesavamoorthy, Associate Professor, Dept of CSE who have been driving force behind the completion of our Project report and helping us in every aspect of project development.

Last but not the least we extend our thanks to all the people in the Department of Computer Science, for always being helpful. We are very grateful to our parents and well-wishers for their continuous moral support and encouragement.

TABLE OF CONTENT

LIST OF FIGURES

Chapter 1

INTRODUCTION

This is an application targeted towards a field that has much to improve in terms of quality of education and the various ways and methods of propagating the information targeted towards learners. This app enables a way of learning through fun puzzles and quizzes involving many obstacles and fun elements essentially making learning a very fun and innovative process.

The use of games and activities is very minimal in the current architecture of educational process. This project aims at alleviating the gap and forming strong foundational bonds between games and learning. Games have always been thought of as a mode of relaxing and having fun, although deemed in a sense not good or a waste of time by the caretakers of learners. Studies prove that gaming can actually increase brain co-ordination and function significantly and now combined with learning real world technologies and courses, this could revolutionize how we look at gaming.

This is step towards a better and more connected social way of learning, that is sure to be fun, interactive and a better way of learning.

Chapter 2

SYSTEM REQUIREMENT

i. Operating systems: Windows XP,7,8,10,11, Linux, MacOS X

ii. Processors: Intel x64, AMD x64, apple silicon, ARM based processors

iii. Size: >100MB Hard disk space

iv. RAM: minimum 2 GB

v. Resolution: minimum 1280*720, other resolutions supported at 16:9 aspect ratio

vi. GPU: any integrated or dedicated GPU with minimum 256MB VRAM

Chapter 3

DESIGN

3.1 ER DIAGRAM

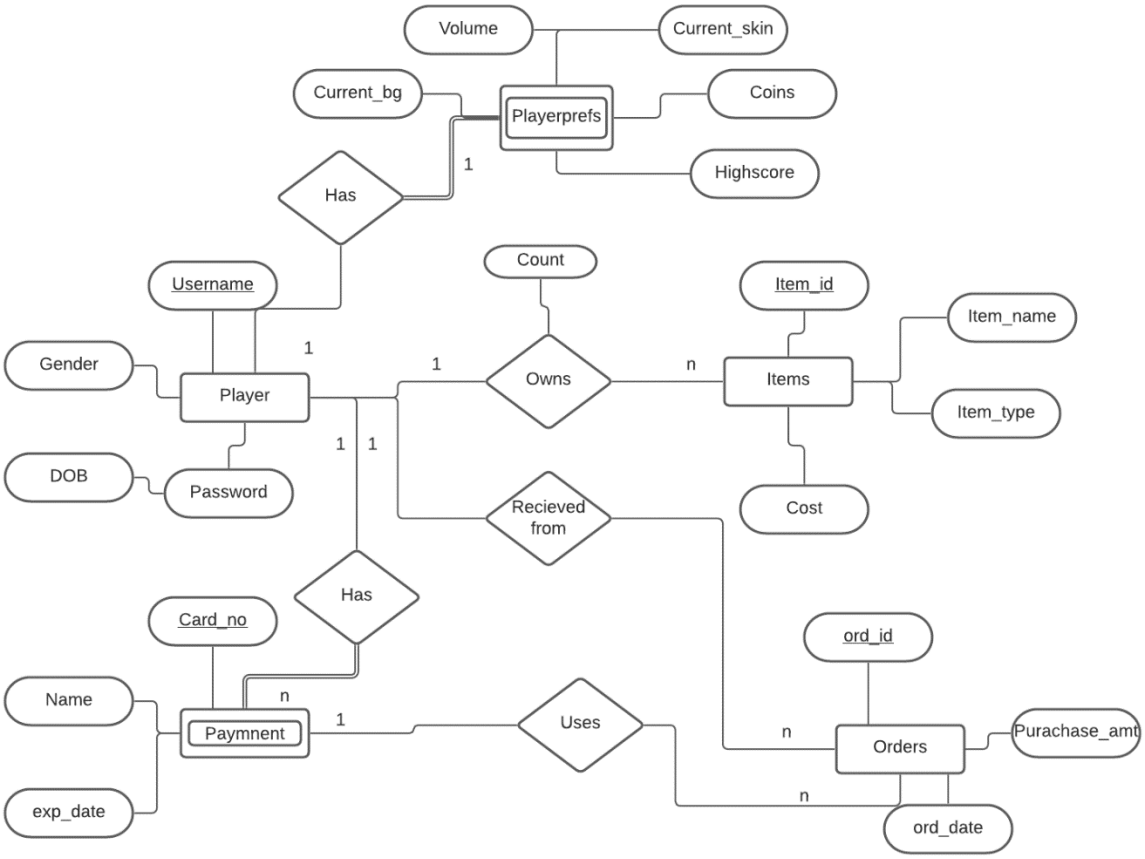


Figure 3.1 ER Diagram

3.2 SCHEMA DIAGRAM

The Schema of the project is shown in Figure 3.2

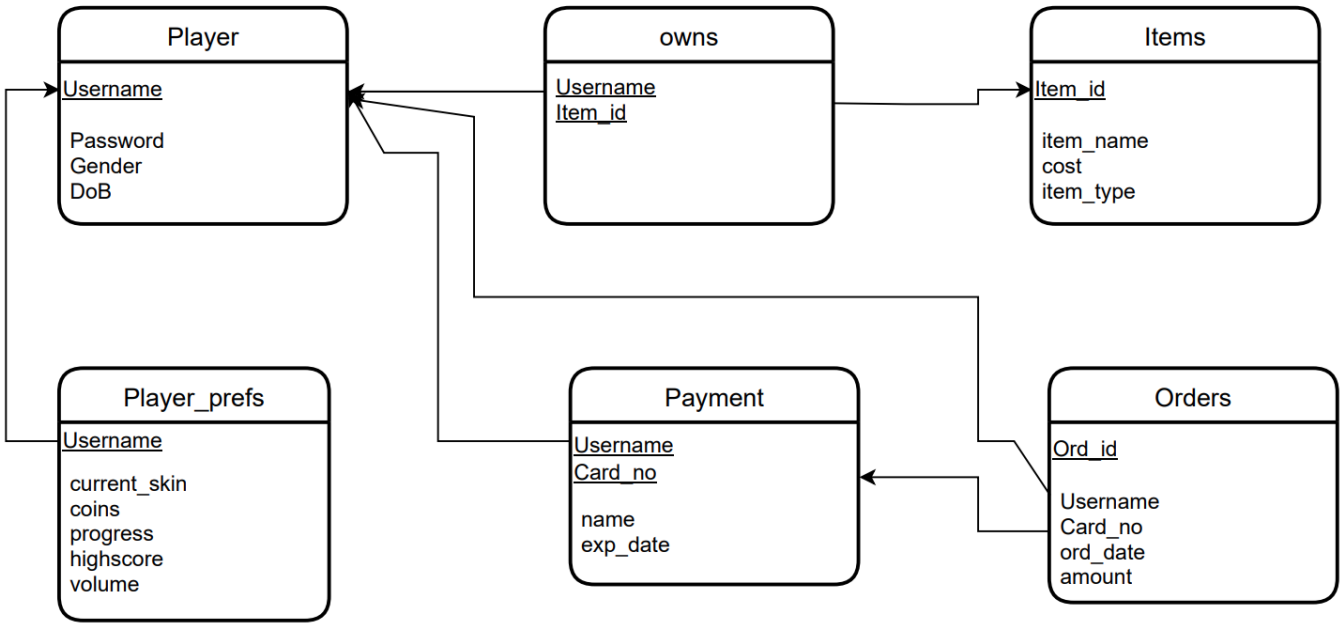


Figure 3.2 Schema Diagram

Chapter 4

IMPLEMENTATION

The 2D Platformer game allows the user a wide variety of functions include:

1.Login - The user can login with his/her credentials if he/she has previously registered on the application.

2.Register - A new user can register his/her profile on the application providing a suitable username, Date of birth, gender and a password.

3.Shop - The user can shop a wide variety of items that includes various characters skins and backgrounds.

4.Options - The user can customize his/her playing experience by going into the options and adjusting the required settings.

5.Buy Coins - The user van purchase the in-game currency of coins using real money and can make a transaction through his/her credit/debit card.

6.Payments - The payments page can be used to add or store the various payments methods that the user has for easier and faster checkout.

7.Orders - The user can look at the previous purchase history and order details that have been previously transacted by him/her.

8.Play – The user can play the various levels that the game offers by choosing to play the game. The game consists of many puzzles and quizzes that are used to progress further in the game. The questions are course related and a cheat sheet is provided before the start of each level for brushing up the required topics coming up in the level. The game consists of many interactable tiles and objects. Coins can be collected for buying in-game items.

Buying Coins

```
public class BuyCoinsPanel : MonoBehaviour
{
    SqliteDatabase DB;
    public Converter C;
    public TMP_Dropdown TD;
    public Slider slider;
    int[] amounts = { 10, 20, 35, 50, 100 };
    int[] coins = { 100, 500, 1000, 2000, 5000 };
    List<string> cards = new List<string>();
    int flag = 0;

    public TextMeshProUGUI alert;
    void Awake()
    {
        DB = new SqliteDatabase("GameDB.db");
        C = new Converter();
        FetchSavedCards();
    }
    void Start()
```

```
{
    FetchSavedCards();
}
public void FetchSavedCards()
{
    string query = C.SelectTable("payment", PlayerPrefs.GetString("username"));
    DataTable res = DB.ExecuteQuery(query);
    foreach (DataRow row in res.Rows)
    {
        cards.Add(row["card_no"].ToString());
    }
    if (cards.Count > 0)
    {
        flag = 1;
        TD.ClearOptions();
        TD.AddOptions(cards);
    }
}
public void Buy()
{
    if (flag == 1)
```

```
        {
            int i = (int)slider.value;
            PlayerPrefs.SetInt("coins", PlayerPrefs.GetInt("coins", 0) + coins[i]);
            string query = "Update playerprefs set coins = " + PlayerPrefs.GetInt("coins", 0) + "
where username = \"" + PlayerPrefs.GetString("username", "user") + "\"";
            Debug.Log(query);
            DB.ExecuteNonQuery(query);
            CreateOrdersEntry(i);
        }
        else{
            alert.text = "U need to add a valid payment method first";
        }

    }
    public void CreateOrdersEntry(int i)
    {
        int ord_id = 0;
        string query = "select max(ord_id) from orders";
        DataTable res = DB.ExecuteQuery(query);

        if (res.Rows[0]["max(ord_id)"] != null)
        {
```

```
            ord_id = Int32.Parse(res.Rows[0]["max(ord_id)"].ToString())+1;
        }


        Debug.Log(ord_id);
        string date = System.DateTime.Now.ToString("yyyy-MM-dd");
        query = C.InsertTable("orders", new string[] { PlayerPrefs.GetString("username",
"user"), TD.options[TD.value].text, ord_id.ToString(), date, amounts[i].ToString() });
        try{
            DB.ExecuteNonQuery(query);
        }
        catch(Exception E)
        {
            Debug.Log(E);
        }


    }
}
```

Converter Helper Script

```
public class Converter
{
    public string InsertTable(string tablename, string[] values)
    {
```

```
        string query = "insert into " + tablename + " values ( ";
        for(int i=0;i<values.Length;i++)
        {
            string val = values[i];
            query += "\""+val+"\"";
            if(i!=values.Length-1)
            {
                query+=",";
            }
        }

        query+=");";
        return query;
    }
    public string UpdateTable(string tablename, string newvaltag, string newval, string oldvaltag, string oldval)
    {
        string query = "update table " + tablename + " set " + newvaltag +" = "+"\""+ newval+"\"" + " where " + oldvaltag +" = " +"\"" + oldval+"\"";
        query+=";";
        return query;
    }
```

```
    public string SUpdateTable(string tablename, string newvaltag, string newval, string
username)
    {
        string query = "update " + tablename + " set " + newvaltag +" = "+"\""+ newval+"\"" + "
where username  = " +"\"" + username+"\"";
        query+=";";
        return query;
    }
    public string SelectTable(string tablename, string username)
    {
        string query = "select * from " + tablename +" where username  = " +"\""
+username+"\"";
        query+=";";
        return query;
    }
}
```

Orders

```
public class Orders : MonoBehaviour
{
    SqliteDatabase DB;
    public Converter C;
    public GameObject cardPrefab;
```

```
public GameObject content;

void Awake()

{

    DB = new SqliteDatabase("GameDB.db");

    C = new Converter();

    FetchOrders();

}

void Start()

{

    FetchOrders();

}

void FetchOrders()

{

    int childs = content.transform.childCount;

    for (int i = childs - 1; i >= 0; i--)

    {

        GameObject.Destroy(content.transform.GetChild(i).gameObject);

    }

    string query = C.SelectTable("orders", PlayerPrefs.GetString("username"));

    DataTable res = DB.ExecuteQuery(query);

    foreach (DataRow row in res.Rows)
```

```
        {
            GameObject card = Instantiate(cardPrefab) as GameObject;
            card.SetActive(true);
            card.transform.GetChild(0).GetComponent<TextMeshProUGUI>().text =
row["ord_id"].ToString();
            card.transform.GetChild(1).GetComponent<TextMeshProUGUI>().text =
row["card_no"].ToString();
            card.transform.GetChild(2).GetComponent<TextMeshProUGUI>().text =
row["ord_date"].ToString();
            card.transform.GetChild(3).GetComponent<TextMeshProUGUI>().text =
row["amount"].ToString();
            card.transform.SetParent(content.transform, false);
        }
    }
}
```

Payment

```
public class Payment : MonoBehaviour
{
    // Start is called before the first frame update
    SqliteDatabase DB;
    public TMP_InputField cardNo;
    public TMP_InputField cardName;
    public TMP_InputField expdate;
```

```
    public GameObject cardPrefab;


    public GameObject content;

    public Converter C;

    void Awake()

    {

        DB = new SqliteDatabase("GameDB.db");

        C = new Converter();

        FetchSavedCards();

    }


    public void SaveCard()

    {

        try

        {

            string query = C.InsertTable("payment", new string[] {
PlayerPrefs.GetString("username"), cardNo.text, cardName.text, expdate.text });

            Debug.Log(query);

            DB.ExecuteNonQuery(query);

            FetchSavedCards();

        }

        catch (Exception e)
```

```
        {
            Debug.Log(e);
            Debug.Log("error");
        }
    }
    public void FetchSavedCards()
    {
        // for (int i = 0; i < 10; i++)
        // {
        //     GameObject card = Instantiate(cardPrefab) as GameObject;
        //     card.SetActive(true);
        //     card.transform.GetChild(0).GetComponent<TextMeshProUGUI>().text =
"11234654";
        //     card.transform.GetChild(1).GetComponent<TextMeshProUGUI>().text =
"11234654";
        //     card.transform.GetChild(2).GetComponent<TextMeshProUGUI>().text =
"11234654";
        //     card.transform.SetParent(content.transform, false);
        // }
        int childs = content.transform.childCount;
        for (int i = childs - 1; i >= 0; i--)
        {
```

```
            GameObject.Destroy(content.transform.GetChild(i).gameObject);
        }
        string query = C.SelectTable("payment", PlayerPrefs.GetString("username"));
        DataTable res = DB.ExecuteQuery(query);
        foreach (DataRow row in res.Rows)
        {
            GameObject card = Instantiate(cardPrefab) as GameObject;
            card.SetActive(true);
            card.transform.GetChild(0).GetComponent<TextMeshProUGUI>().text =
row["card_no"].ToString();
            card.transform.GetChild(1).GetComponent<TextMeshProUGUI>().text =
row["name"].ToString();
            card.transform.GetChild(2).GetComponent<TextMeshProUGUI>().text =
row["exp_date"].ToString();
            card.transform.SetParent(content.transform, false);
        }
    }
}
```

ShopManager

```
public class ShopManager : MonoBehaviour
{
    int coins;
```

```
SqliteDatabase DB;

public TextMeshProUGUI coinsText;

int[] cost = { 0, 1000, 2000, 3000, 4000, 5000, 0, 1000, 2000, 3000, 4000, 5000 };

int[] items;

public TextMeshProUGUI[] texts;

public int currentSkin = 0;

public int currentBackground = 0;

Converter C = new Converter();

public TextMeshProUGUI prompt;

void Awake()

{

    DB = new SqliteDatabase("GameDB.db");

    currentSkin = PlayerPrefs.GetInt("current_skin", 0);

    currentBackground = PlayerPrefs.GetInt("current_bg", 0);

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

    coinsText.text = coins.ToString();

    items = new int[12];

    for (int i = 0; i < 12; i++)

        items[i] = 0;


    FetchItems();
```

```
        for (int i = 0; i < 12; i++)
            if (items[i] == 1)
            {
                texts[i].text = "Equip";
            }
    }

    // Update is called once per frame
    void FetchItems()
    {
        string query = C.SelectTable("owns", PlayerPrefs.GetString("username"));
        Debug.Log(query);
        DataTable dt = DB.ExecuteQuery(query);
        foreach (DataRow row in dt.Rows)
        {
            items[Int32.Parse(row["item_id"].ToString())] = 1;
        }
    }
    public void Buy(int i)
    {
        if (items[i] == 1)
```

```
        {
            if (i < 6)
            {
                currentSkin = i;
                prompt.text = "Skin " + i + " Equipped";
                PlayerPrefs.SetInt("current_skin", i);
                UpdateDBskin();
            }
            else
            {
                currentBackground = i;
                prompt.text = "Background " + i + " Equipped";
                PlayerPrefs.SetInt("current_bg", i);
                UpdateDBbg();

            }

        }
        else if (coins >= cost[i])
        {
            coins -= cost[i];
```

```
        PlayerPrefs.SetInt("coins", coins);

        string query = "Update playerprefs set coins = " + coins + " where username = \"" +
PlayerPrefs.GetString("username", "user") + "\"";

        Debug.Log(query);

        DB.ExecuteNonQuery(query);

        items[i] = 1;

        coinsText.text = coins.ToString();

        texts[i].text = "Equip";

        if (i < 6)

        {

           prompt.text = "Skin " + i + "  Bought and Equipped";

           PlayerPrefs.SetInt("current_skin", i);

        }

        else

        {

           prompt.text = "Baackground " + i + "  Bought and Equipped";

           PlayerPrefs.SetInt("current_bg", i);

        }

        UpdateDBowns(i);

    }

    else

    {
```

```
            prompt.text = "Not Enough Coins";
        }
    }
    void UpdateDBskin()
    {
        string query = C.SUpdateTable("playerprefs", "current_skin",
PlayerPrefs.GetInt("current_skin").ToString(), PlayerPrefs.GetString("username"));
        Debug.Log(query);
        DB.ExecuteNonQuery(query);
    }
    void UpdateDBbg()
    {
        string query = C.SUpdateTable("playerprefs", "current_bg",
PlayerPrefs.GetInt("current_bg").ToString(), PlayerPrefs.GetString("username"));
        Debug.Log(query);
        DB.ExecuteNonQuery(query);
    }
    void UpdateDBowns(int item)
    {
        string query = C.InsertTable("owns", new string[] { PlayerPrefs.GetString("username"),
item.ToString() });
        Debug.Log(query);
```

```
        DB.ExecuteNonQuery(query);

    }

}
```

Chapter 5

DISCUSSIONS AND SCREENSHOTS

Fig 5.1 Registration Menu

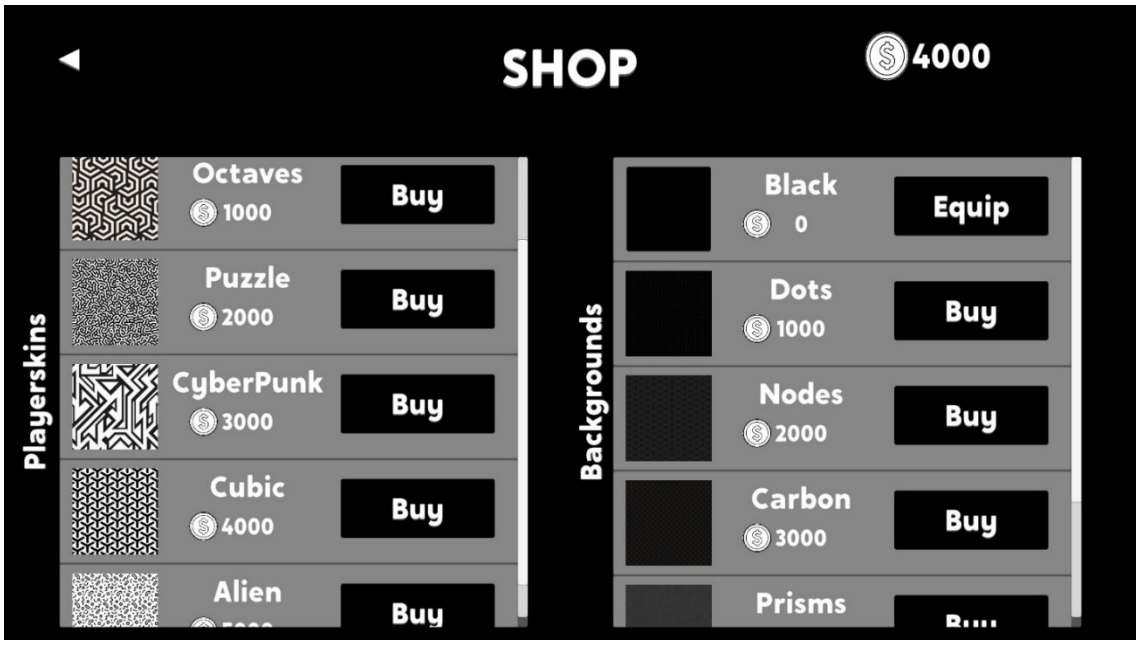


Fig 5.2 Shopping Portal

Fig 5.3 Main Menu

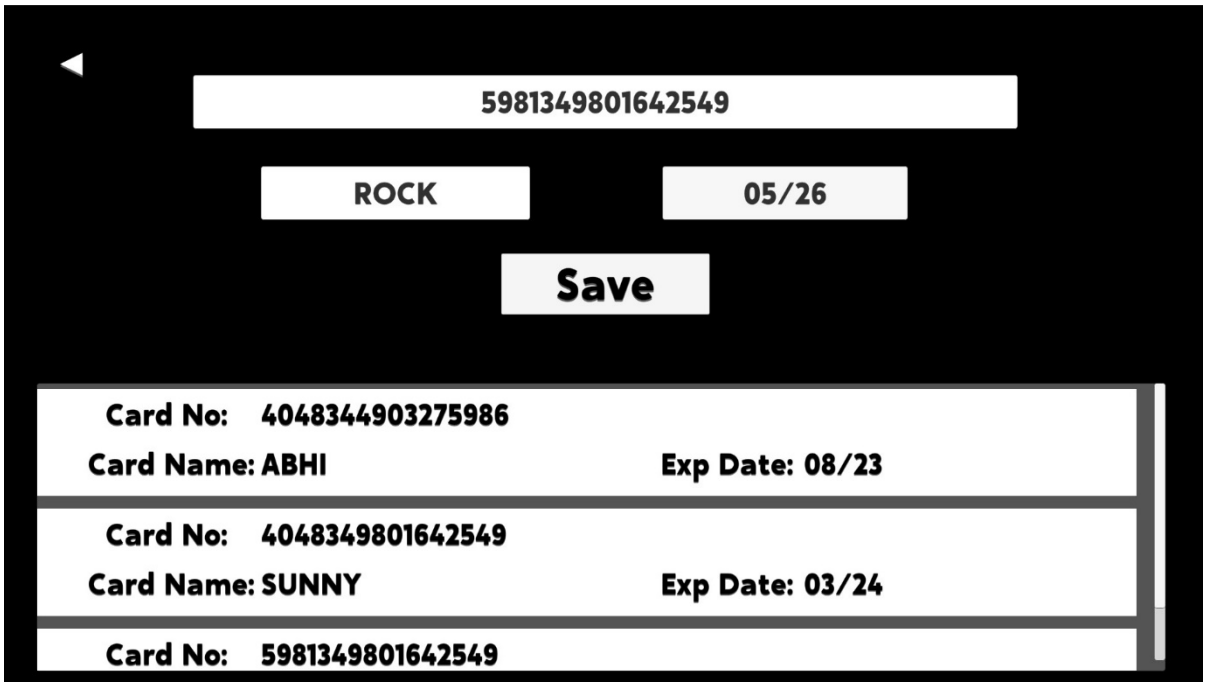


Fig 5.4 Payment Portal

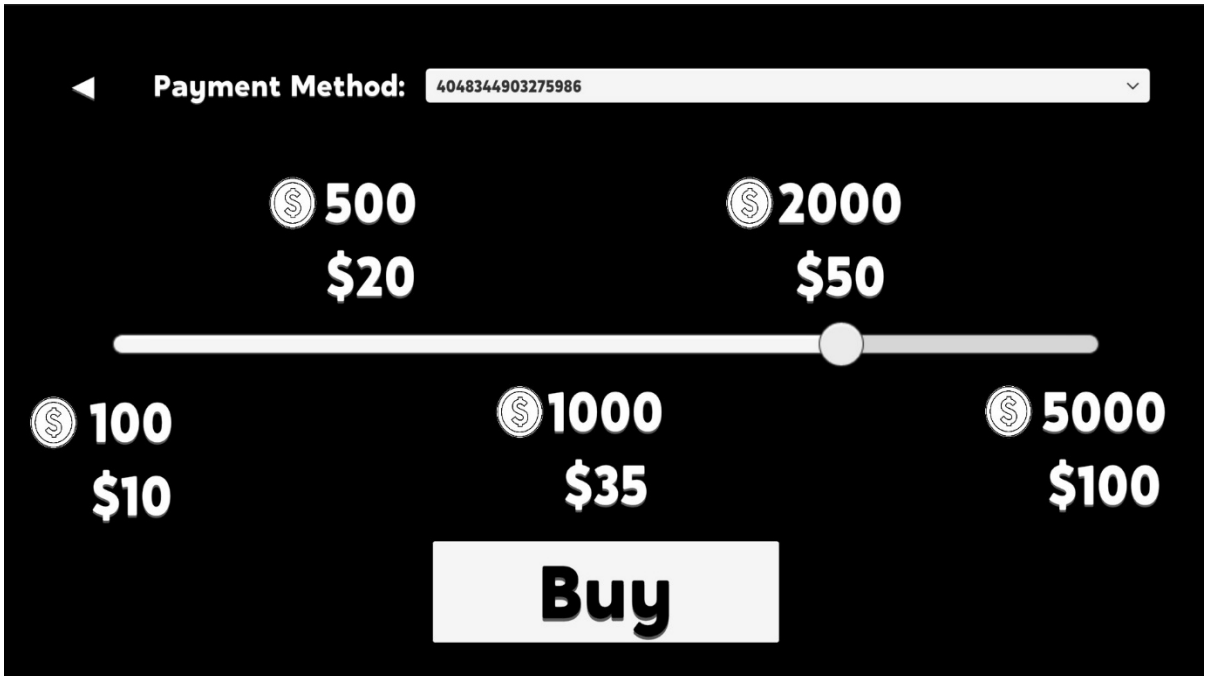


Fig 5.5 Purchase Portal

Fig 5.6 View of all Orders

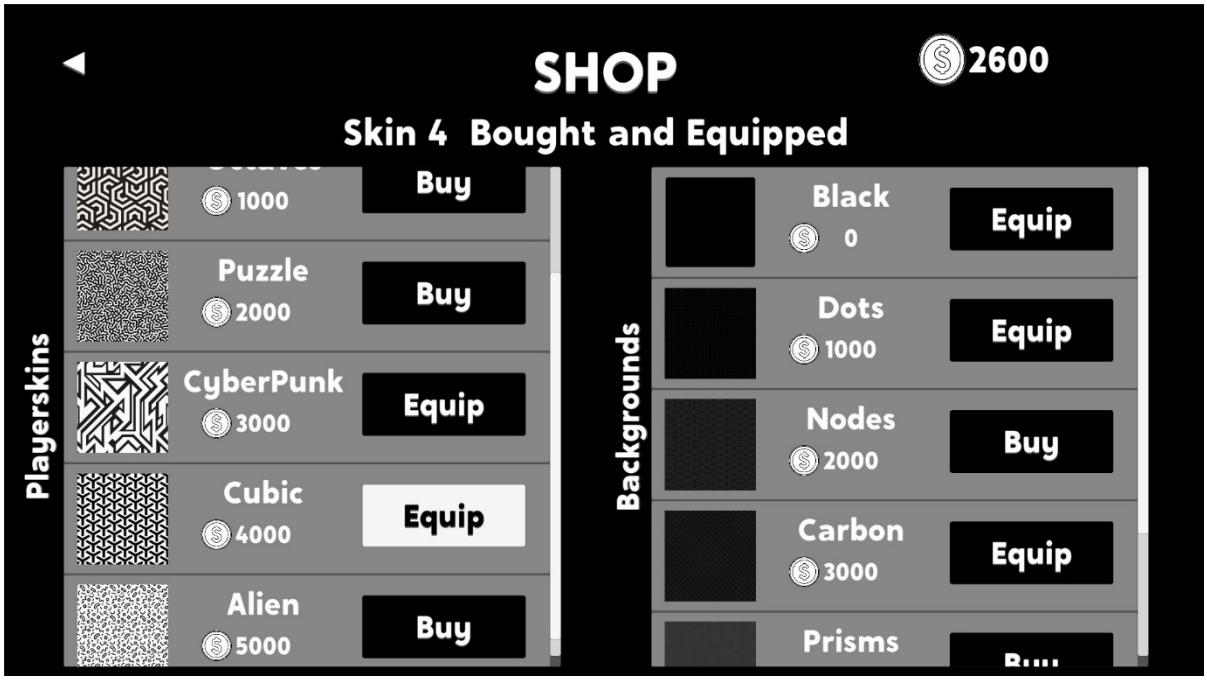


Fig 5.7 Shopping Menu

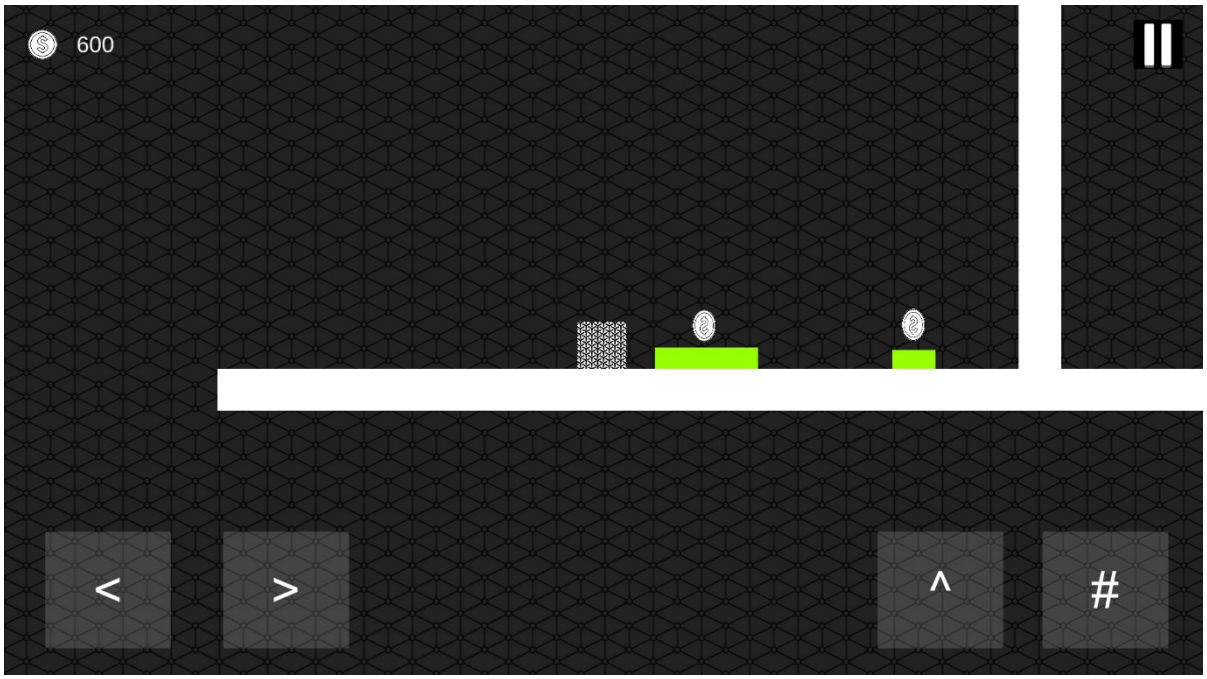


Fig 5.8 Gameplay Instance

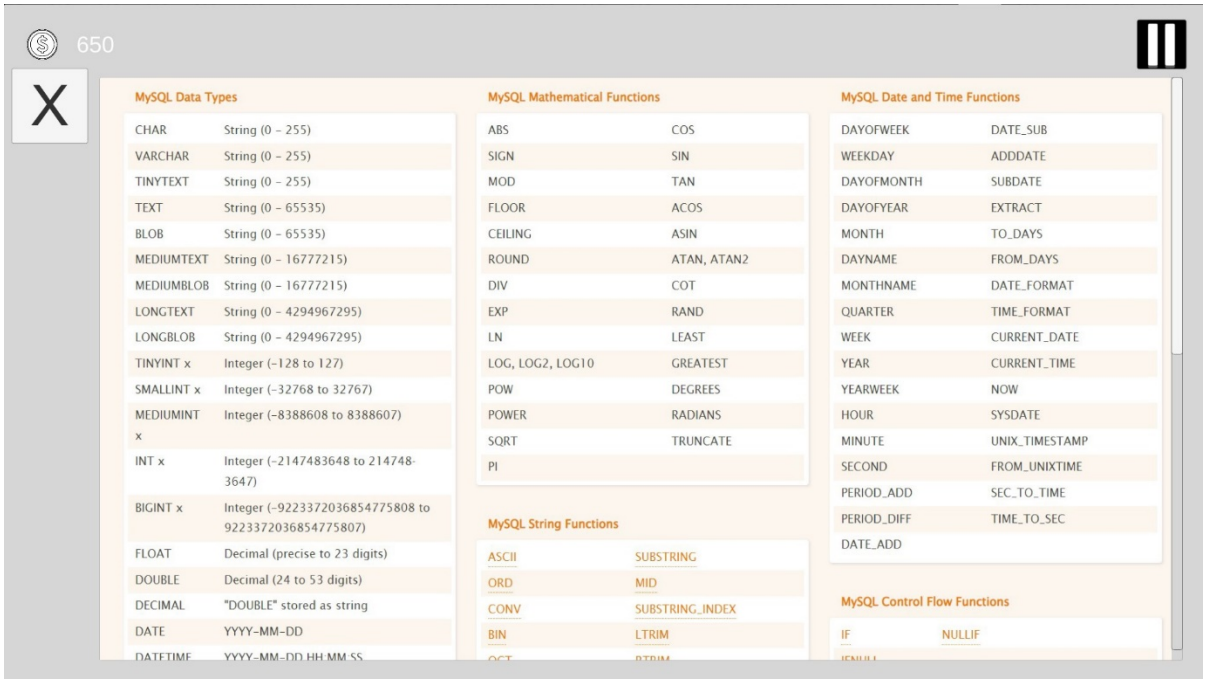


Fig 5.9 Gameplay Instance

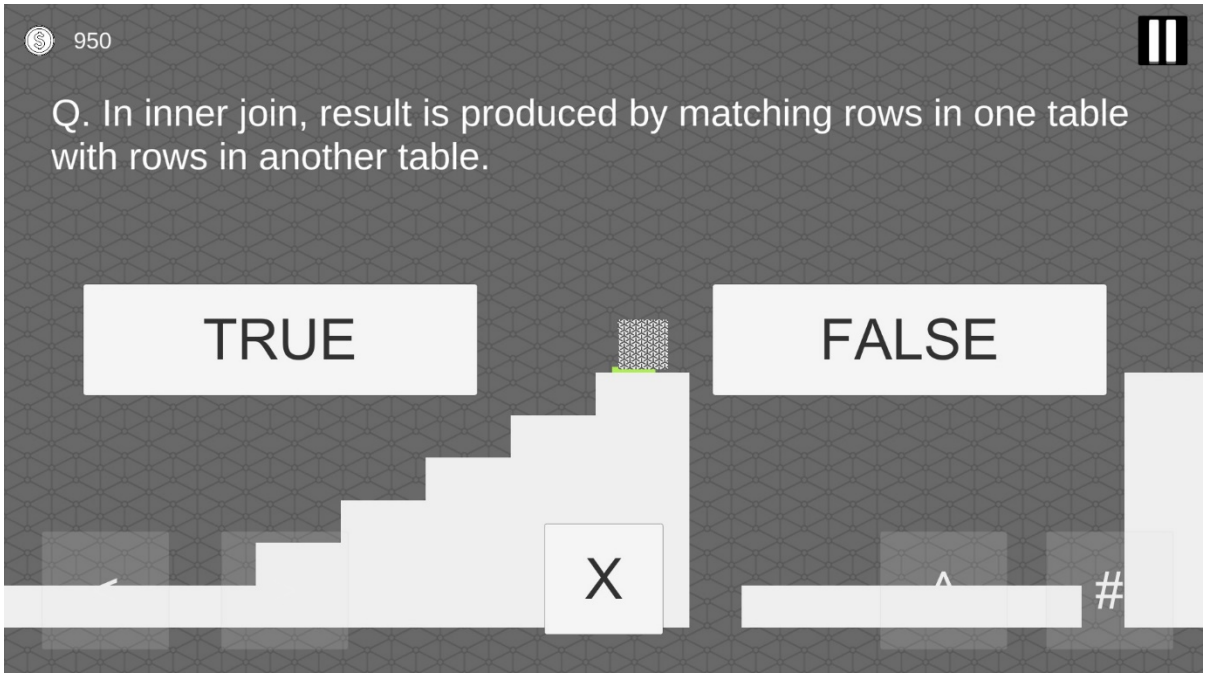


Fig 5.10 Gameplay Instance

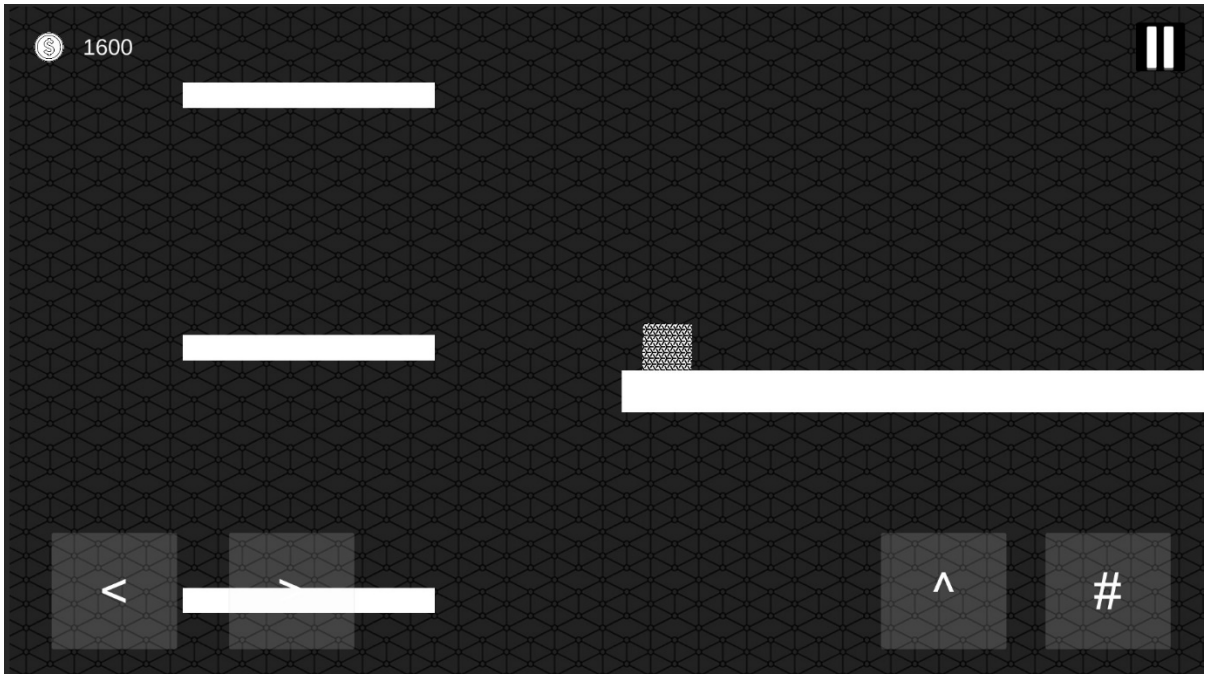


Fig 5.11 Gameplay Instance

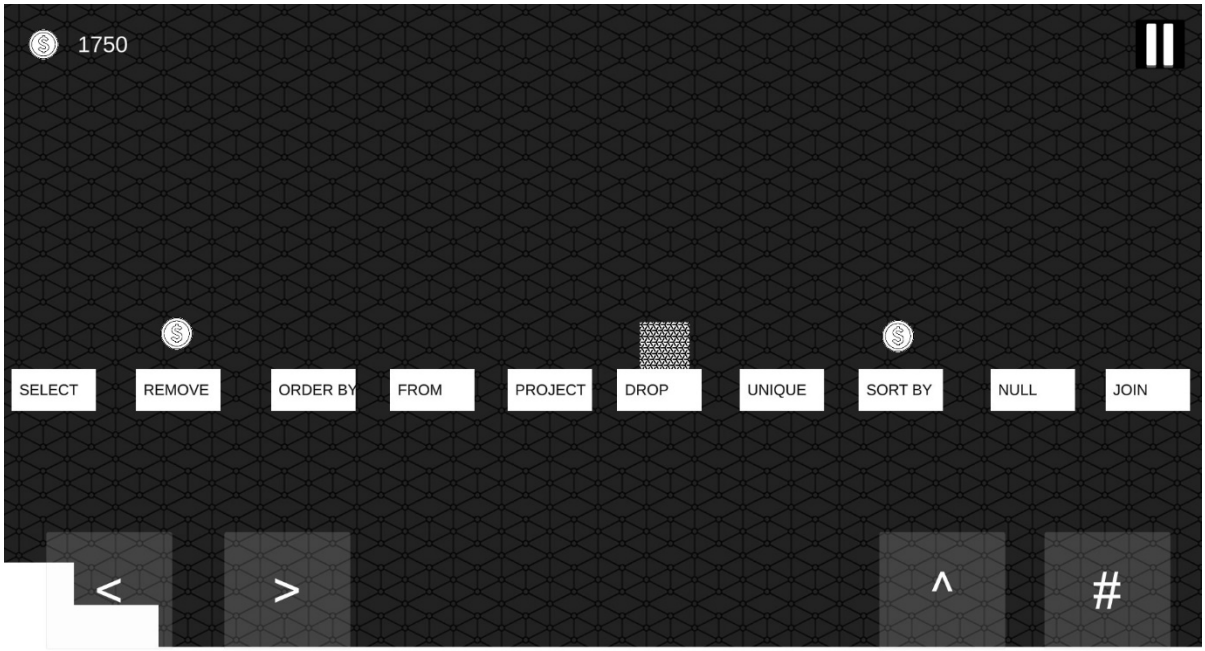


Fig 5.12 Gameplay Instance

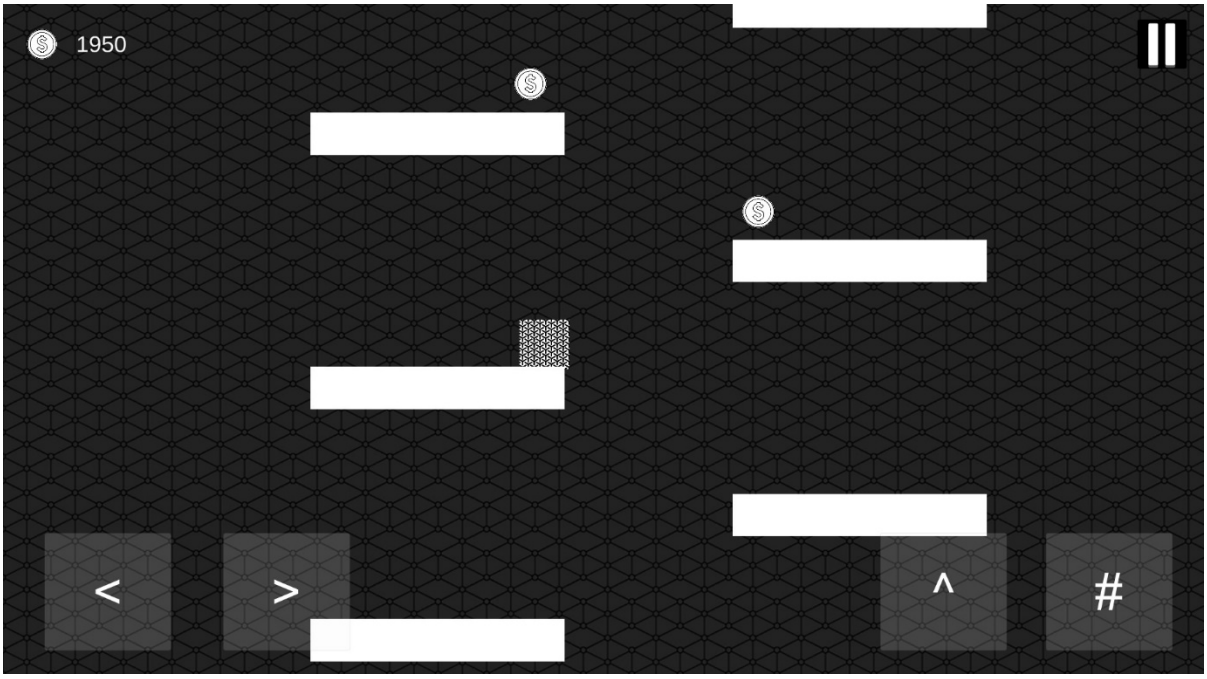


Fig 5.13 Gameplay Instance

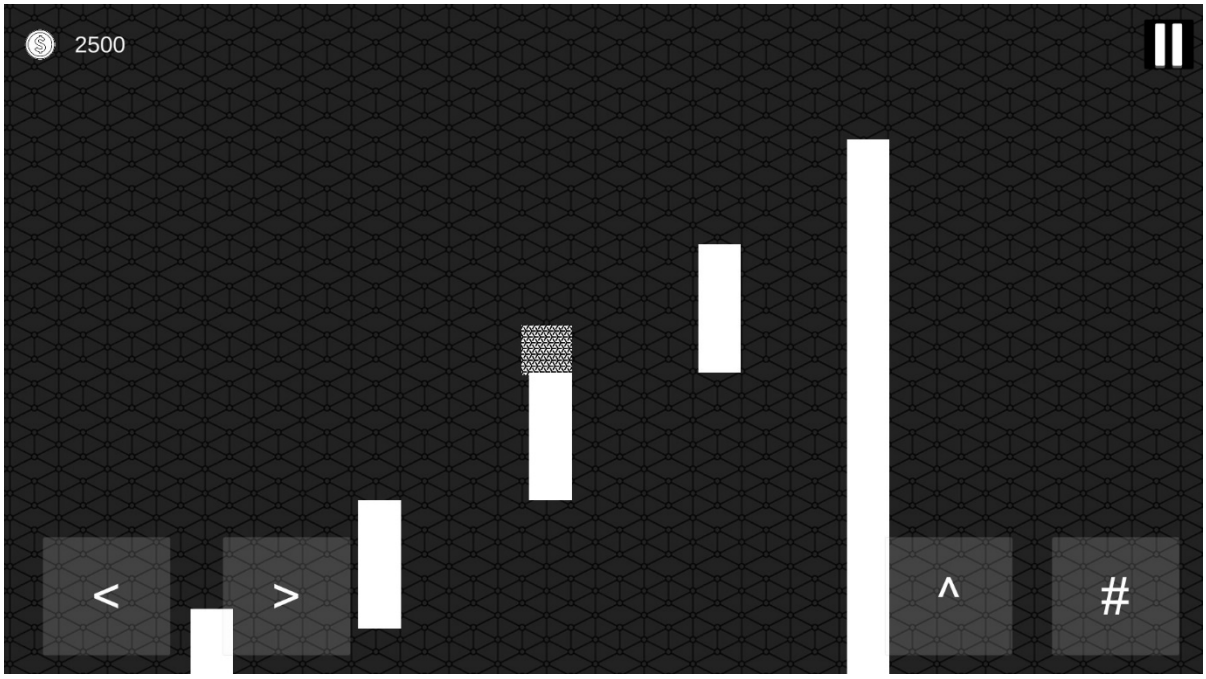


Fig 5.14 Gameplay Instance

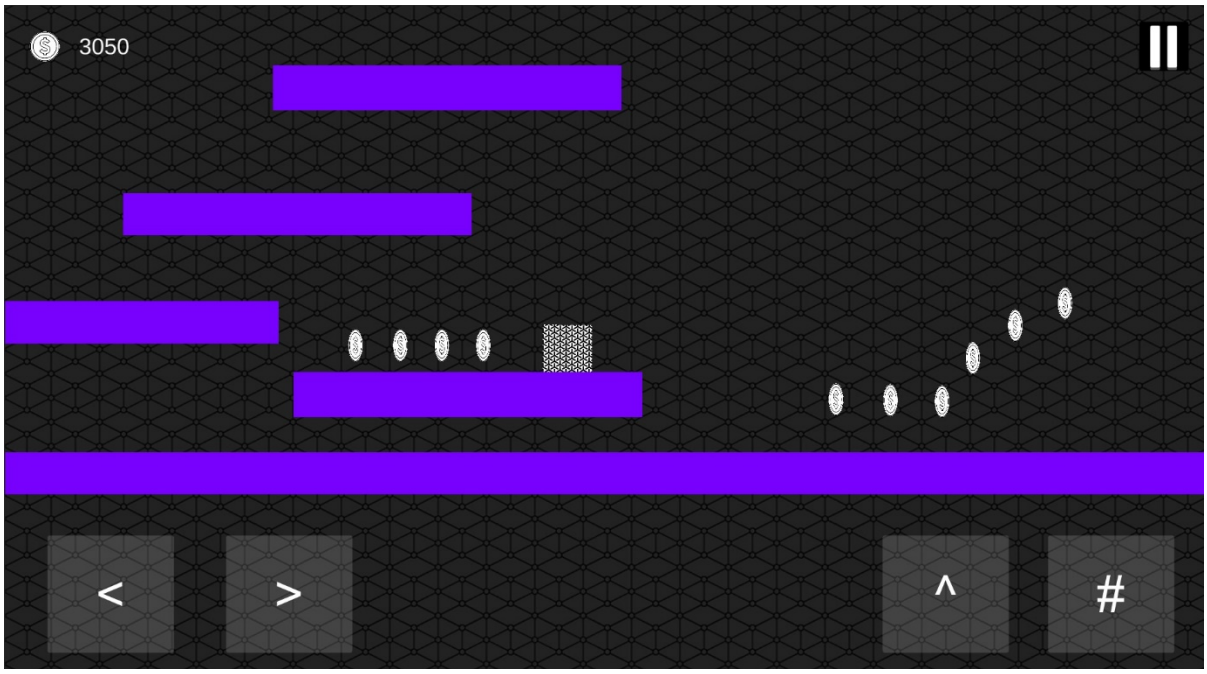


Fig 5.15 Gameplay Instance

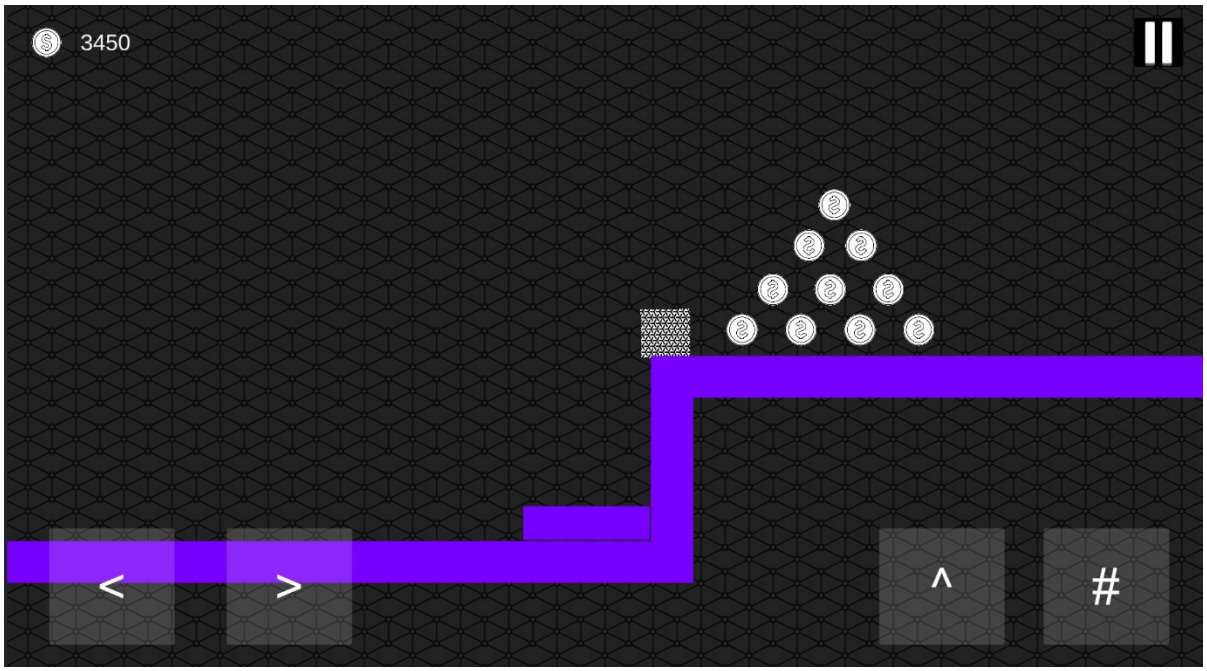


Fig 5.16 Gameplay Instance

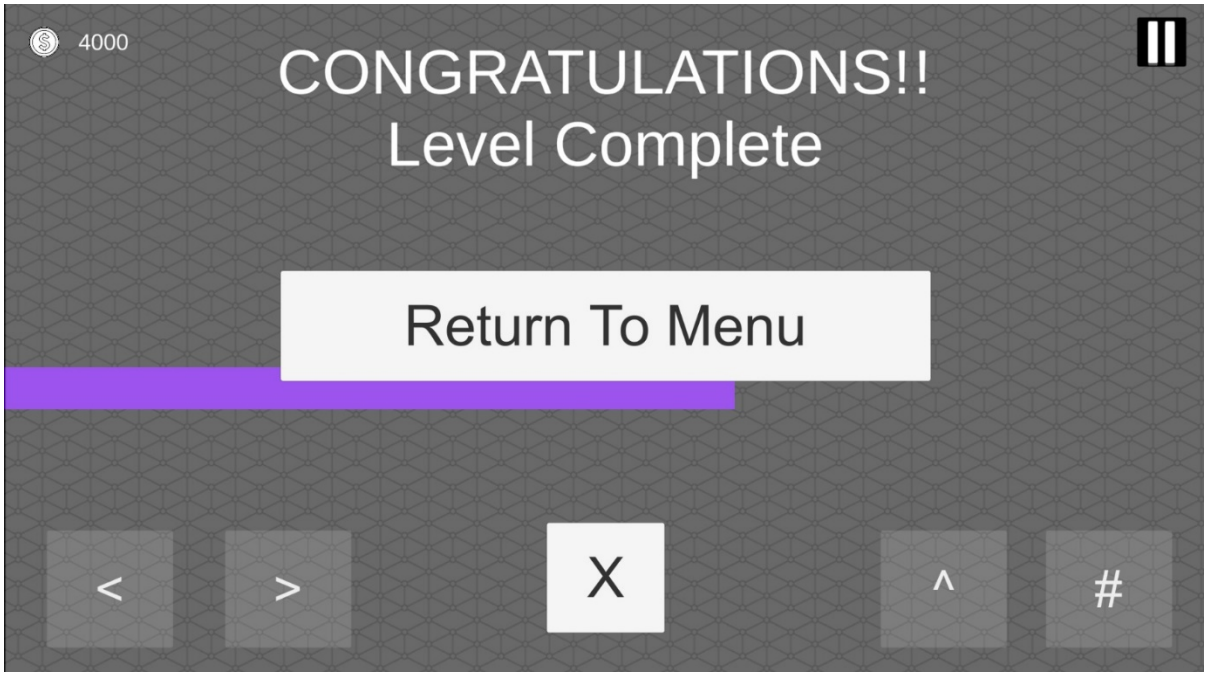


Fig 5.17 Gameplay Instance

Fig 5.18 Gameplay Instance

Chapter 6

CONCLUSION AND FUTURE SCOPE

The project on Database Management System allowed us to think creatively and develop our own application using the languages C#, SQL and Technologies like Unity and MySQL.

Future Scope:

i. Multiplayer Control
ii. Expanding the shop items
iii. Adding more courses and quizzes
iv. Making the UI more Playful and Interactive.

Chapter 7

BIBILOIGRAPHY

[1] https://github.com/Busta117/SQLiteUnityKit

[2] https://docs.unity3d.com/Manual/index.html

[3] https://www.sqlite.org/docs.html

[4] https://stackoverflow.com/