UTM

UNIVERSITI TEKNOLOGI MALAYSIA

Group Assignment

SECJ2013-04 STRUKTUR DATA DAN ALGORITMA
(DATA STRUCTURE AND ALGORITHM)

Dr Norsham binti Idris

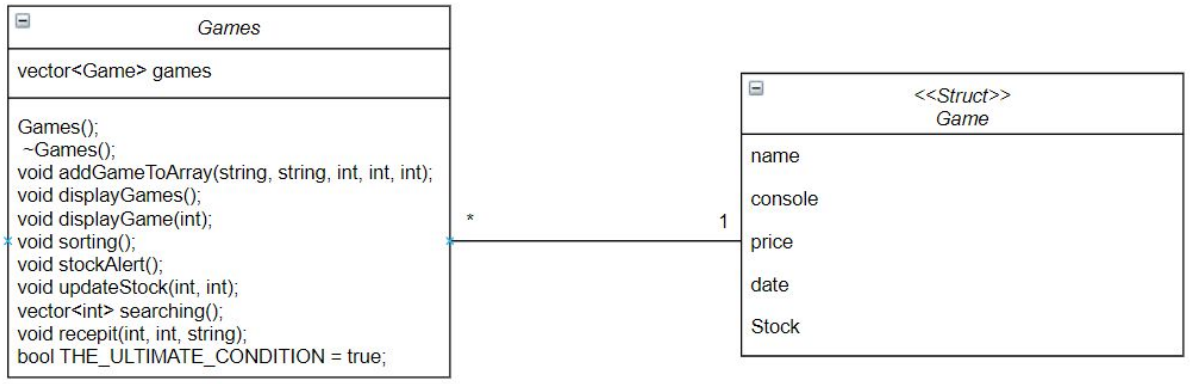
	Name	Matric
1	ABDULRAHMAN MOHAMMED AQEL ASSAGGAF	A18CS4054
2	ABDULAZIZ ABDULRAHMAN AWAD BA HAJ	A18CS4057
3	ABDALLAH MALAM MODU USMAN	A18CS4027
4	SHAMEMA AFRUZA JOUTY	A19EC4028

TABLE OF CONTENT

Group member’s tasks

	Name	Matric	Tasks
1	ABDULRAHMAN MOHAMMED AQEL ASSAGGAF	A18CS4054	Planning, Class diagram, Writing code.
2	ABDULAZIZ ABDULRAHMAN AWAD BA HAJ	A18CS4057	Planning, Class diagram, Writing code.
3	ABDALLAH MALAM MODU USMAN	A18CS4027	Writing assignment report, Writing code.
4	SHAMEMA AFRUZA JOUTY	A19EC4028	Writing code.

Class diagram of the system



Source code documentation (source code with comments)

Game.hpp

```
#include <iostream>
#include <string>
#include <vector>
using namespace std;

struct Game
{
    string name;
    string consoleName;
    int price;
    int date;
    int stock;
};

class Games
{
private:
    vector<Game> games;

public:
    Games();
    ~Games();
    void addGameToArray(string, string, int, int, int);
    void displayGames();    // display all the games
    void displayGame(int); // display one game
    void sorting();         //selection sorting
    void stockAlert();
    void updateStock(int, int);
    vector<int> searching(); //sequential searching
    void recepit(int, int, string);
    bool THE_ULTIMATE_CONDITION = true;
};
```

Game.cpp

```
#include "game.hpp"

//empty constructor
Games::Games()
{
}

Games::~Games()
{
    // delete games;
}
void Games::addGameToArray(string _name, string _consoleName, int
_price, int _date, int _stock)
{
    games.push_back(Game{_name, _consoleName, _price, _date, _stock});
} //add Game
void Games::displayGames()
{
    cout << "Game\t\tConsole\t\tPrice\t\tDate\t\tstock" << endl;
    cout <<
"=========================================================================="
<< endl;
    for (size_t i = 0; i < games.size(); i++)
    {
        cout << games[i].name << "\t\t" << games[i].consoleName <<
"\t\t" << games[i].price << "\t\t" << games[i].date << "\t\t" <<
games[i].stock << endl;
    }
    cout << endl;
} // display all games

void Games::displayGame(int index)
{
    cout << "Game\t\tConsole\t\tPrice\t\tDate\t\tstock" << endl;
    cout <<
"=========================================================================="
<< endl;
    cout << games[index].name << "\t\t" << games[index].consoleName <<
"\t\t" << games[index].price << "\t\t" << games[index].date << "\t\t"
<< games[index].stock << endl
         << endl;
```

```
} //display one game

void Games::sorting()
{
    int swp_index = 0; //tracks smallest found value
    Game temp;
    int type;
    cout << "<< choose from the list >> :"
         << endl
         << endl
         << "1. Sort By Name"
         << endl
         << "2. Sort By Concole Name"
         << endl
         << "3. Sort By Price"
         << endl
         << "4. Sort By Date"
         << endl
         << "5. Sort By Stock"
         << endl
         << endl;
    cout << "enter your type : ";
    cin >> type;
    cout << endl;
    for (int a = 0; a < games.size() - 1; a++)
    {
        swp_index = a;
        for (int index = a + 1; index < games.size(); index++)
        {
            switch (type)
            {
            case 1:
                if (games[index].name < games[swp_index].name)
                {
                    swp_index = index;
                }
                break;
            case 2:
                if (games[index].consoleName <
games[swp_index].consoleName)
                {
                    swp_index = index;
                }
```

```
                break;
            case 3:
                if (games[index].price < games[swp_index].price)
                {
                    swp_index = index;
                }
                break;
            case 4:
                if (games[index].date < games[swp_index].date)
                {
                    swp_index = index;
                }
                break;
            case 5:
                if (games[index].stock < games[swp_index].stock)
                {
                    swp_index = index;
                }
                break;
            }
        }

        temp = games[a];
        games[a] = games[swp_index];
        games[swp_index] = temp;
    }
} //sorting

void Games::stockAlert()
{
    int value = 10;
    int flag = 0;
    for (size_t i = 0; i < games.size(); i++)
    {
        if (games[i].stock <= value)
        {
            cout << "This game has less than " << value << " in stock "
<< endl
                 << endl;
            displayGame(i);

            flag++;
        }
```

```
    }
    if (flag == 0)
    {
        cout << "All the product has more than 10 in the stock !" <<
endl
             << endl;
    }
} //socket aleart

void Games::updateStock(int index, int num)
{
    if (num > 0)
    {
        games[index].stock = games[index].stock + num;
        cout << "Done!!" << endl
             << endl;
    }
    else if (num < 0)
    {
        num = num * -1;
        if (games[index].stock < num)
        {
            cout << "Stock not enough!!" << endl
                 << endl;
        }
        else
        {
            games[index].stock = games[index].stock - num;
            cout << "Done!!" << endl
                 << endl;
            THE_ULTIMATE_CONDITION = false;
        }
    }

} //updateStock

vector<int> Games::searching()
{
    int choice;
    string stringData;
    int numData;
    int flag = 0;
    vector<int> indexFound;
```

```
    cout << "<< choose from the list >> :"
         << endl
         << endl
         << "1. searching By Name"
         << endl
         << "2. searching By Concole Name"
         << endl
         << "3. searching By Price"
         << endl
         << "4. searching By Date"
         << endl
         << "5. searching By Stock"
         << endl
         << "6. Back"
         << endl
         << endl;
    cout << "enter your type : ";
    cin >> choice;
    cout << endl
         << endl;

    if (choice == 1 || choice == 2)
    {
        cout << "Entet the name you are searching for: ";
        cin >> stringData;
        cout << endl;
    }
    if (choice == 3 || choice == 4 || choice == 5)
    {
        cout << "Entet the number you are searching for: ";
        cin >> numData;
        cout << endl;
    }
    if (choice == 6)
    {
        indexFound.push_back(-1);
        return indexFound;
    }

    for (size_t i = 0; i < games.size(); i++)
    {
        switch (choice)
        {
```

```
        case 1:
            if (games[i].name == stringData)
            {
                indexFound.push_back(i);
                flag++;
            }
            break;
        case 2:
            if (games[i].consoleName == stringData)
            {
                indexFound.push_back(i);
                flag++;
            }
            break;
        case 3:
            if (games[i].price == numData)
            {
                indexFound.push_back(i);
                flag++;
            }
            break;
        case 4:
            if (games[i].date == numData)
            {
                indexFound.push_back(i);
                flag++;
            }
            break;
        case 5:
            if (games[i].stock == numData)
            {
                indexFound.push_back(i);
                flag++;
            }
            break;
        default:
            cout << "Wrong Output" << endl
                 << endl;
            break;
        }
    }
    if (flag == 0)
    {
```

```
        indexFound.push_back(-1);
        return indexFound;
    }

    //no match found
    return indexFound;
} // searching

void Games::recepit(int index, int num, string customerName)
{
    cout << "Here is your receipt" << endl;
    cout << "Name: " << customerName << endl;
    cout << "Game Name: " << games[index].name << endl;
    cout << "Game Console: " << games[index].consoleName << endl;
    cout << "Price: " << games[index].price * num << "RM" << endl;
    cout << num << " Copies" << endl;
    cout << "Thank You for shopping with us :) Vist us again!" << endl
         << endl;
}
```

main.cpp

```
#include <iostream>
#include <string>
#include <vector>
#include <fstream>
#include "game.hpp"
using namespace std;

void startMenue();
void ownerMenue();
void coustomerMenue();

int main()
{
    ifstream myReadFile;
    //games object
    Games games;
    //open the file again to get the data
    myReadFile.open("data.txt");

    //reding the data from txt file
    if (myReadFile.is_open())
```

```
    {
        string name, consoleName;
        int price, stock, date;
        while (!myReadFile.eof())
        {
            myReadFile >> name;
            myReadFile >> consoleName;
            myReadFile >> price;
            myReadFile >> date;
            myReadFile >> stock;
            games.addGameToArray(name, consoleName, price, date,
stock);
        }
        int choice;
        while (choice != 3) //start menue
        {
            startMenue();
            cin >> choice;
            cout << endl;
            if (choice == 1) //owener menue
            {
                while (choice != 5)
                {
                    ownerMenue();
                    cin >> choice;
                    cout << endl;
                    if (choice > 5)
                    {
                        cout << "Invalid number, please enter again" <<
endl
                             << endl;
                    }
                    if (choice == 1)
                    {
                        games.sorting();
                    }
                    if (choice == 2)
                    {
                        vector<int> data;
                        data = games.searching();
                        int input, input2;
                        if (data[0] == -1)
                        {
```

```
                            cout << "No date has been found !!" << endl
                                 << endl;
                        }
                        else
                        {
                            for (size_t i = 0; i < data.size(); i++)
                            {
                                cout << "============" << i + 1 <<
"============" << endl;
                                games.displayGame(data[i]);
                            }
                            cout << "which game you want to update the
stock ? : ";
                            cin >> input;
                            cout << endl;
                            cout << " How many coppies You want to
order ? ";
                            cin >> input2;
                            cout << endl;
                            games.updateStock(data[input - 1], input2);
                        }
                    }
                    if (choice == 3)
                    {
                        games.stockAlert();
                    }
                    if (choice == 4)
                    {
                        games.displayGames();
                    }
                }
            }
            if (choice == 2) // coustomer menue
            {
                while (choice != 4)
                {
                    coustomerMenue();
                    cin >> choice;
                    cout << endl;
                    if (choice > 3)
                    {
                        cout << "Invalid number, please enter again" <<
endl
```

```
                            << endl;
                    }
                    if (choice == 1)
                    {
                        cout << "Choose the type of searching you want
" << endl;
                        vector<int> data;
                        data = games.searching();
                        int input, input2;
                        if (data[0] == -1)
                        {
                            cout << "No date has been found !!" << endl
                                 << endl;
                        }
                        else
                        {
                            for (size_t i = 0; i < data.size(); i++)
                            {
                                cout << "============" << i + 1 <<
"============" << endl;
                                games.displayGame(data[i]);
                            }
                            cout << "choose the game you want to Buy ?
: ";
                            cin >> input;
                            cout << endl;
                            do
                            {
                                cout << " How many coppies You want to
Buy ? ";
                                cin >> input2;
                                cout << endl;
                                games.updateStock(data[input] - 1,
input2 * -1);
                            } while (THE_ULTIMATE_CONDITION);

                            string customerName;
                            cout << "Enter your name please: ";
                            cin >> customerName;
                            cout << endl;
                            cout << endl;
                            games.recepit(data[input], input2,
customerName);
```

```
                        }
                    }
                    if (choice == 2)
                    {
                        games.displayGames();
                    }
                }
            }
        }
    }

    system("pause");
    return 0;
}

//functions
void ownerMenue()
{
    cout << "<< choose from the list >> :"
         << endl
         << endl
         << "1. Sort Games"
         << endl
         << "2. Order Game"
         << endl
         << "3. Stock Alert"
         << endl
         << "4. Display games"
         << endl
         << "5. Back"
         << endl
         << endl;
    cout << "enter your choice : ";
}
void startMenue()
{
    cout << "<< choose from the list >> :"
         << endl
         << endl
         << "1. Shop Owner"
         << endl
         << "2. Customer"
         << endl
```

```
         << "3. Exit"
         << endl
         << endl;
    cout << "enter your choice : ";
}

void coustomerMenue()
{
    cout << "<< choose from the list >> :"
         << endl
         << endl
         << "1. Buy a game"
         << endl
         << "2. Display all games"
         << endl
         << "4. Back"
         << endl
         << endl;
    cout << "enter your choice : ";
}
```

Explanation of the sorting/searching/list techniques used

Sorting Techniques:

In our project we used the selection sort algorithm which sorts an array by repeatedly finding the minimum element from the unsorted part and putting it at the beginning.

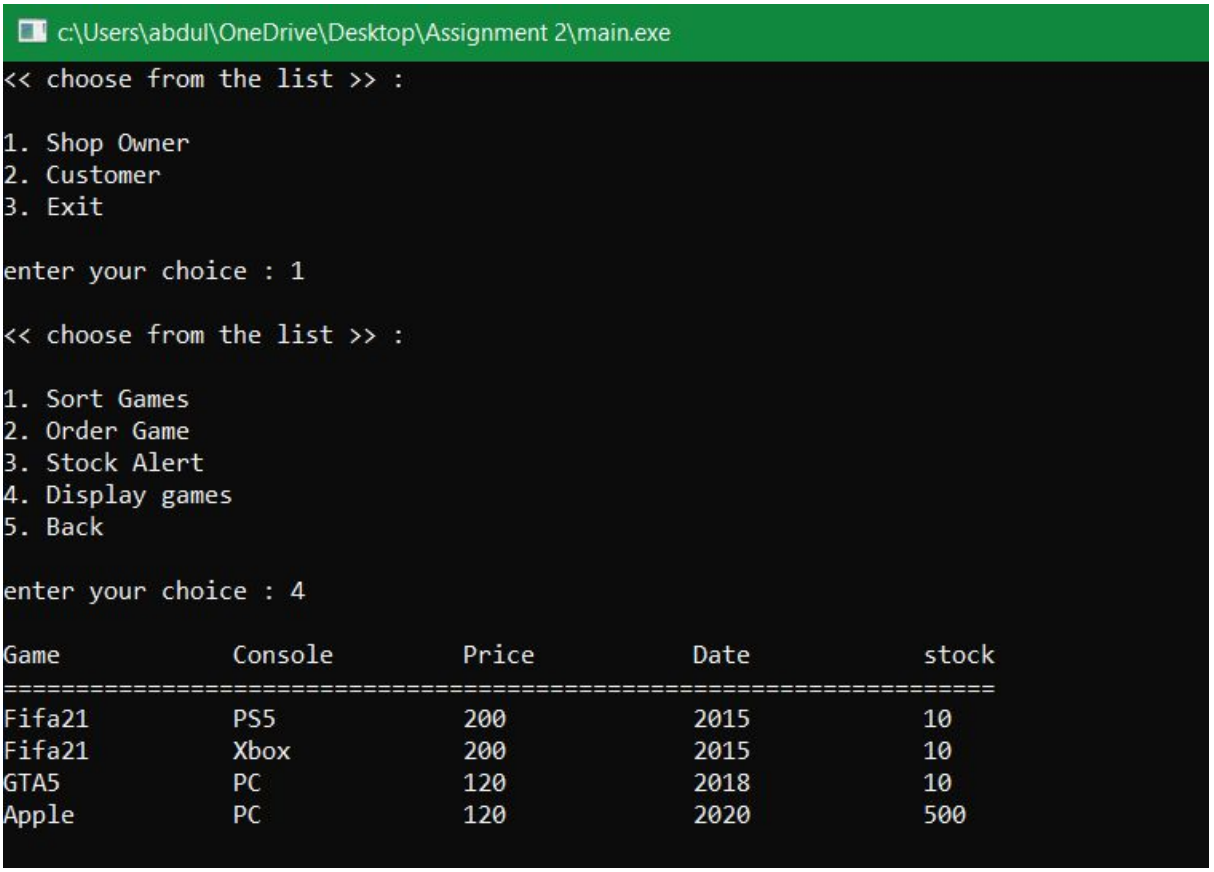
We used one function called sorting(), it takes no arguments, but inside the function we give the user five different ways to sort the data, which is Sort By Name, Sort By Console Name, Sort By Price, Sort By Date, or Sort By Stock. After that we take his/her choice and perform the selection sort on the data.

Searching Techniques:

Regarding the searching method, we used the sequential searching algorithm, sequential search is a very simple search algorithm.It goes over all items one by one. Every item is checked and if a match is found then that particular item is returned, otherwise the search continues till the end of the data collection.

Screenshot of the output

Owner Menu



```
<< choose from the list >> :

1. Shop Owner
2. Customer
3. Exit

enter your choice : 1

<< choose from the list >> :

1. Sort Games
2. Order Game
3. Stock Alert
4. Display games
5. Back

enter your choice : 1

<< choose from the list >> :

1. Sort By Name
2. Sort By Concole Name
3. Sort By Price
4. Sort By Date
5. Sort By Stock

enter your type : 3

<< choose from the list >> :

1. Sort Games
2. Order Game
3. Stock Alert
4. Display games
5. Back

enter your choice : 4

Game            Console         Price           Date            stock
=====================================================================
GTA5            PC              120             2018            10
Apple           PC              120             2020            500
Fifa21          PS5             200             2015            10
Fifa21          Xbox            200             2015            10
```

```
<< choose from the list >> :

1. Sort Games
2. Order Game
3. Stock Alert
4. Display games
5. Back

enter your choice : 2

<< choose from the list >> :

1. searching By Name
2. searching By Concole Name
3. searching By Price
4. searching By Date
5. searching By Stock
6. Back

enter your type : 1


Entet the name you are searching for: GTA5

============1============
Game            Console         Price           Date            stock
=============================================================================
GTA5            PC              120             2018            10

which game you want to update the stock ? : 1

 How many coppies You want to order ? 50

Done!!

<< choose from the list >> :

1. Sort Games
2. Order Game
3. Stock Alert
4. Display games
5. Back

enter your choice : 4

Game            Console         Price           Date            stock
=============================================================================
GTA5            PC              120             2018            60
Apple           PC              120             2020            500
Fifa21          PS5             200             2015            10
Fifa21          Xbox            200             2015            10
```

```
<< choose from the list >> :

1. Sort Games
2. Order Game
3. Stock Alert
4. Display games
5. Back

enter your choice : 3

This game has less than 10 in stock

Game            Console         Price           Date            stock
=================================================================
Fifa21          PS5             200             2015            10

This game has less than 10 in stock

Game            Console         Price           Date            stock
=================================================================
Fifa21          Xbox            200             2015            10
```

Customer Menu

```
<< choose from the list >> :

1. Shop Owner
2. Customer
3. Exit

enter your choice : 2

<< choose from the list >> :

1. Buy a game
2. Display all games
4. Back

enter your choice : 2

Game            Console         Price           Date            stock
=================================================================
GTA5            PC              120             2018            60
Apple           PC              120             2020            500
Fifa21          PS5             200             2015            10
Fifa21          Xbox            200             2015            10
```

```
<< choose from the list >> :

1. Shop Owner
2. Customer
3. Exit

enter your choice : 2

<< choose from the list >> :

1. Buy a game
2. Display all games
4. Back

enter your choice : 1

Choose the type of searching you want
<< choose from the list >> :

1. searching By Name
2. searching By Concole Name
3. searching By Price
4. searching By Date
5. searching By Stock
6. Back

enter your type : 3


Entet the number you are searching for: 120

============0============
Game            Console         Price           Date            stock
=========================================================================
GTA5            PC              120             2018            10

============1============
Game            Console         Price           Date            stock
=========================================================================
Apple           PC              120             2020            500

choose the game you want to Buy ? : 1

 How many coppies You want to Buy ? 3

Done!

Enter your name please: Abdulaziz
```

```
Here is your receipt
Name: Abdulaziz
Game Name: Apple
Game Console: PC
Price: 360RM
3 Copies
Thank You for shopping with us :) Vist us again!

<< choose from the list >> :

1. Buy a game
2. Display all games
4. Back

enter your choice : 2

Game            Console         Price           Date            stock
=============================================================================
Fifa21          PS5             200             2015            10
Fifa21          Xbox            200             2015            10
GTA5            PC              120             2018            10
Apple           PC              120             2020            497
```