

ANKARA YILDIRIM BEYAZIT UNIVERSITY  
BUSINESS SCHOOL DEPARTMENT OF MANAGEMENT  
INFORMATION SYSTEMS

Senior Project  
Final Report

Acquirer:  
Asst. Prof. İhsan Tolga Medeni  
Research Associate Dilan Özcan Kalfa

Prepared by  
Enes Eyüboğlu  
Berke Can Yaman  
İnanç Haluk Aksoy  
Ahmad Kuresi  
Mustafa Balta

This report is consisting of 4 main components:

- Final Report (Execution and Method)
- SPMP document
- SRS document
- SDD document

These components are displayed respectively as above in the document.

GitHub Link of the project:

<https://github.com/mustafabalta8/BSG-2.0>

# Business Simulation Game Final Report

## Final Report (Execution and Method)

## Table of Contents

<b>1. Back story &amp; purpose of the game .....</b>	<b>2</b>
<b>2. How to play .....</b>	<b>2</b>
<b>3. Tools and Infrastructure .....</b>	<b>2</b>
<b>3.1 Tools needed.....</b>	<b>2</b>
<b>3.2 Infrastructure .....</b>	<b>5</b>
<b>4. Product Acceptance Criteria.....</b>	<b>5</b>
<b>5. Testing .....</b>	<b>6</b>
<b>5.1 Functional Testing .....</b>	<b>6</b>
<b>5.2 Compatibility Testing .....</b>	<b>6</b>
<b>5.3 Performance Testing.....</b>	<b>6</b>
<b>5.4 Soak Testing .....</b>	<b>6</b>
<b>5.5 Security Testing.....</b>	<b>6</b>
<b>5.6 Sound Testing.....</b>	<b>6</b>
<b>6. Meeting .....</b>	<b>8</b>
<b>7. Udemy Courses .....</b>	<b>8</b>
<b>8. Deficiencies/Limitation .....</b>	<b>8</b>
<b>9. Main Panel .....</b>	<b>9</b>
<b>9.1 Software Development Panel .....</b>	<b>9</b>
<b>9.2 Contract Panel .....</b>	<b>9</b>
<b>9.2.1 Product List Panel.....</b>	<b>10</b>
<b>9.2.2 Create Product.....</b>	<b>10</b>
<b>9.2.3 Enhance Product Panel.....</b>	<b>10</b>
<b>9.3 Office Panel .....</b>	<b>11</b>
<b>9.3.1 Rent Option Panel .....</b>	<b>11</b>
<b>9.3.2 Office Asset Panel.....</b>	<b>12</b>
<b>9.4 Buy Share .....</b>	<b>12</b>
<b>9.4.1 Bought Share Panel .....</b>	<b>13</b>
<b>9.5 News Panel .....</b>	<b>13</b>
<b>9.6 Take Credit Panel.....</b>	<b>13</b>
<b>9.6.1 Balance Sheet Panel .....</b>	<b>14</b>
<b>9.7 Hire Employee Panel.....</b>	<b>15</b>

<b>9.7.1 My Employee Panel.....</b>	<b>15</b>
<b>9.7.2 Improve Employee Panel.....</b>	<b>16</b>
<b>9.8 Skill Tree Panel.....</b>	<b>16</b>
<b>10. Creating a C# Script in Unity.....</b>	<b>17</b>
<b>10.1 Scripts of BSG.....</b>	<b>19</b>
<b>10.2 Scripts Detail.....</b>	<b>20</b>
<b>10.3 Anatomy of the Company script .....</b>	<b>26</b>
<b>10.4 Function.....</b>	<b>28</b>
<b>11. References .....</b>	<b>32</b>

## **1. Back story & purpose of the game**

So basically, the story of the game goes like this, our main player who the players will play as is wrongfully fired from the company where he worked at his whole life. With that indignation he decides to launch his own tech company with all the severance payment that he was given and with one ambition and goal in his mind to become the most valuable company out there and to run his old company obsolete. As mentioned, the purpose in the game is to become the most valuable company in the stock market.

## **2. How to play**

Here we will concisely describe in verbal form on how to play the game later in the Panel section all the available windows, options, and actions that user can take will be displayed using screenshots (refer to page no:). So, when logged into the game player can see whole bunch of panels down below such as the stock panel, employee pane etc. and some panels are comprised of sub panels which presented when users click on them. Moreover, above the screen there are some indicators for instance time, money, and date. So, there are whole bunch of things users can do within the game keeping in mind the ultimate goal and for that user will have take on contract work, built their own software, hire new employees, buy furniture, and increment their employee's skill level. Since each action has varying consequences, user will have to think strategically on how to make their next move. So basically, players are in charge of how soon to reach the ultimate goal depending on their decision and moves.

## **3. Tools and Infrastructure**

### **3.1 Tools The tools needed for game development are as follows,**

#### **1. Game Engine For this project**

Unity will be used as the game engine. It has a lot of advantages over other game engines, mainly advantage is that is easier to use, and the games created are targeted at multiple platforms. If and when the game is going to be published into a new ecosystem Unity will provide much more flexibility.

#### **2. Coding Unity supports C#**

an industry-standard language with some similarities to Java or C++. Therefore, scripts that will control custom actions and interaction within the game will be written in C#. Developers are free to use any code editor they see fit.

### **3. Art and Design**

Adobe Photoshop and Illustrator will be the choice for digital art and design. Although there is numerus other software, Photoshop offers wider range of well-established tools and filters and will be used for painting 61 and jobs that require precise pixel placement, stroke-based painting and filters. Illustrator will be used to make vector art such as UI elements and head up textures as a vector package.

### **4. GitHub**

GitHub was used to collaborate on the project throughout from start to end. Repository that could be accessed by all the member was created as main. Since the project was divided into different tasks and assigned to each team members, through GitHub we were all simultaneously able to work on game and commit the changes to main repository for all member to update the project. Subbranches were created within the GitHub before pushing the changes to main repository to ensure changes were compatible and secure. In total we made about 160 commits to main repository in GitHub.

### **5. SQLite**

For the purposes of data storage, we choose to use SQLite since it is secure and free. SQLite was used to store all the data that needed to be saved and retrieved later and our game particularly revolves around that. The database contains multiple different tables to store all the necessary information.

#### **3.2 Infrastructure**

The game will not have a multiplayer feature; therefore, a powerful hosting server is not necessary. However, to push further updates and patches presence of a hosting server is needed. The game will be shipped by internet and installed directly on the users' device.

### **4. Product Acceptance Criteria**

The game should be in accordance with the storyline and design guide that are pre-determined in earlier steps moreover and importantly the game should be playable and enjoyable by the

targeted audience. During the development of the game, it should be programmed such that there should be as minimum as possible bugs and deficit. So therefore, during that phase we will thoroughly scrutinize and do a in depth testing of the product. All the cased that has been determined will be put to test to test the success of the game and the following checklist will be used for the conclusion of the final test.

- All the identified requirement is rightfully and purposefully implemented into the end products and module.
- All the modules have successfully completed their respective unit tests.
- All the components and module are successfully integrated and working in harmony and the integration test have passed successfully.
- The final documentation for the game is completed and includes all the necessary features.

## **5. Testing**

The testing phase will be divided into different stages each involving the test of different aspect of the game.

### **5.1 Functional Testing**

This is a functionality QA testing style where test is conducted to search and locate generic problems such user interface, graphics, game mechanic issue and game integrity. Moreover, using this, the game will be tested in the following areas, user interface whether it is easy to use and user friendly, checking menu structure, color, backgrounds and loading time, audio alignment and time out.

### **5.2 Compatibility Testing**

Here the game will be tested to see if it able to run across different PC with different operating systems such as Windows, Linux and Mac and additionally the various configuration of hardware and software. This test will be done through installing and uninstalling on all supported operating systems.

### **5.3 Performance Testing**

In performance testing the overall working of the game is tested and if necessary, tuning is performed to optimize the gaming experience. Here we will test the battery consumption and graphic performance of the game to check whether prolonged playing of the game has any bearing over its quality. Processor and memory test will be done to see how much strain game puts on memory and processor. Network testing will be done to see how smooth the process is of downloading the game from the cloud and how much internet bandwidth does it consume.

### **5.4 Soak Testing**

Here the game is open and left running for as long as possible in numerous modes of operation for example title screen. We will start the game and character will be left to stand in the idle position for let us say 24 hours, this will allow us to detect any crashes brought on memory or game engine and cloud.

### **5.5 Security Testing**

In this section or stage game will be tested for any kind of bugs that will allow third party to gain unauthorized access to the games code and game itself and make alterations.

### **5.6 Sound Testing**

Testing to see if there is any error in sound file loading and listening for any disorientation that may be present.

After all the development phases are completed, we will thoroughly test the game this means that all the requirement of the game that we allocated in the initial phase will be put to test. The test will be conducted in conjunction with the development team. Rigorous test of the game will enable us to fix the bugs and create patches for any kind of disorientation in the game.

Besides the aforementioned test we will additionally have gamers come in to play the game from start to end, in this way we can improve the user experience through getting feedback on the overall working of the game.

## **6. Meetings**

Throughout the project we conducted like dozens of meetings, the platforms that we used for our meetings were Discord, Google Meet, and TeamViewer. The meetings were held to update the group about the task each member was working on and the progress of the game so far or to help one another if they hit a hurdle along the way. For that purpose, we mostly used TeamViewer where one of the team members who knew how to tackle the problem would connect and help. Each of our meetings lasted about at least two and half hours where we not only discussed our current progress but also how to move forward, our timeline, our overall progress, and the assigning of new task till next meeting.

## **7. Udemy Courses**

In order for us to take on this project we had to learn many of the concept, tools, development environment and C# programming specific to unity and for that we took Udemy courses. These courses enabled us to learn how to work with 2D graphics, how to incorporate sound, how to use Unity game engine with all its perks and tools and last but not least C# programming with specific libraries attributed to unity. Reference to these courses are provided in form of link under reference section.

## **8. Deficiencies/Limitation**

There are several limitations associated with this game, one that is most explicit is that it is not a 3D game and therefore does not offer the features attributed to the 3D games. Moreover, another limitation is that this game is only available on PC and therefore its format is not supported and cannot be played on neither PlayStation nor Xbox. Additionally, the game will be hosted on cloud-based server and the user will have to log into that server and download the game from there and will have to have internet connection whilst playing the game, so therefore player should have to have decent internet connection. Secondly, there will be the disadvantage associated with the cloud, which are the risk of cyber-attack and the risk of cloud being crash due to technical defect. Lastly, though game has high resolution, but since the game is in pixels the quality may not yield the normal 4k exposure and fully immerse the player.

## 9. Main Panels



The game consists of 7 main panels. Some of these panels have subpanels. Others are just a single panel. The actual gameplay of the game proceeds through these panels.

### 9.1 Software Development Panels

Software development has 4 subpanels. These are contracts panel, products list panel, enhance product panel and create product panel. These panels focuses on players products and how player can manage them.

### 9.2 Contracts Panel



Contract Panel show player where he can buy contracts.

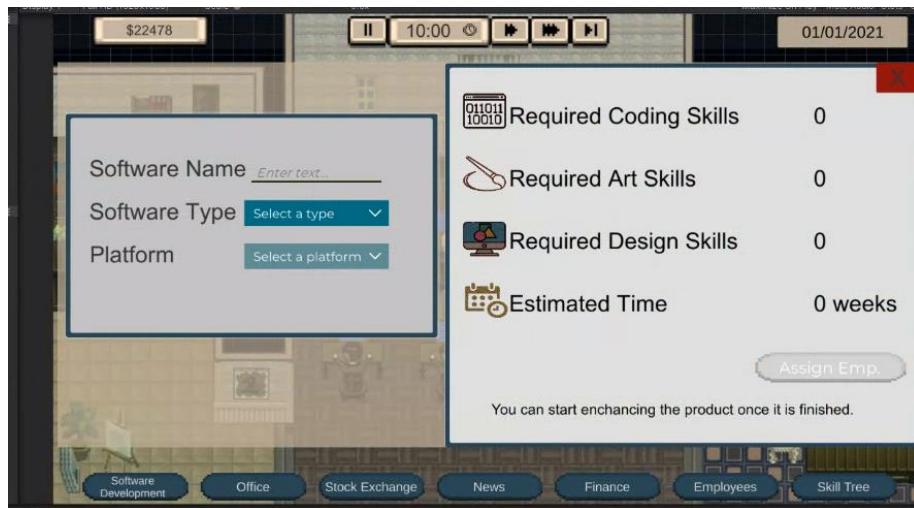
### 9.2.1 Products List Panel



PRODUCTS				
NAME	TYPE	PLATFORM	QUALITY	AMOUNT OF SALE
test	Console	Operating System	2	552
test 2	Desktop	Mobile App	4	1525
test product	Desktop	Mobile App	4	1905
fgfdfg	Mobile Device	Mobile App	3	786
	Mobile	Mobile App	4	1234

Product list panel lists player the products he owns.

### 9.2.2 Create Product



The Create Product panel allows the player to input product details and manage required skills. It includes fields for Software Name, Software Type, Platform, and Estimated Time. A note at the bottom states: "You can start enhancing the product once it is finished."

Software Name	Enter text
Software Type	Select a type
Platform	Select a platform
Required Coding Skills	0
Required Art Skills	0
Required Design Skills	0
Estimated Time	0 weeks

In this panel, player create products. In order to do that he should have required skills.

### 9.2.3 Enhance Product Panel



This panel allows player to enhance the product which he created in the create products panel.

### 9.3 Office Panel



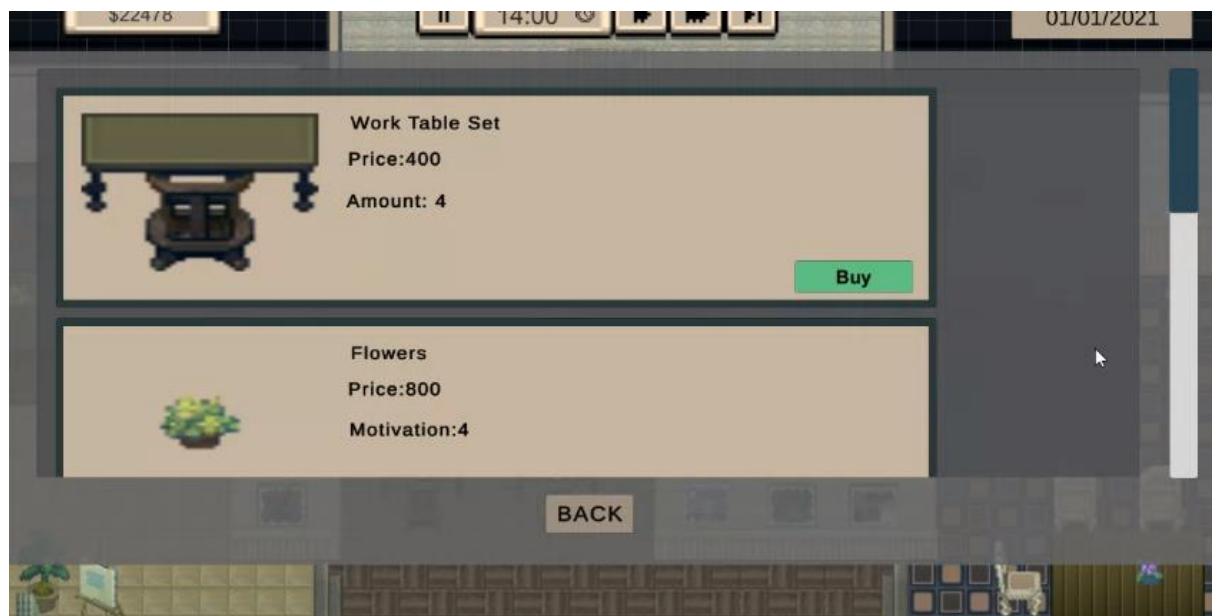
In Office panel there are 2 subpanels which are rent options panel and assets panel.

#### 9.3.1 Rent Options Panel



In this panel, player see his rent options.

### 9.3.2 Office Assets Panel



This panel allow players to see the assets which are buyable for their offices.

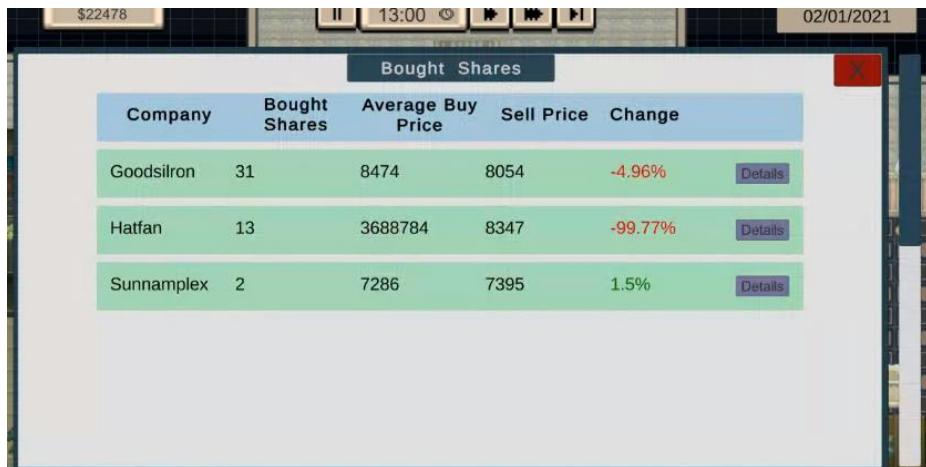
### 9.4 Buy Shares

The screenshot shows a "Buy Share" panel with a list of companies and their details:

Buy Share				
RANK	COMPANY	BRAND VALUE	PRICE	SHARES
1	Hatfan	95	\$8347	98/100
2	Goodsilron	90	\$8054	98/100
3	Sunnamplex	87	\$7395	100/100
4	Toughzap	83	\$6924	100/100
5	Betatech	83	\$6841	100/100
6	Doncon	83	\$6831	100/100

In buy shares panel, players can buy other firms' shares'.

#### 9.4.1 Bought Shares Panel



Company	Bought Shares	Average Buy Price	Sell Price	Change	
Goodsilron	31	8474	8054	-4.96%	<a href="#">Details</a>
Hatfan	13	3688784	8347	-99.77%	<a href="#">Details</a>
Sunnamplex	2	7286	7395	1.5%	<a href="#">Details</a>

In this panel, players see the shares they bought in details. In addition they can sell them to gain profit.

#### 9.5 News Panel



**Best Bitcoin wallets in 2021**  
Keep your crypto safe with one of the best Bitcoin wallets around.

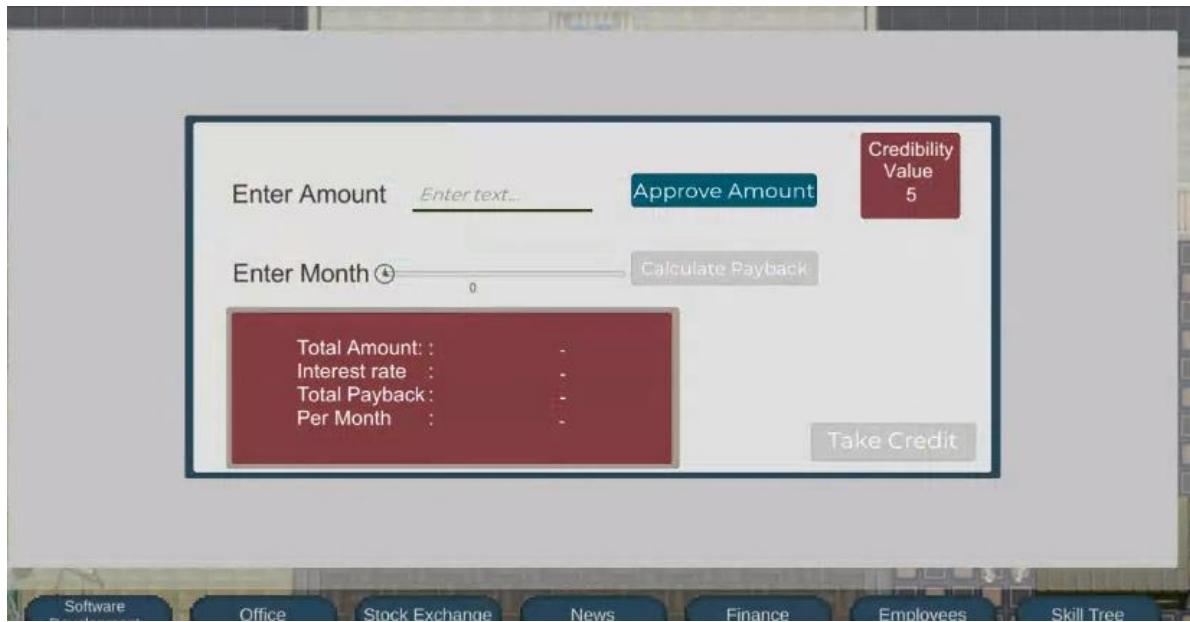
Software Times 8120

**Megasoft Edge Windows 10 brings a game-changing feature to Edge to beat Chrome**  
Megasoft's Edge browser has been gaining a dedicated following in recent months thanks to an ever-increasing list of features that its users seem to favor over offerings from rivals such as Boogie Chrome or Mozilla Firefog

Software Times 2197

In news panel, players can see news from technology world.

#### 9.6 Take Credit Panel



This panel allows players to take credit. The credibility value calculates how much they can take. The higher credibility value is the higher amounts of money they can get.

### 9.6.1 Balance Sheet Panel

Assets		Liabilities	
Cash	22478	Purchase of Inventory	-500
Gains from Contracts	0	Advertising	0
Gains from Features	0	Management and Staff	0
Inventory	0	Training	0
Total Current Assets	0	Taxes	0
Total Assets	22478	Total Liabilities	-10120

Balance sheet panel is the panel where players see their assets and liabilities. They do not only see their total assets and liabilities, they also see the detailed information for those terms too.

## 9.7 Hire Employee Panel



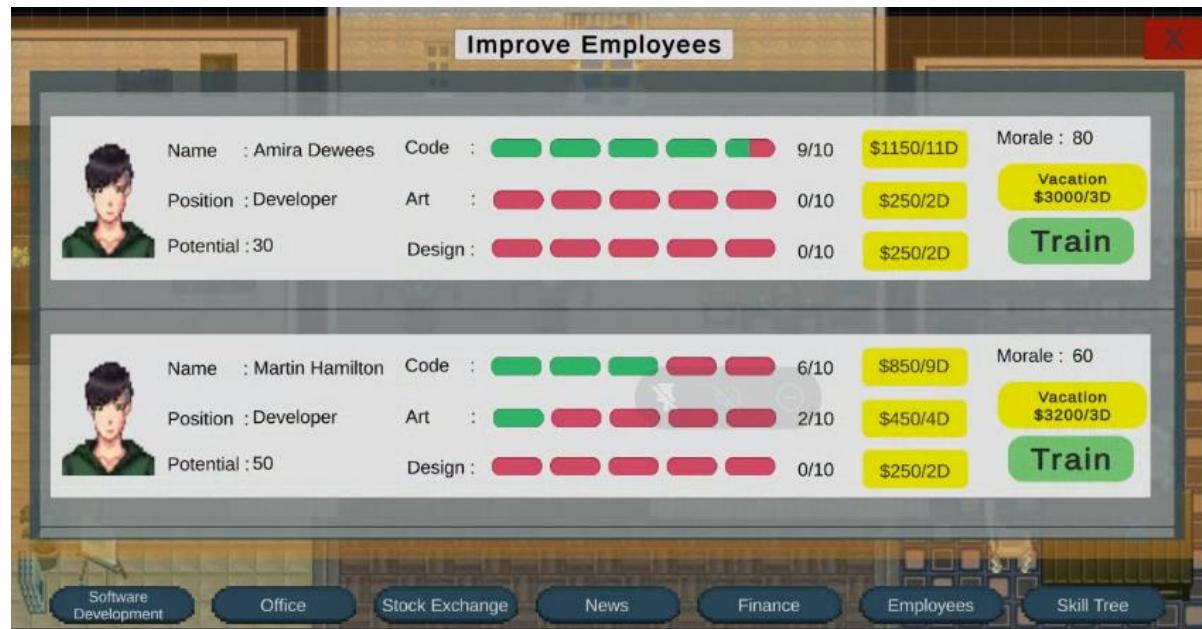
This panel allows players to hire employees. Each employee has different skill levels and salaries.

### 9.7.1 My Employees Panel



In this panel, players can see their employees. They can also fire them if they think they do not deserve their salaries.

### 9.7.2 Improve Employees Panel



Improve employees panel allow players to improve their employees by training them in a single area.

### 9.8 Skill Tree Panel

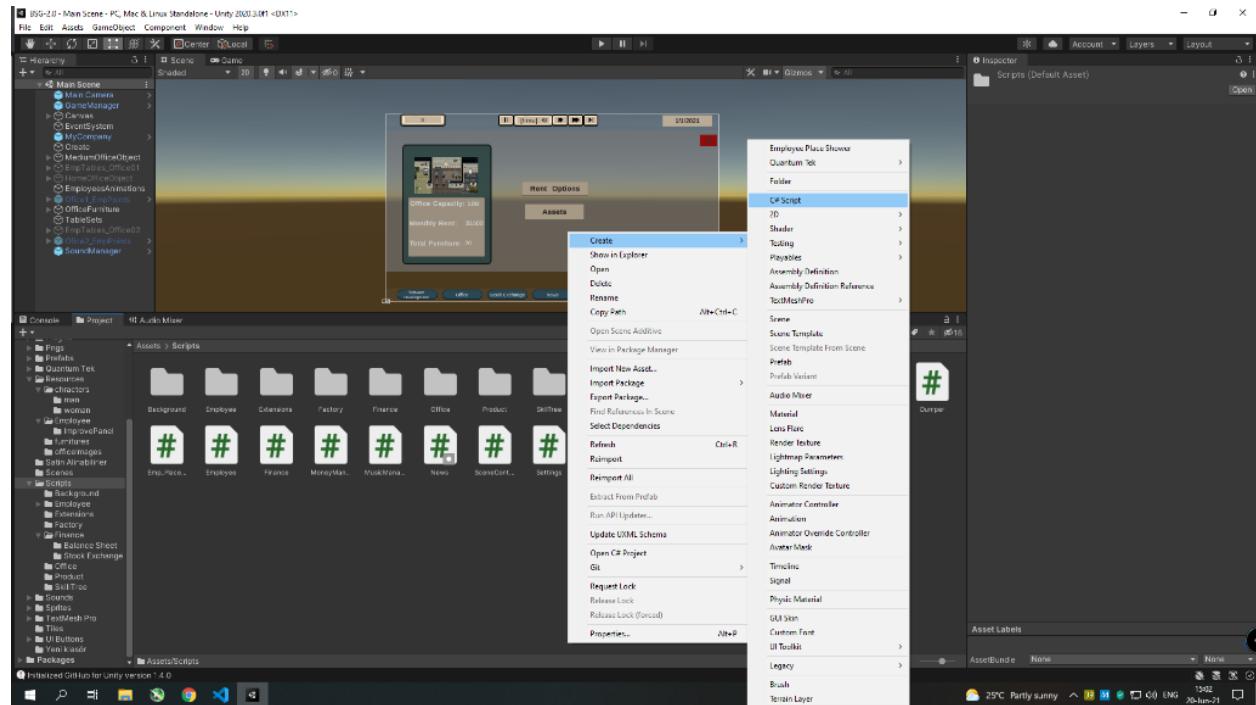


Skill tree panel allows player to have different kinds of skills. Once one of these skills is bought player can use that passive service all game.

## 10 Creating a C# script in Unity

Unity supports the C# programming language natively. Therefore, the BSG has 41 C# scripts in various places that handle its GameObjects. These scripts allow us to trigger game events, modify Component properties over time and respond to user input in various ways.

A script “ExampleScript” is created using the Unity GUI as shown below.



Created C# file contains pre coded content as shown below.

```
using UnityEngine;
using System.Collections;

public class CreatedScript: MonoBehaviour {

    // Use this for initialization
    void Start () {
```

```
}

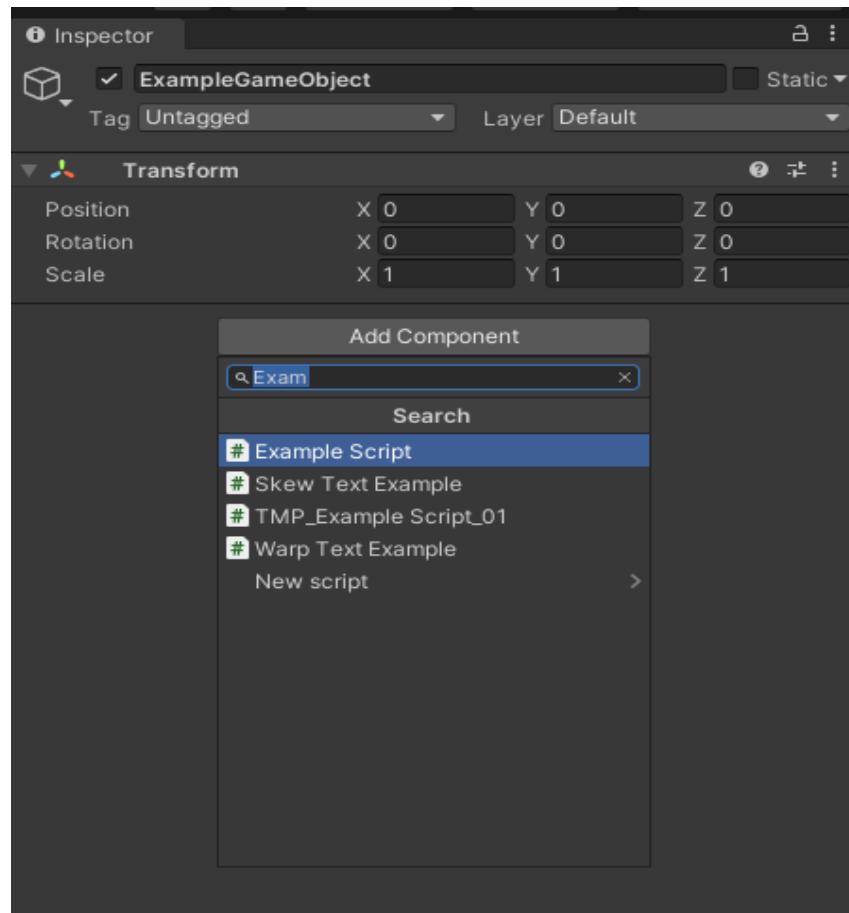
// Update is called once per frame
void Update () {

}

}
```

The ExampleScript contains two functions. Start function will be called before the gameplay begins. Update function is called once per frame.

To add this script to the GameObject, we can drag and drop it from the project manager or add component from the GameObject's inspector.



Now, whenever the game starts, ExampleScript's Start() and Update() functions will be run.

## 10.1 Scripts of BSG

Folder structure of the Script folder which contains all the scripts is given below.

```
Scripts:
| AcceptContract.cs
| Ads.cs
| AssignEmp.cs
| Company.cs
| Contract.cs
| DbManager.cs
| Dumper.cs
| Employee.cs
| Emp_Place_Points.cs
| ExampleScript.cs
| Finance.cs
| MoneyManager.cs
| MusicManager.cs
| News.cs
| SceneController.cs
| Settings.cs
| TimeManager.cs
| UIManager.cs
+---Background
| BackgroundWorker.cs
| NewDay.cs
| Notifications.cs
+---Employee
| | ImproveEmployee.cs
+---Extensions
| Extensions.cs
+---Factory
| CreateContract.cs
| CreateEmployee.cs
| CreateOfficeUI.cs
| CreateProduct.cs
+---Finance
| |
| +---Balance Sheet
| | BalanceSheet.cs
| \--Stock Exchange
| | BoughtShares.cs
| | BoughtSharesDetails.cs
| | SC_Company.cs
| | StockExchange.cs
| | TextMeshProGui.cs
```

```
----Office
|   FurnitureUI.cs
|   MyOffice.cs
|   Office.cs
|   TableSetUI.cs
+---Product
|   ListProduct.cs
|   Product.cs
|   ProductManager.cs
+---SkillTree
|   Skill.cs
|   SkillTreeOperator.cs
```

## 10.1 Script Details

### 1. AcceptContract.cs

This script is responsible for the contract logic of the game. It orchestrates how the contracts are listed, how a contract will be taken on, how it starts, how and when it ends.

### 2. Ads.cs

This script is responsible for the advertisement logic of the game. User can start a new advertisement campaign.

### 3. AssignEmp.cs

This script allows user to choose employees to assign to an ongoing project, contract or product creation.

### 4. Company.cs

This script holds most of the information that is needed in order to play the game. Rented office, bank balance, general company information, employees and skills are stored in this script. We will go into detail of this script later on in this chapter.

## **5. Contract.cs**

This script assigns values of the created contract to the respective GameObject.

## **6. DbManager.cs**

This script is the workhorse of the game project. All the database management logic is orchestrated within this script. It connects to the pre-created database and establishes a connection thus granting us the ability to read, insert, delete and update to this database.

## **7. Employee.cs**

This script houses all the necessary information about an employee such as its name, salary, its overall power, coding, design and art skills, morale, age, potential, position, profile picture, whether it is active or not. Also, this script is responsible for training, vacation, hire and fire logic of the employees.

## **8. Emp\_Place\_Points.cs**

This script contains pre given positions of the employee animation on the office tilemap.

## **9. Finance.cs**

This script is used for finance related issues. It helps user take credit based on its credibility value. Credits are given at 5 different levels based on the company credibility value.

## **10. MoneyManager.cs**

This script manages all the money related issues. Adding and subtracting money to and from bank balance is done using this script. It also stores the bank transactions in the database.

## **11. MusicManager.cs**

This script orchestrates all the sound related issues. The background music is played from this script. It also plays notification sound, button on and off sound, money sound and new day sound.

## **12. News.cs**

News script shows news to the user in the News panel. 2 new news stories are being shown every day.

## **13. SceneController.cs**

This script helps change the menu to the game and back.

## **14. Settings.cs**

This script manages different resolutions that the game will be run on.

## **15. TimeManager.cs**

TimeManager script manages all the time related issues. It holds the time that can be accessed from everywhere from the project. It can stop, speed up, speed down and restart the time. It also converts the turn-based timescale to real dates starting from 01.01.2021.

## **16. UIManager.cs**

Controls opening and closing of some panels. Play sounds.

## **17. BackgroundWorker.cs**

This script is a background worker that checks issues related to products, salaries and office and runs every month in the game. Product function checks for product sales, salary function checks for salaries of employees and subtracts from the bank balance, office rent function does the same for the office rent.

## **18. NewDay.cs**

Controls the new day effect on the day change screen.

## **19. Notifications.cs**

This script is responsible for the in-game notifications. pushNotification function gets a payload as parameter and shows the payload to the user for a given time.

## **20. ImproveEmployee.cs**

This script calculates potential of the employees, new skill prices, vacation prices and improves the employees according to those prices.

## **21. CreateContract.cs**

This script creates random contract to be listed in the contracts panel.

## **22. CreateEmployee.cs**

This script creates an Employee object from the pre-saved data in the database.

## **23. CreateOfficeUI.cs**

CreateOfficeUi is responsible for the object within the office. It creates and places those objects that can be bought.

## **24. BalanceSheet.cs**

This script manages the balance sheet panel. It fetches and lists assets and liabilities based on categories from the database.

## **25. BoughtShares.cs**

This script manages the bought shares panel. It fetches and lists previously bought shares from the database.

## **26. BoughtSharesDetails.cs**

This script is a helper script that extends from the bought shares script. It fetches previously bought shares based on the company selected from the database.

## **27. StockExchange.cs**

This script manages the stock exchange panel. It lists stocks from companies, calculates prices and changes over time. Allows user to buy stocks.

## **28. FurnitureUI.cs**

This script controls the buying process of buyable objects from the office. It checks if user can buy, if so places the object in the office and subtracts money from the bank balance.

## **29. MyOffice.cs**

Checks which office is rented at the moment and places the objects bought into the office.

### **30. Office.cs**

Allows user to rent a new office. Shows how much the rent is for each office and its capacity.

### **31. TableSetUI.cs**

Allocates a set of table objects for an employee to occupy. A table object consists of a worktable and office chair. If the user does not meet minimum requirements for a table object.

### **32. ListProduct.cs**

This script manages the products panel. It fetches and lists created products from the database.

### **33. Product.cs**

This script contains an object for a product.

### **34. ProductManager.cs**

ProductManager is responsible for all the product related issues in the project. It manages creation and enhancement of the products from the respective panels.

### **35. Skill.cs**

This script manages the skill tree panel. It fills in the buttons and details of the clicked button to its respective area.

## 36. SkillTreeOperator.cs

Operates the Skill script by taking in parameters from Unity and passing them through to the Skill script.

### 10.2 Anatomy of the Company script

The Company script is made out of 571 lines of code. It regulates the most important things in the game. It holds information related to office, bank, general company information, employees, employee animation and skills.

#### 10.2.1 Variables

- Office related
  - officeId (public int)
  - ourOfficeCapacity (public int)
  - ourOfficeRent (public int)
  - motivation (public int)
  - ourOffice (Serializable GameObject)
- Bank related
  - balance (public int)
- General Company Information
  - companyName (public string)
  - companyPower (public int)
  - totalSalary (public int)
  - credibility (public int)
  - employees (public List<>)

- Skill related

- skill (public KeyValuePair<string,int>)
- thrifty (public bool)
- chafferer (public bool)
- motivator (public bool)
- fertile (public bool)
- peaceful (public bool)
- sensei (public bool)
- business\_class (public bool)
- capitalist (public bool)
- dream\_team (public bool)
- instructive\_leader (public bool)
- thrifty\_level (public int)
- chafferer\_level (public int)
- motivator\_level (public int)
- fertile\_level (public int)
- peaceful\_level (public int)
- sensei\_level (public int)
- business\_class\_level (public int)
- capitalist\_level (public int)
- dream\_team\_level (public int)
- instructive\_leader\_level (public int)

- Employee related

- MyEmployeeHolder (Serializable GameObject)
- MyEmployee (Serializable GameObject)
- Employee\_woman[1-9] (Serializable Sprite)
- Employee\_man[1-9] (Serializable Sprite)
- motivator\_activator (public bool)
- morale\_boost (public int)
- References to other scripts
  - dbManager
  - employeeFactory

### 10.3 Functions

- Start() function

```

dbManager = FindObjectOfType<DbManager>(); //get the database manager script

employeeFactory = FindObjectOfType<CreateEmployee>(); //get the employee script

ShowUpdateOnOfficeValues(); // call to the ShowUpdateOnOfficeValues() function

FillEmployeePoints(); // call to the FillEmployeePoints() function

getEmployeesFromDatabase(employeeFactory); //call to the getEmployeesFromDatabase() function

getCompanyDataFromDatabase(); //call to the getCompanyDataFromDatabase() function

getSkills(); //call to the getSkills() function

employeeFactory.createRandomEmployee(5); //creates X random employees

```

Start function will be called before the gameplay begins. First database manager script and employee factory scripts are stored into the respective variables. Then a series of functions are called.

```
public void ShowUpdateOnOfficeValues()  
{  
    ...  
}
```

ShowUpdateOnOfficeValues() updates the officeId, ourOfficeCapacity and rent variables from the database. It checks if buyable objects are bought, then adds the pre calculated amount of morale to all employees. Employee morale boost is calculated by another function called moraleCalculator(int base\_morale) which gets the base morale from the object as a parameter, then calculates the final amount by looking at the acquired skills.

```
public void FillEmployeePoints()  
{  
    ...  
}
```

FillEmployeePoints() function fills the pre-defined points on the office tilemap with employees.

```
void getEmployeesFromDatabase(CreateEmployee employeeFactory)  
{  
    ...  
}
```

getEmployeesFromDatabase(CreateEmployee employeeFactory) function gets the employees that have been hired from the database. It takes the employeeFactory as a parameter. First, the object holding employees is cleared by deleting all its children. Then, all hired employees are selected and fetched from the database. The returned employees are created into an employee object and added to the employees list. This script also puts the data about the created employee into pre-defined text objects to be shown in the game. Finally, the before-calculated morale boost is added to the employees' morale.

```
public void getSkills()  
{  
    ...  
}
```

getSkills() function calls a series of getSkill() functions when called.

```
public void getSkill(string skillName)  
{  
    ...  
}
```

getSkill() takes one string as a parameter and checks if the user acquired the given skill in the parameter. If the given skill is acquired, its boolean will become true and skill\_level variable will be filled with the level acquired.

```
public void EmployeeAnimation()  
{  
    ...  
}
```

EmployeeAnimation() function plays the animation for the employees that have been hired. Each employee has its own animation and this animation will be played by this function on the predefined positions.

```
public void AddEmployeeToCompany(Employee employee)  
{  
    ...  
}
```

AddEmployeeToCompany(Employee employee) function requires an Employee object as parameter and is called when the user decides to hire a new employee. It updates the employees' hiring status in the database and adds it to the companies' employees list.

```
public void FireEmployee(Employee employee)  
{  
    ...  
}
```

FireEmployee(Employee employee) function requires an Employee object as parameter and is called when the user decides to fire one of its employees. It updates the employees' hiring status in the database and removes the employee from the companies' employees list.

## 11 References

<https://www.youtube.com/watch?v=MWOvwegLDI0>

<https://www.udemy.com/share/101WeyAEISchlRRngH/>

<https://www.udemy.com/share/101CyqAEISchlRRngH/>

<https://www.udemy.com/share/101WjsAEISchlRRngH/>

<https://ngakademi.com/courses/adan-zye-temel-c-programlama-egitimi/>

<https://ngakademi.com/courses/ozel-ders-formatinda-adan-zye-nesne-tabanli-programlama-egitimi/>

<https://pixel-me.tokyo/en/>

<https://www.youtube.com/watch?v=a1RFxtuTVsk>

<https://opengameart.org/>

<https://craftpix.net/categorys/2d-game-kits/>

<https://mycolor.space/>

<https://www.sprites-resource.com/>

<https://pngtree.com/>

<https://unity3d.com/unity/whats-new/2020.3.0>

<https://pikbest.com/>

<https://2darray.itch.io/office-pie>

<https://pixabay.com/tr/>

<https://www.freepik.com/free-photos-vectors/free-pics>

<https://www.udemy.com/course/build-a-tycoon-business-sim-in-unity3d-c-game-development/#content>

[https://www.youtube.com/playlist?list=PLLzCsPvhLpWyzp8\\_rDyaiNMZiVQaLeCzL](https://www.youtube.com/playlist?list=PLLzCsPvhLpWyzp8_rDyaiNMZiVQaLeCzL)

<https://www.raywenderlich.com/418-how-to-save-and-load-a-game-in-unity#toc-anchor-005>

[https://www.youtube.com/watch?v=IkWqzjVS0TI&ab\\_channel=AkbarProject](https://www.youtube.com/watch?v=IkWqzjVS0TI&ab_channel=AkbarProject)

ANKARA YILDIRIM BEYAZIT UNIVERSITY  
BUSINESS SCHOOL  
DEPARTMENT OF MANAGEMENT INFORMATION  
SYSTEMS

**Business Simulation Game**  
**Software Project Management Plan**

*Acquirer:*

*Asst. Prof. İhsan Tolga Medeni*

*Research Associate Dilan Özcan*

*Prepared by*

*Enes Eyüboğlu*

*Berke Can Yaman*

*İnanç Haluk Aksoy*

*Ahmed Qureshi*

*Mustafa Balta*

## Version History

<b>Version #</b>	<b>Implemented By</b>	<b>Revision Date</b>	<b>Reason</b>
1.0	<i>BSG Team</i>	<i>25.11.20 20</i>	<i>Initial version</i>
1.1	<i>BSG Team</i>	<i>28.11.20 20</i>	<i>Gantt Chart, Budget Updates</i>

## Table of Contents

<b>Version History</b> .....	2
1 - Introduction .....	5
1.1 Purpose, Scope and Objectives .....	5
1.2 Assumptions and Constraints .....	6
1.2.1 Constraints: .....	6
1.2.2 Assumptions: .....	7
1.3 Project Deliverables .....	7
1.4 Project schedule & Budget Summary .....	7
1.5 Evolution Plan .....	9
2. References .....	9
3. Definition .....	9
4. Project Organization .....	10
4.1 External Interfaces .....	10
4.2 Internal Structure .....	11
4.3 Roles and Responsibilities .....	11
4.3.1 Roles .....	11
4.3.2 Responsibilities .....	13
5. Managerial Process Plan .....	14
5.1 Start-up Plan .....	14
5.1.1 Staffing Plan .....	14
5.1.2 Resource Acquisition Plan .....	16
5.2 Work Plan .....	16
5.2.1 Work Activities .....	17
5.2.2 Schedule Allocation .....	52
5.2.3 Resource Allocation .....	53
5.2.4 Budget Allocation .....	53
5.3 Control Plan .....	54
5.3.1 Requirement Control Plan .....	54
5.3.2 Schedule Control Plan .....	54

5.3.3 Budget Control Plan .....	55
5.3.4 Quality Control Plan.....	55
5.3.5 Reporting Plan .....	55
5.4 Risk Management Plan.....	55
5.5 Closeout Plan.....	57
6. Technical Process Plan .....	58
6.1    The Process .....	58
6.2    Methodology .....	58
6.3    Tools and Infrastructure.....	60
6.3.1    Tools .....	60
6.3.2    Infrastructure .....	61
6.4 Version Control .....	61
6.5 Product Acceptance Tasks .....	62
6.5.1 Product Acceptance Criteria .....	62
6.5.2 Schedule.....	62
6.5.3 Documentation.....	62
7. APPENDIXES.....	63
7. Appendix A: Gantt Chart .....	63

# 1 - Introduction

Software Project Management Plan (SPMP) is done as part of the senior project, which is game development, by the undergraduate students who are majoring in management information systems at Ankara Yıldırım Beyazıt University (AYBU). This project consists of two part and needs to be completed in over a year. The first part includes research and writing of documents that comprises of this one SPMP and SRS, and SDD and needs to be submitted by the end of the first semester. The second part which is due second semester is where we will actually build the game to its required and identified capacity and thereby successfully fulfill the objectives of the senior project.

## 1.1 Purpose, Scope and Objectives

This document is like blueprint for creating or developing software as it identifies and established all the necessary steps such as roles, responsibilities, processes, and schedules for managing the software development process. Moreover, it provides the budgeting and due date of each phase in a software development and the tools the method that will be incorporated to build the software successfully.

This document contains a plan for achieving the goals of the project. The goals will be defined in more detail in the Software Requirements Specification (SRS) document. In the game application, software company management of the user to be tested in a way and user decisions will either lead to success or fail. Either way, our goal is for the user to have fun while playing.

The purpose of this project is to create business simulation game for any player who is interested in simulation type of game. In this game, player starts off as a software start-up company and manage this company as you would manage a company in real life. In the game, player will create software/technology products which in turn will

increase the valuation of the company. This game's ultimate purpose is to become the most valuable company in the stock market.

The actor in the game is **Player** and **AI system** of the game to some extent

Player will be able to:

- Create account
- Select region
- Run startup
- Invest
- Hire employee
- Buy or sell share
- Seek credit from bank

AI system

- Record player activity
- Render decision based on player action
- Control stock market
- Render and control crisis or disaster

## 1.2 Assumptions and Constraints

### 1.2.1 Constraints:

- Our product will be developed by five students as a part of Senior Project. Therefore, there will be no payments for this project.
- The other important constraint is the fixed schedule to finish the project. Preparation of Documents end at the end of fall term, and the Software Development process ends at the end of spring term.

- The software development process and documents will be prepared according to the IEEE Standards 1058 – 1998. 1058 is the SPMP Standard.

### 1.2.2 Assumptions:

- Complete the project within the given Deadline.
- Create a tested game application with corrected errors.
- Make sure the interface of the application is user friendly.

### 1.3 Project Deliverables

The table below details the deliverables that need to be submitted throughout this project and the date they will be submitted.

<b>Product Delivery</b>	<b>Delivery Date</b>	<b>Delivery Location</b>	<b>Format of the Media</b>	<b>Delivery Media</b>
SPMP	02.12.2020	Email/Dropbox	Soft Copy	PDF
SRS	09.12.2020	Email/Dropbox	Soft Copy	PDF
SDD	01.01.2021	Email/Dropbox	Soft Copy	PDF
BSG Demo	15.04.2021	University	Soft Copy	PDF
Final Product	03.05.2021	University	Soft Copy & Hard Copy	PDF

### 1.4 Project schedule & Budget Summary

Product	
Delivery	Date
SPMP	02.12.2020
SRS	09.12.2020
SDD	01.01.2021
BSG Demo	15.04.2021
Final Product	03.05.2021

The table above displays the documents & product's due date

Name	Duration	Cost
<b>BSG</b>	<b>114</b>	<b>£100,000.00</b>
<b>Initiation</b>	<b>10</b>	<b>£10,500.00</b>
<b>Planning</b>	<b>20</b>	<b>£5,000.00</b>
<b>Execution</b>	<b>50</b>	<b>£25,000.00</b>
<b>Game Design</b>	<b>8</b>	<b>£15,000.00</b>
<b>Implementation of the Game</b>	<b>5</b>	<b>£10,000.00</b>
<b>Testing</b>	<b>7</b>	<b>£4,000.00</b>
<b>Maintenance</b>	<b>2</b>	<b>£10,000.00</b>
<b>Evaluation</b>	<b>4</b>	<b>£2,500.00</b>
<b>Conclusion</b>	<b>2</b>	<b>£3,000.00</b>
<b>Marketing</b>	<b>6</b>	<b>£15,000.00</b>

The table above contains the phases of the game development project, additionally the time it will take to complete each phase and cost of overall and each phase.

## 1.5 Evolution Plan

After the documents been reviewed by the evaluator and us as team and as we move on with the project new information may surface or we may face challenges that may require us to re adjust the cost, the timeline or changes within the game itself to ensure such transformation occurs as smoothly as possible we plan on to gather each week and review the flow of project, whether it going according to the pre-set timeline or whether any adjustment to game is required. If any changes are made, necessary revision to the effected documents will be made and new version of final product will be released.

## 2. References

IEEE STD 1058-1998, IEEE Standard for Software Project Management Plans

<https://unity.com/how-to/programming-unity>

<https://www.scrum.org/resources/blog/scrum-process-explained>

<https://e.huawei.com/en/eblog/enterprise-networking/switches/software-or-hardware-sdn>

## 3. Definition

WBS = Work Breakdown Structure

SSD: Software Design Document

SPMP: Software Project Management Plan

SRS: Software Requirement Specification

UC: Use Case

UI: User Interface

## 4. Project Organization

This section specifies the process model for the project and its organizational structure, and project roles and responsibilities.

### 4.1 External Interfaces

**Acquirer/Supervisor:** Assoc. Prof. Dr. İhsan Tolga Medeni, Research Associate  
Dilan Özcan

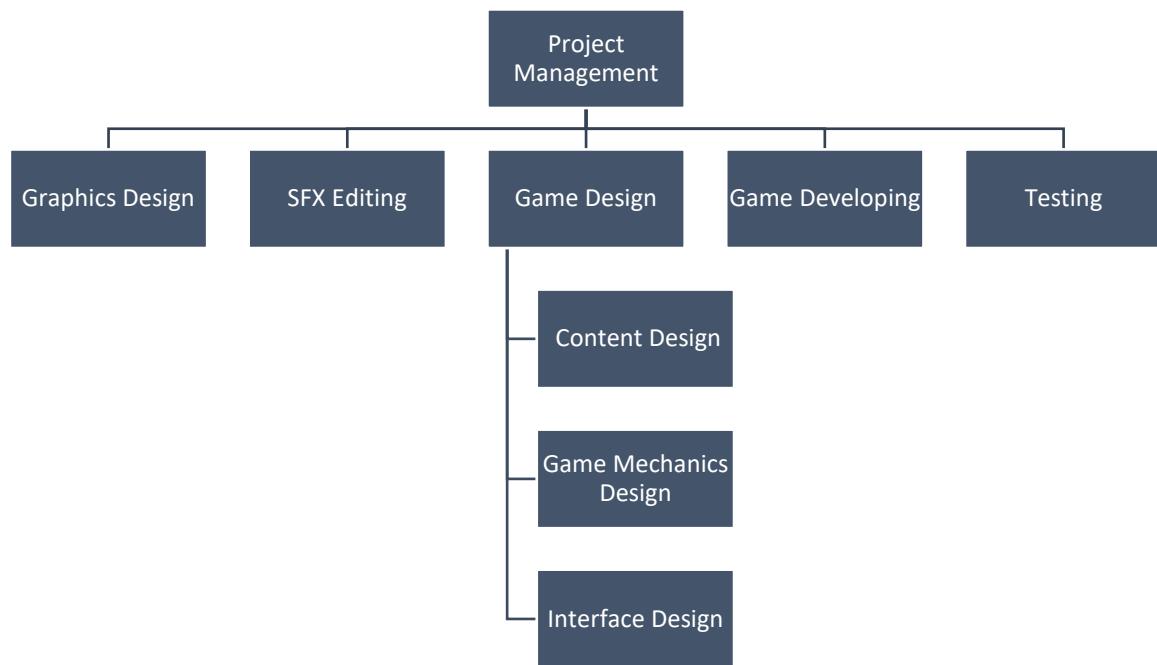
**User:** Players/Gamers

**Project Team:** Berke Can Yaman, Enes Eyüpoğlu, İnanç Haluk Aksoy, Ahmed Qureshi, Mustafa Balta



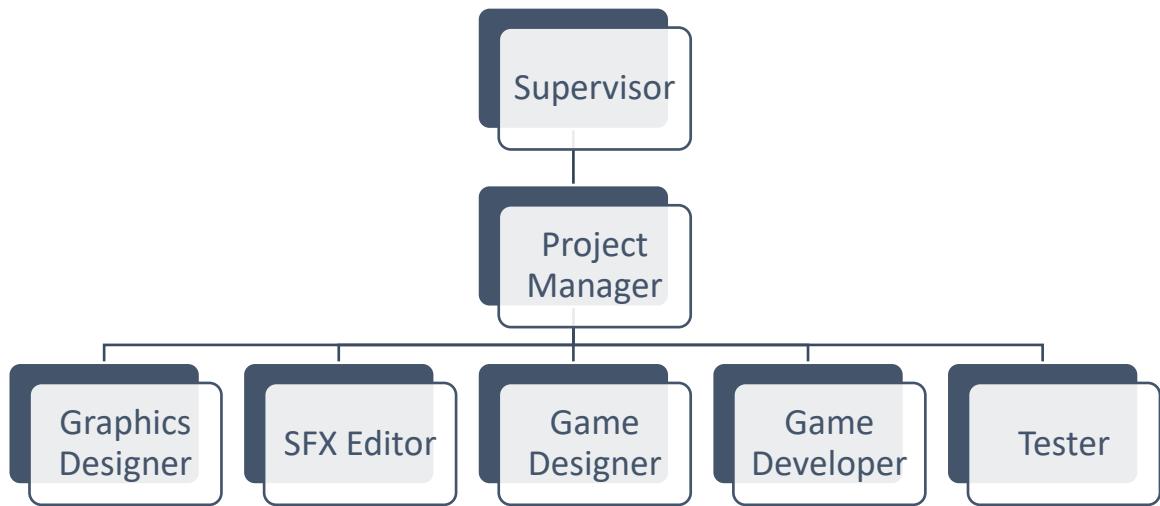
## 4.2 Internal Structure

All the responsibilities/duties will be shared by all of the members of this project except one area. Graphics design work will be done by Berke Can Yaman, because he is the only one who is capable of graphics design. The major parts of our project are graphics design, game design, game development and testing. The application/game will be revised according to the tests and supervisors.



## 4.3 Roles and Responsibilities

### 4.3.1 Roles



### **Game Designer:**

- Determine and define game-play mechanics.
- Create the rules, characters, settings, stories.
- Create documents related to the project before implementation of the project.

### **Game Developer:**

- Translates requirements into complicated but clean and efficient code.
- Troubleshooting code and fixing bugs.

### **Graphics Designer:**

- Conceptualizing visuals based on requirements.
- Creates images and layouts by hand or using design software.

### **Project Manager:**

- Responsible for organizing the team members.
- Checks fulfillment of requirements.

**SFX Editor (Sound effects editor):**

- Responsible for finding, manufacturing, recording, and editing the sound effects.

**Tester:**

- Responsible for ensuring the quality assurance of a product by playing the game in as many ways as possible.

#### 4.3.2 Responsibilities

Name	E-mail	Roles and Responsibilities
Mustafa Balta	mustafa.balta.tr@hotmail.com	Project Manager Game Designer Game Developer Game Tester
Haluk İnanç Aksoy	haluk@ihaksoy.com	Game Designer Game Developer Game Tester
Enes Eyüpoğlu	enes29378@gmail.com	Game Designer Game Developer Game Tester
Berke Can Yaman	berkecan62@gmail.com	Graphics Designer Game Designer

		Game Developer SFX Editor
Ahmed Qureshi		Game Designer Game Developer SFX Editor

## 5. Managerial Process Plan

In this part, the administrative cycle of the BSG (Business Simulation Game) venture is clarified. The zones covered incorporate danger the executives and staffing, needs, suspicions, conditions, limitations.

### 5.1 Start-up Plan

#### 5.1.1 Staffing Plan

The BSG project development is composed of the following:

**BSG Project Manager** - Responsible for: asset the executives, financial plan, timing and faculty identified with IRM venture scope; all parts of the BSG plan, usage,

testing and quality; BSG module mix; And the security of staff and gear. Sufficient preparing to guarantee that all BSG faculty and staff are satisfied; IRM/BSG fills in as task association and contact point. Connects with framework suppliers and other applicable associations; It surveys and affirms all conveyances for the BSG venture; And through the head of the IRM Department.

**BSG Project Leads** - Responsible for the interface with BSG Project Manager; For individual direction and task in your own place; Coding, documentation, arrangement control, unit testing and nature, all things considered; BSG Project Manager to plan and execute the undertaking.

**BSG Programmers** - Responsible for configuration, coding, documentation, arrangement control, unit testing and nature, all things considered; works with the client association and BSG venture lead to plan and execute venture expectations.

**BSG Support Team** - Responsible for utilizing equipment and programming gave by the seller. These duties incorporate working framework uphold, information base organization backing, and advancement apparatus uphold. This degree of help is needed in the creating site. These individuals are on their destinations. These people will answer to and interface with the BSG venture lead at their site.

**Document Production Support** - An individual who will speak to an association liable for specialized guideline, word preparing and archive appropriation for BSG; Works with venture colleagues to create documentation yield.

**Complex Wide Item** - Technical help for programming plan and execution. Lessen, favor, support framework surveys and make an asset for the framework plan and working framework.

**V & V Representative** - An individual who speaks to a hierarchical substance answerable for leading V and V exercises on the BSG.

### 5.1.2 Resource Acquisition Plan

Equipment and programming assets are sufficient for a task of this scale. Be that as it may, to test and circulate the venture, web facilitating is needed to get applications from candidates. Then again, extra PC isn't required, in light of the fact that plan, usage and testing the task will be done from staff's PCs.

### 5.2 Work Plan

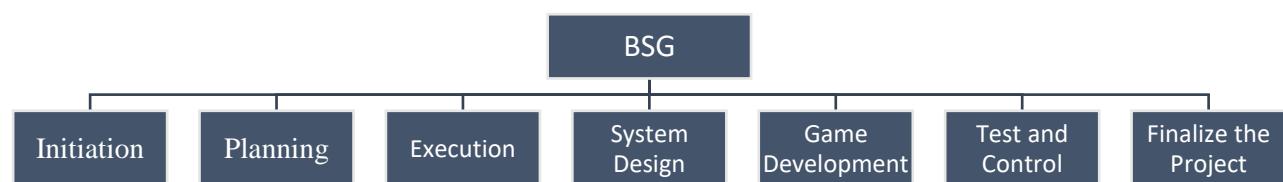
This thing of the SPMP ought to determine business exercises, schedule, assets and spending subtleties for the product venture.

### 5.2.1 Work Activities

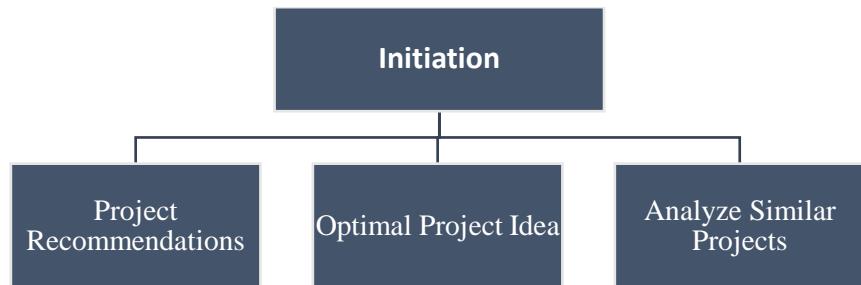
- The essential WBS(Work Breakdown Structures) is on underneath. The task stages are Initiation, Planning, Execution, System Design, Game Development, Test and Control, Finalized the Project.

Execution stages is advancement measure for the task and the its sub-stages are User Requirements, Use Cases, Complete the SRS Document, Submit SRS, Review SRS, Submit the Updated SPMP

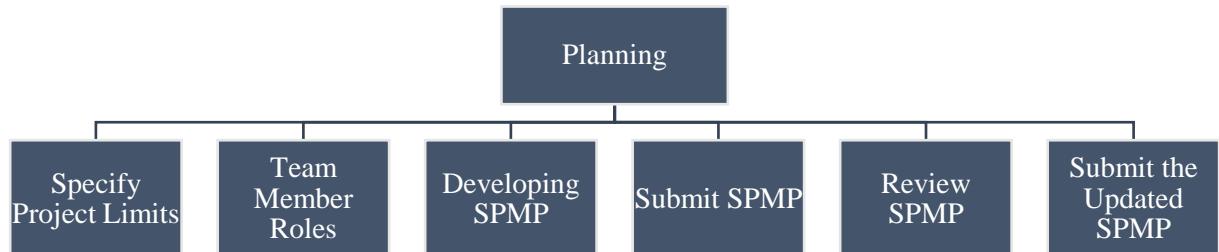
The venture's Work Breakdown Structures are on below.



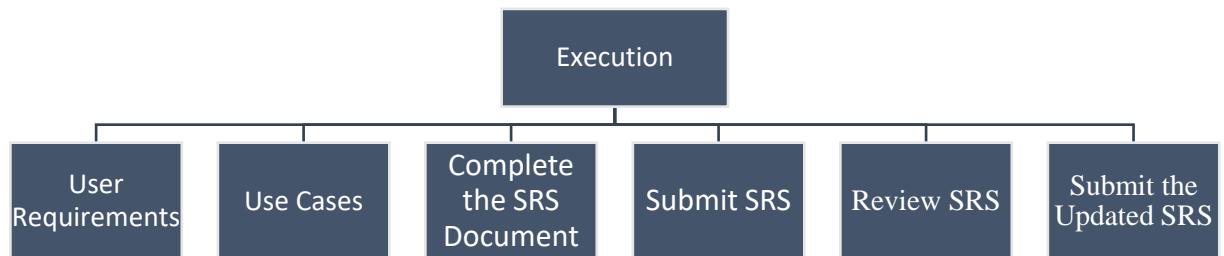
## Initiation Phase Details



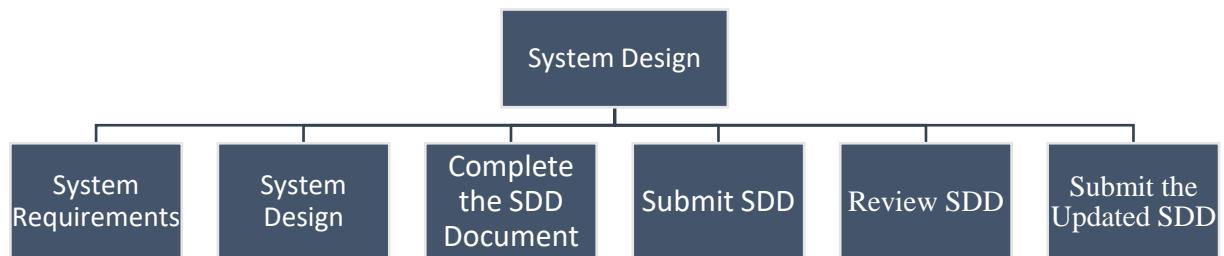
## Planning Phase



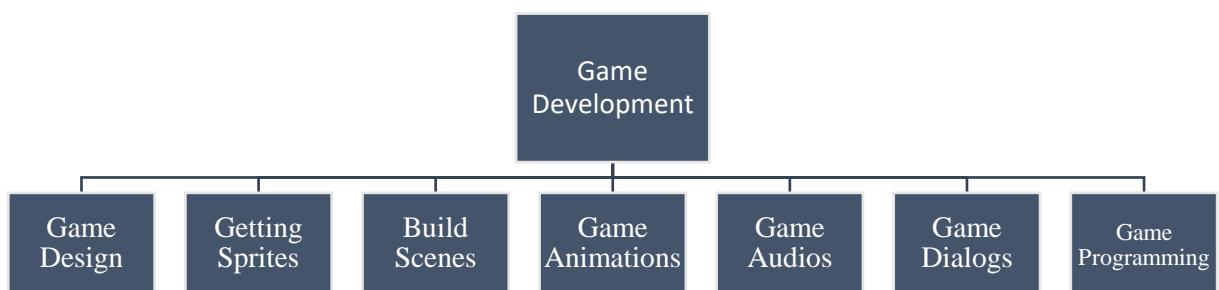
## Execution Phase



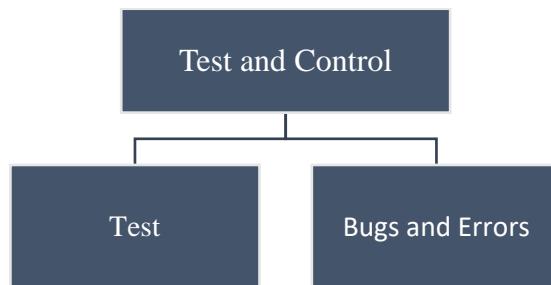
## System Design Phase



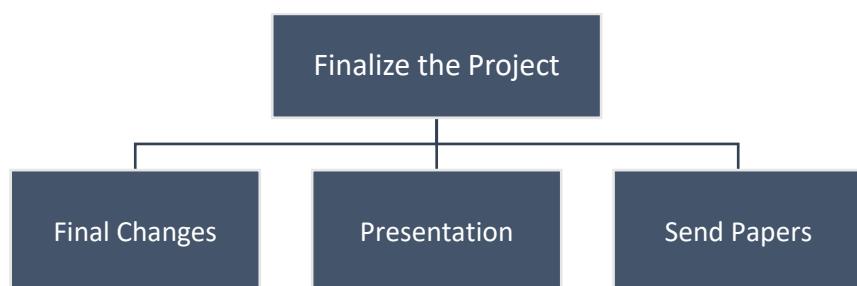
## Game Development Phase



## Test and Control Phase



## Finalize the Project Phase



The work breakdown structure for each activity are given below

<b>Activity Number</b>	1
<b>Activity Code</b>	BSG
<b>Activity Name</b>	Business Simulation Game
<b>Estimated Duration</b>	77 days
<b>Input</b>	-
<b>Outputs</b>	The Project
<b>Predecessor</b>	-
<b>Successor</b>	1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7
<b>Completion Criteria</b>	“Business Simulation Game” software product
<b>Implementation</b>	
Personnel Assigned	Team Members
Starting Date	11 November 2020
Completion Date	15 January 2021

<b>Activity Number</b>	1.1
------------------------	-----

<b>Activity Code</b>	Init
<b>Activity Name</b>	Initiation
<b>Estimated Duration</b>	12 days
<b>Input</b>	-
<b>Outputs</b>	Project details
<b>Predecessor</b>	1
<b>Successor</b>	1.1.1, 1.1.2, 1.1.3
<b>Completion Criteria</b>	All team members submitted their ideas.
<b>Implementation</b>	
Personnel Assigned	Team Members
Starting Date	11 November 2020
Completion Date	23 November 2020

<b>Activity Number</b>	1.1.1
<b>Activity Code</b>	ProRec
<b>Activity Name</b>	Project Recommendations
<b>Estimated Duration</b>	4 days
<b>Input</b>	-
<b>Outputs</b>	Project Ideas

<b>Predecessor</b>	1.1
<b>Successor</b>	-
<b>Completion Criteria</b>	Finding possible project ideas.
<b>Implementation</b>	
Personnel Assigned	Team Members
Starting Date	11 November 2020
Completion Date	15 November 2020

<b>Activity Number</b>	1.1.2
<b>Activity Code</b>	OpProId
<b>Activity Name</b>	Optimal Project Idea
<b>Estimated Duration</b>	4 days

<b>Input</b>	Project Ideas
<b>Outputs</b>	The Optimal Project Idea
<b>Predecessor</b>	1.1
<b>Successor</b>	-
<b>Completion Criteria</b>	Choosing the optimal project idea.
<b>Implementation</b>	
Personnel Assigned	Team Members
Starting Date	15 November 2020
Completion Date	19 November 2020

<b>Activity Number</b>	1.1.3
<b>Activity Code</b>	AnSimPro
<b>Activity Name</b>	Analyze Similar Projects
<b>Estimated Duration</b>	4 days
<b>Input</b>	-
<b>Outputs</b>	Strengthen Ideas
<b>Predecessor</b>	1.1
<b>Successor</b>	-

<b>Completion Criteria</b>	Intensify the optimal project idea.
<b>Implementation</b>	
Personnel Assigned	Team Members
Starting Date	19 November 2020
Completion Date	23 November 2020

<b>Activity Number</b>	1.2
<b>Activity Code</b>	Pl
<b>Activity Name</b>	Planning
<b>Estimated Duration</b>	14 days
<b>Input</b>	-
<b>Outputs</b>	SPMP Document
<b>Predecessor</b>	1
<b>Successor</b>	1.2.1, 1.2.2, 1.2.3, 1.2.4, 1.2.5, 1.2.6
<b>Completion Criteria</b>	SPMP document is created.
<b>Implementation</b>	
Personnel Assigned	Team Members
Starting Date	23 November 2020
Completion Date	2 December 2020

<b>Activity Number</b>	1.2.1
<b>Activity Code</b>	SpPL
<b>Activity Name</b>	Specify Project Limits
<b>Estimated Duration</b>	1 day
<b>Input</b>	-
<b>Outputs</b>	Limits, objectives and goals
<b>Predecessor</b>	1.2
<b>Successor</b>	-
<b>Completion Criteria</b>	Drawing limits and determining objectives and goals.
<b>Implementation</b>	
Personnel Assigned	Team Members
Starting Date	23 November 2020
Completion Date	24 November 2020

<b>Activity Number</b>	1.2.2
<b>Activity Code</b>	TeMeRo
<b>Activity Name</b>	Team Member Roles
<b>Estimated Duration</b>	1 day
<b>Input</b>	-

<b>Outputs</b>	Roles for each member
<b>Predecessor</b>	1.2
<b>Successor</b>	-
<b>Completion Criteria</b>	Determining team members` roles in the project.
<b>Implementation</b>	
Personnel Assigned	Team Members
Starting Date	24 November 2020
Completion Date	25 November 2020

<b>Activity Number</b>	1.2.3
<b>Activity Code</b>	D-SPMP
<b>Activity Name</b>	Developing SPMP
<b>Estimated Duration</b>	3 days
<b>Input</b>	-
<b>Outputs</b>	SPMP Document
<b>Predecessor</b>	1.2
<b>Successor</b>	-
<b>Completion Criteria</b>	Developed SPMP document
<b>Implementation</b>	

Personnel Assigned	Team Members
Starting Date	25 November 2020
Completion Date	28 November 2020

<b>Activity Number</b>	1.2.4
<b>Activity Code</b>	S-SPMP
<b>Activity Name</b>	Submit SPMP
<b>Estimated Duration</b>	1 days
<b>Input</b>	SPMP Document
<b>Outputs</b>	Submitted SPMP Document
<b>Predecessor</b>	1.2
<b>Successor</b>	-
<b>Completion Criteria</b>	Submitting the SPMP document to the instructor
<b>Implementation</b>	
Personnel Assigned	Team Members
Starting Date	28 November 2020
Completion Date	29 November 2020

<b>Activity Number</b>	1.2.5
------------------------	-------

<b>Activity Code</b>	S-SPMP
<b>Activity Name</b>	Review SPMP
<b>Estimated Duration</b>	1 days
<b>Input</b>	Submitted SPMP Document
<b>Outputs</b>	Reviewed SPMP Document
<b>Predecessor</b>	1.2
<b>Successor</b>	-
<b>Completion Criteria</b>	Reviewing the submitted SPMP document
<b>Implementation</b>	
Personnel Assigned	Team Members
Starting Date	29 November 2020
Completion Date	30 November 2020

<b>Activity Number</b>	1.2.6
<b>Activity Code</b>	StheU-SPMP
<b>Activity Name</b>	Submit the Updated SPMP
<b>Estimated Duration</b>	2 days
<b>Input</b>	Reviewed SPMP Document
<b>Outputs</b>	The Final SPMP Document

<b>Predecessor</b>	1.2
<b>Successor</b>	-
<b>Completion Criteria</b>	Submit the Updated SPMP document to the instructor
<b>Implementation</b>	
Personnel Assigned	Team Members
Starting Date	30 November 2020
Completion Date	2 December 2020

<b>Activity Number</b>	1.3
<b>Activity Code</b>	Ex
<b>Activity Name</b>	Execution
<b>Estimated Duration</b>	14 days
<b>Input</b>	-
<b>Outputs</b>	The Final Product
<b>Predecessor</b>	1

<b>Successor</b>	1.3.1, 1.3.2, 1.3.3, 1.3.4, 1.3.5, 1.3.6
<b>Completion Criteria</b>	Finalize the Software
<b>Implementation</b>	
Personnel Assigned	Team Members
Starting Date	2 December 2020
Completion Date	16 December 2020

<b>Activity Number</b>	1.3.1
<b>Activity Code</b>	URe
<b>Activity Name</b>	User Requirements
<b>Estimated Duration</b>	2 days
<b>Input</b>	-
<b>Outputs</b>	User requirements defined
<b>Predecessor</b>	1.3
<b>Successor</b>	-
<b>Completion Criteria</b>	Specifying user requirements
<b>Implementation</b>	
Personnel Assigned	Team Members
Starting Date	2 December 2020

Completion Date	4 December 2020
-----------------	-----------------

<b>Activity Number</b>	1.3.2
<b>Activity Code</b>	UCa
<b>Activity Name</b>	Use Cases
<b>Estimated Duration</b>	2 days
<b>Input</b>	User Requirements
<b>Outputs</b>	Use Cases
<b>Predecessor</b>	1.3
<b>Successor</b>	-
<b>Completion Criteria</b>	Creating use cases
<b>Implementation</b>	
Personnel Assigned	Team Members
Starting Date	4 December 2020
Completion Date	6 December 2020

<b>Activity Number</b>	1.3.3
<b>Activity Code</b>	Cothe-SRS-Do
<b>Activity Name</b>	Complete the SRS Document

<b>Estimated Duration</b>	3 days
<b>Input</b>	Use Cases
<b>Outputs</b>	SRS Document
<b>Predecessor</b>	1.3
<b>Successor</b>	-
<b>Completion Criteria</b>	Developed SRS document
<b>Implementation</b>	
Personnel Assigned	Team Members
Starting Date	6 December 2020
Completion Date	9 December 2020

<b>Activity Number</b>	1.3.4
<b>Activity Code</b>	S-SRS
<b>Activity Name</b>	Submit SRS
<b>Estimated Duration</b>	3 days
<b>Input</b>	SRS Document
<b>Outputs</b>	Submitted SRS Document
<b>Predecessor</b>	1.3

<b>Successor</b>	-
<b>Completion Criteria</b>	Submitting the SRS document to the instructor
<b>Implementation</b>	
Personnel Assigned	Team Members
Starting Date	9 December 2020
Completion Date	12 December 2020

<b>Activity Number</b>	1.3.5
<b>Activity Code</b>	S-SRS
<b>Activity Name</b>	Review SRS
<b>Estimated Duration</b>	2 days
<b>Input</b>	Submitted SRS Document
<b>Outputs</b>	Reviewed SRS Document
<b>Predecessor</b>	1.3
<b>Successor</b>	-
<b>Completion Criteria</b>	Reviewing the submitted SRS document
<b>Implementation</b>	
Personnel Assigned	Team Members
Starting Date	12 December 2020

Completion Date	14 December 2020
-----------------	------------------

<b>Activity Number</b>	1.3.6
<b>Activity Code</b>	StheU-SRS
<b>Activity Name</b>	Submit the Updated SRS
<b>Estimated Duration</b>	2 days
<b>Input</b>	Reviewed SRS Document
<b>Outputs</b>	The Final SRS Document
<b>Predecessor</b>	1.3
<b>Successor</b>	-
<b>Completion Criteria</b>	Submit the Updated SRS document to the instructor
<b>Implementation</b>	
Personnel Assigned	Team Members
Starting Date	14 December 2020
Completion Date	16 December 2020

<b>Activity Number</b>	1.4
<b>Activity Code</b>	SysDes
<b>Activity Name</b>	System Design

<b>Estimated Duration</b>	10000 Days
<b>Input</b>	-
<b>Outputs</b>	The Final SDD Document
<b>Predecessor</b>	1
<b>Successor</b>	1.4.1, 1.4.2, ...
<b>Completion Criteria</b>	Creating The Final SDD Document
<b>Implementation</b>	
Personnel Assigned	Team Members
Starting Date	16 December 2020
Completion Date	

<b>Activity Number</b>	1.4.1
<b>Activity Code</b>	SysReq
<b>Activity Name</b>	System Requirements
<b>Estimated Duration</b>	2 Days
<b>Input</b>	-
<b>Outputs</b>	System requirements defined
<b>Predecessor</b>	1.4
<b>Successor</b>	-

<b>Completion Criteria</b>	Defining system requirements
<b>Implementation</b>	
Personnel Assigned	Team Members
Starting Date	16 December 2020
Completion Date	18 December 2020

<b>Activity Number</b>	1.4.2
<b>Activity Code</b>	SysDe
<b>Activity Name</b>	System Design
<b>Estimated Duration</b>	3 Days
<b>Input</b>	-
<b>Outputs</b>	Er Diagram
<b>Predecessor</b>	1.4
<b>Successor</b>	-
<b>Completion Criteria</b>	The system analyzed, ER diagram created
<b>Implementation</b>	
Personnel Assigned	Team Members
Starting Date	18 December 2020
Completion Date	21 December 2020

<b>Activity Number</b>	1.4.3
<b>Activity Code</b>	Cothe-SDD-Do
<b>Activity Name</b>	Complete the SDD Document
<b>Estimated Duration</b>	3 days
<b>Input</b>	-
<b>Outputs</b>	SDD Document
<b>Predecessor</b>	1.4
<b>Successor</b>	-
<b>Completion Criteria</b>	Developed SDD document
<b>Implementation</b>	
Personnel Assigned	Team Members
Starting Date	21 December 2020
Completion Date	24 December 2020

<b>Activity Number</b>	1.4.4
<b>Activity Code</b>	S-SDD
<b>Activity Name</b>	Submit SRS

<b>Estimated Duration</b>	3 days
<b>Input</b>	SRS Document
<b>Outputs</b>	Submitted SRS Document
<b>Predecessor</b>	1.4
<b>Successor</b>	-
<b>Completion Criteria</b>	Submitting the SDD document to the instructor
<b>Implementation</b>	
Personnel Assigned	Team Members
Starting Date	24 December 2020
Completion Date	27 December 2020

<b>Activity Number</b>	1.4.5
<b>Activity Code</b>	S-SDD
<b>Activity Name</b>	Review SDD
<b>Estimated Duration</b>	2 days
<b>Input</b>	Submitted SDD Document
<b>Outputs</b>	Reviewed SDD Document
<b>Predecessor</b>	1.4
<b>Successor</b>	-

<b>Completion Criteria</b>	Reviewing the submitted SDD document
<b>Implementation</b>	
Personnel Assigned	Team Members
Starting Date	27 December 2020
Completion Date	29 December 2020

<b>Activity Number</b>	1.4.6
<b>Activity Code</b>	StheU-SDD
<b>Activity Name</b>	Submit the Updated SDD
<b>Estimated Duration</b>	2 days
<b>Input</b>	Reviewed SDD Document
<b>Outputs</b>	The Final SDD Document
<b>Predecessor</b>	1.4
<b>Successor</b>	-
<b>Completion Criteria</b>	Submit the Updated SDD document to the instructor
<b>Implementation</b>	
Personnel Assigned	Team Members
Starting Date	29 December 2020
Completion Date	1 January 2021

<b>Activity Number</b>	1.5
<b>Activity Code</b>	GaDev
<b>Activity Name</b>	Game Development
<b>Estimated Duration</b>	70 days
<b>Input</b>	-
<b>Outputs</b>	Completed Game
<b>Predecessor</b>	1
<b>Successor</b>	1.5.1, 1.5.2, 1.5.3, 1.5.4, 1.5.5, 1.5.6, 1.5.7
<b>Completion Criteria</b>	Complete the “Business Simulation Game”
<b>Implementation</b>	
Personnel Assigned	Team Members
Starting Date	1 January 2021
Completion Date	2021

<b>Activity Number</b>	1.5.1
<b>Activity Code</b>	GaDes
<b>Activity Name</b>	Game Design
<b>Estimated Duration</b>	5 days
<b>Input</b>	-

<b>Outputs</b>	Game Design Ideas
<b>Predecessor</b>	1.5
<b>Successor</b>	-
<b>Completion Criteria</b>	Members decide the optimal design for the game.
<b>Implementation</b>	
Personnel Assigned	Team Members
Starting Date	1 January 2021
Completion Date	6 January 2021

<b>Activity Number</b>	1.5.2
<b>Activity Code</b>	GetSp
<b>Activity Name</b>	Getting Sprites
<b>Estimated Duration</b>	7 days
<b>Input</b>	-
<b>Outputs</b>	Sprites that are ready to use
<b>Predecessor</b>	1.5
<b>Successor</b>	-
<b>Completion Criteria</b>	Sprites which created or imported are ready to use.

<b>Implementation</b>	
<b>Personnel Assigned</b>	Team Members
<b>Starting Date</b>	6 January 2021
<b>Completion Date</b>	13 January 2021
<b>Activity Number</b>	1.5.3
<b>Activity Code</b>	BuSce
<b>Activity Name</b>	Build Scenes
<b>Estimated Duration</b>	5 days
<b>Input</b>	-
<b>Outputs</b>	Primary Scenes
<b>Predecessor</b>	1.5
<b>Successor</b>	-
<b>Completion Criteria</b>	Primary scenes built
<b>Implementation</b>	
<b>Personnel Assigned</b>	Team Members
<b>Starting Date</b>	13 January 2021
<b>Completion Date</b>	18 January 2021

<b>Activity Number</b>	1.5.4
<b>Activity Code</b>	GaAnim

<b>Activity Name</b>	Game Animations
<b>Estimated Duration</b>	5 days
<b>Input</b>	-
<b>Outputs</b>	Animations
<b>Predecessor</b>	1.5
<b>Successor</b>	-
<b>Completion Criteria</b>	Game animations created.
<b>Implementation</b>	
<b>Personnel Assigned</b>	Team Members
<b>Starting Date</b>	18 January 2021
<b>Completion Date</b>	23 January 2021
<b>Activity Number</b>	1.5.5
<b>Activity Code</b>	GaAud
<b>Activity Name</b>	Game Audios
<b>Estimated Duration</b>	7 days
<b>Input</b>	-
<b>Outputs</b>	Game Sounds
<b>Predecessor</b>	1.5
<b>Successor</b>	-

<b>Completion Criteria</b>	Game audios and music's created.
<b>Implementation</b>	
Personnel Assigned	Team Members
Starting Date	23 January 2021
Completion Date	30 January 2021

<b>Activity Number</b>	1.5.6
<b>Activity Code</b>	GaDia
<b>Activity Name</b>	Game Dialogs
<b>Estimated Duration</b>	14 days
<b>Input</b>	-
<b>Outputs</b>	Dialogs
<b>Predecessor</b>	1.5
<b>Successor</b>	-
<b>Completion Criteria</b>	Game dialogs created.
<b>Implementation</b>	
Personnel Assigned	Team Members
Starting Date	30 January 2021
Completion Date	13 February 2021

<b>Activity Number</b>	1.5.7
<b>Activity Code</b>	GaProg
<b>Activity Name</b>	Game Programming
<b>Estimated Duration</b>	21 days
<b>Input</b>	Script Files
<b>Outputs</b>	Updated Script Files
<b>Predecessor</b>	1.5
<b>Successor</b>	-
<b>Completion Criteria</b>	-
<b>Implementation</b>	
Personnel Assigned	Team Members
Starting Date	13 February 2021
Completion Date	6 March 2021

<b>Activity Number</b>	1.6
<b>Activity Code</b>	ContAndTest
<b>Activity Name</b>	Test and Control
<b>Estimated Duration</b>	21 days

<b>Input</b>	Current Software Product
<b>Outputs</b>	Updated Software Product
<b>Predecessor</b>	1
<b>Successor</b>	1.6.1, 1.6.2
<b>Completion Criteria</b>	Test and improve the software product
<b>Implementation</b>	
Personnel Assigned	Team Members
Starting Date	15 March 2021
Completion Date	5 April 2021

<b>Activity Number</b>	1.6.1
<b>Activity Code</b>	Test
<b>Activity Name</b>	Test
<b>Estimated Duration</b>	7 days
<b>Input</b>	Current Software Product
<b>Outputs</b>	Testted Software Product
<b>Predecessor</b>	1.6
<b>Successor</b>	-
<b>Completion Criteria</b>	Current Software Product is tested.

<b>Implementation</b>	
Personnel Assigned	Team Members
Starting Date	15 March 2021
Completion Date	22 March 2021

<b>Activity Number</b>	1.6.2
<b>Activity Code</b>	BugAndEr
<b>Activity Name</b>	Bugs and Errors
<b>Estimated Duration</b>	14 days
<b>Input</b>	Current Software Product
<b>Outputs</b>	Updated Software Product
<b>Predecessor</b>	1.6
<b>Successor</b>	-
<b>Completion Criteria</b>	Bugs and Errors are fixed
<b>Implementation</b>	
Personnel Assigned	Team Members
Starting Date	22 March 2021
Completion Date	5 April 2021

<b>Activity Number</b>	1.7
<b>Activity Code</b>	FinThePrj
<b>Activity Name</b>	Finalize the Project
<b>Estimated Duration</b>	57 days
<b>Input</b>	Current Project
<b>Outputs</b>	Finished Project
<b>Predecessor</b>	1.6
<b>Successor</b>	-
<b>Completion Criteria</b>	Business Simulation Game finished.
<b>Implementation</b>	
Personnel Assigned	Team Members
Starting Date	5 April 2021
Completion Date	1 June 2021

<b>Activity Number</b>	1.7
<b>Activity Code</b>	FinThePrj

<b>Activity Name</b>	Finalize the Project
<b>Estimated Duration</b>	57 days
<b>Input</b>	Current Project
<b>Outputs</b>	Finished Project
<b>Predecessor</b>	1
<b>Successor</b>	1.7.1, 1.7.2, 1.7.3
<b>Completion Criteria</b>	“Business Simulation Game” finished.
<b>Implementation</b>	
Personnel Assigned	Team Members
Starting Date	5 April 2021
Completion Date	2 June 2021

<b>Activity Number</b>	1.7.1
<b>Activity Code</b>	FinChan
<b>Activity Name</b>	Final Changes
<b>Estimated Duration</b>	21 days
<b>Input</b>	Current Project
<b>Outputs</b>	Finished Project

<b>Predecessor</b>	1.7
<b>Successor</b>	-
<b>Completion Criteria</b>	Making final changes on the BSG
<b>Implementation</b>	
Personnel Assigned	Team Members
Starting Date	5 April 2021
Completion Date	31 May 2021

<b>Activity Number</b>	1.7.2
<b>Activity Code</b>	Presentation
<b>Activity Name</b>	Presentation
<b>Estimated Duration</b>	1 days
<b>Input</b>	-
<b>Outputs</b>	Presentation of the project
<b>Predecessor</b>	1.7
<b>Successor</b>	-
<b>Completion Criteria</b>	The project has presented.
<b>Implementation</b>	
Personnel Assigned	Team Members

Starting Date	31 May 2021
Completion Date	1 June 2021

<b>Activity Number</b>	1.7.3
<b>Activity Code</b>	SendPa
<b>Activity Name</b>	Send Papers
<b>Estimated Duration</b>	1 days
<b>Input</b>	Finished Project
<b>Outputs</b>	-
<b>Predecessor</b>	1.7
<b>Successor</b>	-
<b>Completion Criteria</b>	Project documents sent to MIS Department.

### 5.2.2 Schedule Allocation

The timetable assignment is on Graphic 1. Gantt diagram is utilized for the documentation.

### 5.2.3 Resource Allocation

The asset assignment can't be assessed as a result of the insufficient experience. It will be added next update.

### 5.2.4 Budget Allocation

Staff pay and advertising are the principle costs for this undertaking. The staff payment

12.000₺\* by the assessment. The hour of this task is assessed as 6 months. The general financial plan is assessed as 100.000₺ for a year and month to month fixed expense is about 5.000₺.

\*Staff payment = (S \* K \* Z) + R

S (Your hourly wage at work) = 14.52,

K (Your hour coefficient)=1.5,

Z (How long do you plan to finish the project) = 505,

R (Additional fee for possible risks)= 1000

Staff payment = (14.52 \* 505 \* 1.5 ) + 1000 = 12.000 ₺

The items for the budget are as follows:

Name	Duration	Cost
<b>Initiation</b>	<b>10</b>	<b>£10,500.00</b>
<b>Planning</b>	<b>20</b>	<b>£5,000.00</b>
<b>Execution</b>	<b>50</b>	<b>£25,000.00</b>
<b>Game Design</b>	<b>8</b>	<b>£15,000.00</b>
<b>Implementation of the Game</b>	<b>5</b>	<b>£10,000.00</b>
<b>Testing</b>	<b>7</b>	<b>£4,000.00</b>
<b>Maintenance</b>	<b>2</b>	<b>£10,000.00</b>
<b>Evaluation</b>	<b>4</b>	<b>£2,500.00</b>
<b>Conclusion</b>	<b>2</b>	<b>£3,000.00</b>
<b>Marketing</b>	<b>6</b>	<b>£15,000.00</b>
<b>BSG</b>	<b>114</b>	<b>£100,000.00</b>

## 5.3 Control Plan

### 5.3.1 Requirement Control Plan

Necessities for the task is put on the SRS archive and not expected revolutionary changes after the SRS finished. The progressions will be assessed per effect of the change for the undertaking. On the off chance that this prerequisite worthy, the timetable and assets will be doled out once more.

### 5.3.2 Schedule Control Plan

Timetable will be checked week after week at the group gatherings and contrasted with the arranged timetable. In the event that the contrasts among arranged and real is more noteworthy than 21 days, the group will be cautioned by this circumstance. Also, perhaps additional time required.

### 5.3.3 Budget Control Plan

Budget will be checked week by week. At the last work day of the week, Compared to the spending plan and contrasts if more prominent than not-assessed cost (5%). The spending will be redistributed over the errands.

### 5.3.4 Quality Control Plan

The pointers for the nature of the task is code quality, documentation, bug proportion for the line of code. In the event that the product covers all the necessities at the utilitarian and non-practical zone, the venture will be finished the acknowledgment assessment. The acknowledgment test is the significant control plan for the task.

### 5.3.5 Reporting Plan

The reports for the venture will be week after week and constrained by the acquirer and undertaking supervisor.

## 5.4 Risk Management Plan

The accompanying improvement hazards have been recognized for BSG. The principal gathering of dangers will be the most extreme and will keep them from meeting extended venture needs.

### **Equipment and programming are not accessible for advancement**

All equipment and programming acquisitions will be done from one PC to alleviate this likely danger.

### **Equipment and programming not accessible for preparing.**

We have given the option to obtain equipment and programming to an individual who has attempted the buy to decrease this possible danger.

### **There isn't sufficient consultancy for programming.**

The product will be chosen from organizations with demonstrated records.

### **Project financing removed.**

Assisting with diminishing this chance we will educate the funding organization frequently.

### **Equipment or programming preparing was given to the task past the point of no return,**

To decrease this possible danger, we have assessed the preparation needs of the designer staff and the framework client. A preparation plan was made to help total the preparation.

### **Innovative difficulties (for example equipment or framework programming issues).**

Once more, we have allotted a buy pro equipment and programming to lessen this expected danger.

### **Assignment asset necessities are thought little of.**

The extent of the undertaking has been characterized and assets have been doled out. Changes made will cause insufficiency of assets. This will be dealt with by assessment of assets consistently.

### **Inability to follow and change the asset utilization that really reacts**

Synopsis reports will be composed to screen asset utilization. This detailing will assist with improving perceivability at elevated levels likewise will assist with fixing irregularities!

### **Programming issue couldn't be settled as expected.**

The Technical Project Manager will screen and resolve issues on a mission.

### **The acquirer adjusts his perspective on the necessities, or there is contradiction about the prerequisites understanding.**

It should be made extremely obvious to the acquirer that after a specific date the necessities can't change any longer.

## **5.5 Closeout Plan**

The task will have an end in May 2021s. Around then, the group will:

- The task will be put away and chronicled at hard drive disc (HDD) or other relentless stockpiling units. The undertaking source code, documentations, minutes, meeting notices and so forth

- Eventual outcome will be facilitated at cloud and conveyed with establishment report
- Make introduction for the task to the jury.
- Last report and closeout of the venture.
- Exercise educated, insight and information sharing and the exhibition measurements and self-assessment for the task for to improve execution of future ventures.

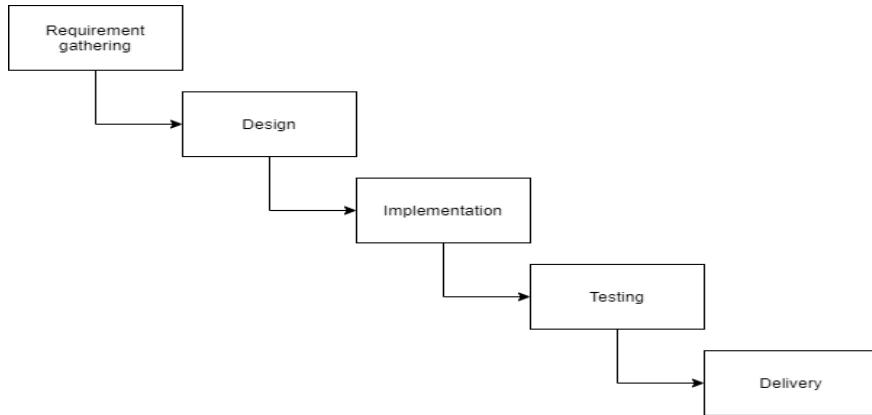
## 6. Technical Process Plan

### 6.1 The Process

The game development process will be organized and planned by a supervisor and those plans are to be executed by multiple teams. Each team will be assigned a task, based on team members' skills and preferences. The tasks will range from back-end software development to front-end UI design. Both teams will have to work closely together because the integrity of UI and the back-end are crucial to the games' success. Teams will be responsible for delivering appropriate documentation respective to their assigned tasks.

### 6.2 Methodology

The game will be developed with a mix of Waterfall and SCRUM methodologies. Firstly, the initial game will be developed using a traditional waterfall methodology. Initial planning, designing, developing and testing is more suitable for a waterfall methodology.

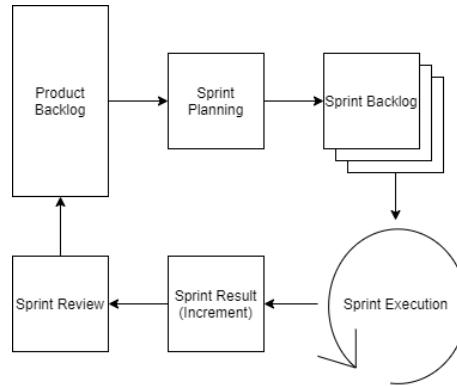


*Waterfall Methodology*

In the requirement gathering phase, multiple documents (SPMP, SRS and SDD) will be created. Using those initial requirements gathered the team will design the game during the design phase. Sticking to the design, the game will be developed into a working software during implementation phase. Testing phase consists of multiple tests that the team and other people will do to help fix potential bugs and defectiveness of the software. After all the phases are finished, the initial game will be ready.

Later increments to the game will be done using a SCRUM methodology. Since the project needs to be finished in a relative short period of time and teams are small, SCRUM methodology will be used if and when a incrementation is needed to the initial game. The team will be divided into smaller teams and each

team will divide the end goals into smaller goals. Therefore, developers should collaborate heavily and daily.



*SCRUM Methodology*

## 6.3 Tools and Infrastructure

### 6.3.1 Tools

The tools needed for game development can be divided into three groups,

#### 1. Game Engine

For this project, Unity will be used as the game engine. It has a lot of advantages over other game engines, mainly advantage is that is easier to use, and the games created are targeted at multiple platforms. If and when the game is going to be published into a new ecosystem Unity will provide much more flexibility.

#### 2. Coding

Unity supports C#, an industry-standard language with some similarities to Java or C++. Therefore, scripts that will control custom actions and interaction within the game will be written in C#. Developers are free to use any code editor they see fit.

#### 3. Art and Design

Adobe Photoshop and Illustrator will be the choice for digital art and design. Although there is numerus other software, Photoshop offers wider range of well-established tools and filters and will be used for painting

and jobs that require precise pixel placement, stroke-based painting and filters. Illustrator will be used to make vector art such as UI elements and head up textures as a vector package.

### 6.3.2 Infrastructure

The game will not have a multiplayer feature; therefore, a powerful hosting server is not necessary. However, to push further updates and patches presence of a hosting server is needed. The game will be shipped by internet and installed directly on the users' device.

## 6.4 Version Control

Teams will be using Git for version control and configuration management. GitHub will be used for storage and collaboration purposes. Each team will have a manager responsible for their respective branch.

Git is a source control system that allows teams to track versions of their work in project containers called repositories. GitHub is a remote hosting system for sharing repositories.

If and when a team member has made changes to the project, he or she will push the changes to the GitHub repository as shown in the figure.



*Pushing changes to GitHub*

If and when a team member wishes to access the up to date version of the project, he or she fetches the current version of the repository on GitHub and continues to work on the up to date version.



### *Pulling and comparing changes from GitHub*

## 6.5 Product Acceptance Tasks

### 6.5.1 Product Acceptance Criteria

The game should be playable, enjoyable and according to the storylines and design guidelines previously determined. The game should be developed in such a way that it has little to no bugs and deficits. The development team will do extensive and in-depth testing of the product. All use cases will be tested.

Following checklist is to be followed for the finalizing tests.

- All the previously determined requirements are correctly and purposefully implemented into the end products and modules.
- All modules have passed their respective unit tests.
- All modules are integrated and working smoothly, and integration tests have passed successfully.
- Documentation for the product is completed and includes all features.

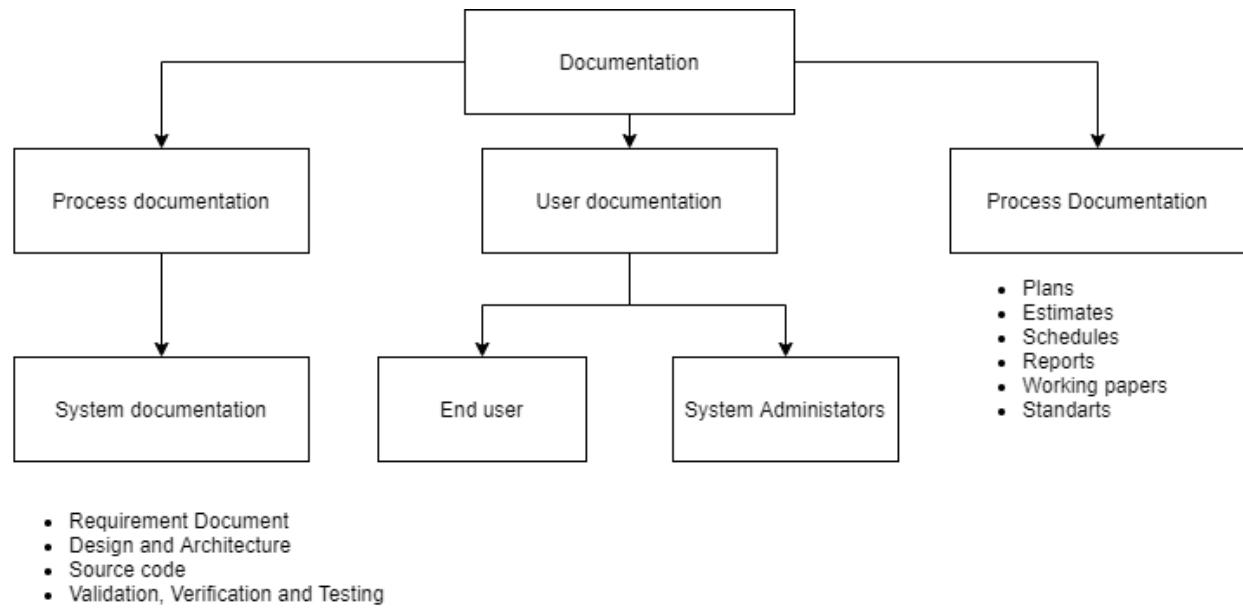
### 6.5.2 Schedule

The development process will follow the schedule that is previously determined and mentioned above.

### 6.5.3 Documentation

The game will have thorough documentation both on technical aspects of the software and the user interaction. Documentation for the user end will be done by the design team, who have designed the storyline and the generally how the

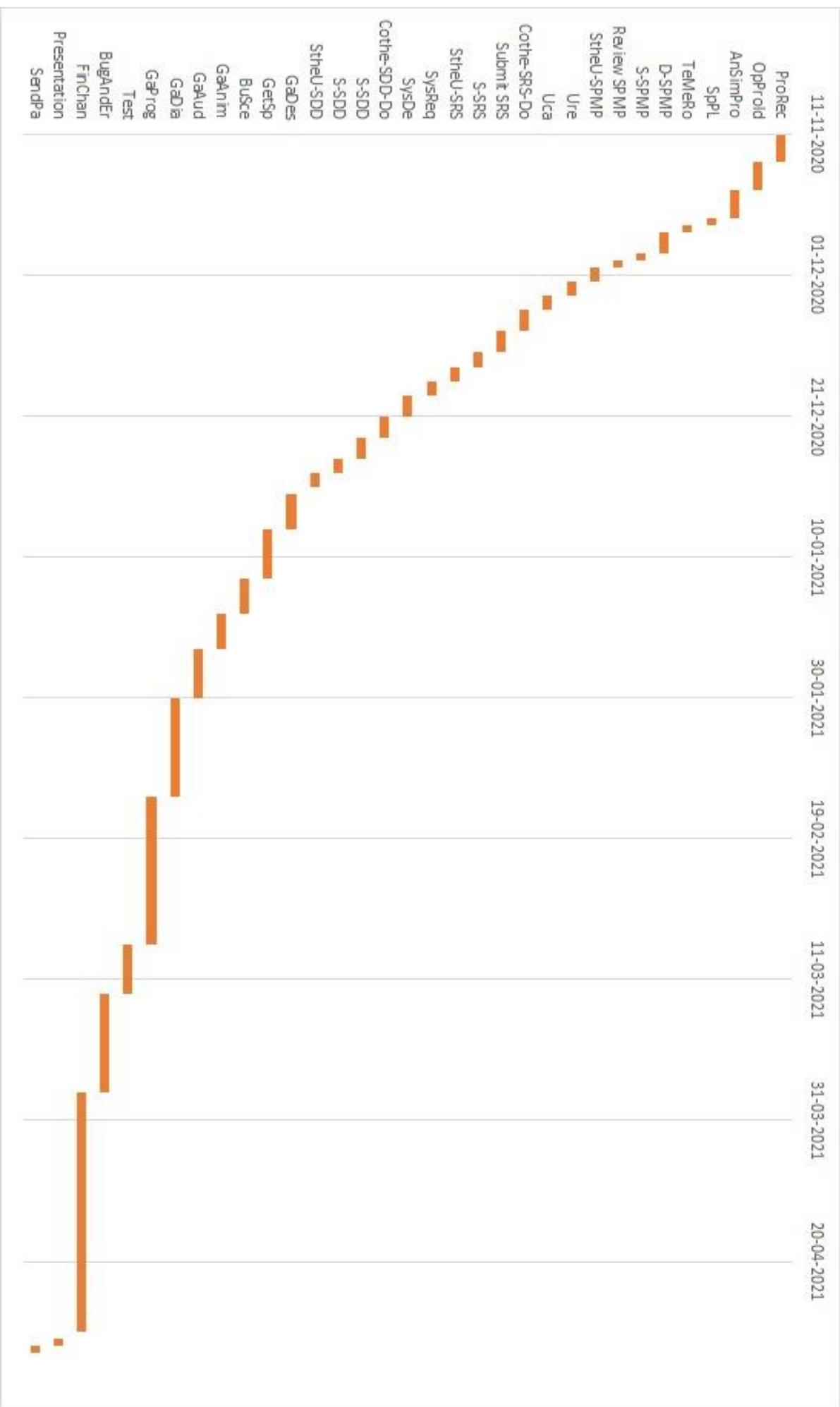
game operates. The technical teams will be responsible for the technical documentation. The documents will be as shown in the figure.



*Documentation*

## 7. APPENDIXES

### 7. Appendix A: Gantt Chart



ANKARA YILDIRIM BEYAZIT UNIVERSITY  
BUSINESS SCHOOL  
DEPARTMENT OF MANAGEMENT INFORMATION  
SYSTEMS

**Business Simulation Game**  
**Software Project Management Plan**

*Acquirer:*

*Asst. Prof. İhsan Tolga Medeni*

*Research Associate Dilan Özcan*

*Prepared by*

*Mustafa Balta*

*Enes Eyüboğlu*

*Berke Can Yaman*

*İnanç Haluk Aksøy*

*Ahmed Qureshi*

## Version History

<b>Version #</b>	<b>Implemented By</b>	<b>Revision Date</b>	<b>Reason</b>
1.0	<i>BSG Team</i>	<i>11.01.20 21</i>	<i>Initial version</i>
1.1	<i>BSG Team</i>	<i>22.01.20 21</i>	<i>Update on second part</i>

## Table of Contents

1.	Introduction.....	5
1.3	Purpose.....	5
1.4	Scope .....	5
1.5	Definitions, Acronyms and Abbreviations.....	6
1.6	References.....	6
1.7	Overview.....	7
2.	Overall Description.....	7
2.1	Product Perspective .....	7
2.2	System Interface.....	8
2.3	User Interface.....	8
2.4	Hardware Interfaces and Requirements .....	8
2.5	Software Interfaces .....	9
2.6	Memory Constraints.....	9
2.7	Operations.....	9
2.8	Game Functions.....	9
2.8.1	User .....	9
2.8.2	User characteristics .....	10
2.9	Constraints .....	10
2.9.1	Regulatory policies .....	10
2.9.2	Hardware limitation .....	10
2.9.3	Language.....	10
2.9.4	Reliability .....	10
2.9.5	Safety.....	10
2.10	Assumptions and Dependencies .....	10
2.11	User documentation.....	11
3.	Specific Requirements.....	11
3.1	External Interface Requirements .....	11
3.1.1	Main Screen.....	11
3.1.2	Marketing Window.....	13
3.1.3	TV Channels Window.....	14
3.1.4	Products Window .....	15
3.1.5	News Window.....	16

3.1.6	Skill Tree Window .....	16
3.1.7	Stock Exchange Window.....	18
3.1.8	Balance Sheet Window.....	19
3.1.9	Software Development Window .....	20
3.1.10	Create New Product .....	21
3.1.11	Enhance Software.....	22
3.1.12	Fix Bugs.....	23
3.1.13	Add a new feature!.....	23
3.1.14	Port to new OS.....	24
3.1.15	Take on Contracts.....	25
3.1.16	Products Window .....	26
3.1.17	Escape Window .....	26
3.1.18	Bank Window .....	28
3.1.19	Take Credit Window .....	29
3.1.20	Invest Money Window .....	30
3.1.21	Social Media Platforms Window .....	32
	.....	32
3.1.22	Billboards Window .....	33
3.1.23	Office Window.....	34
3.1.24	Furnitures Window.....	35
3.1.25	Rent a New Office Windows.....	36
3.1.26	Employees Window.....	39
3.1.27	Improve Window .....	40
3.1.28	Manage Window .....	41
3.1.29	Hardware Interfaces.....	41
3.1.30	Software Interfaces .....	41
3.2	Design Constraints.....	41
3.3	Software system attributes .....	42
3.3.1	Reliability .....	42
3.3.2	Availability .....	42
3.3.3	Security.....	42
3.3.4	Maintainability .....	42
3.3.5	Portability .....	42
3.4	Performance Requirements .....	43

# 1. Introduction

This document introduces the reader to the single player, business simulation game. In introduction chapter, purpose, scope, definitions about the project will be explained and an overview of the document will be given.

## 1.3 Purpose

The aim of the BSG project is to produce a kind of business simulation game. In this simulation game, the player will manage a software company and try to be successful by producing software.

This document will be an explanation of what the BSG project is. The purpose of this document is to explain the software requirements of the BSG game project. The preparation of this document is to assist the developers, instructors, and testers of the project.

## 1.4 Scope

The fundamental goal of this project is to build a single player business simulation game. The targeted features of the project can be listed as below:

- To entertain the users of the game.
- To have well written document.
- To have music and sound effects that appeal to the genre and target users of the game.

If we make a statement about the game content, we can state the following.

In this game, the player starts the game as a manager of a software company. In the game, the player will create software products or do software related business with other companies or individuals.

Successful jobs and products will increase the value of the company. The ultimate goal of this game is to be the most valuable company on the stock market. Each related section of the game is presented to the user under a different window. These sections are as follows, Software Development, Office, Bank, Skill Tree, News, Employee, Products, Marketing, and Stock Exchange. These parts will be explained in detail with user interfaces in the third chapter of this document.

## 1.5 Definitions, Acronyms and Abbreviations

BSG: Business Simulation Game

SDD: Software Design Document

SPMP: Software Project Management Plan

SRS: Software Requirement Specification

UC: Use Case

UI: User Interface

IEEE: The Institute of Electrical and Electronics Engineers, Inc

Player: A person who plays the game.

Unity3D: A game engine which provide to design and develop a game for PC, mobile and console platforms.

Single-Player: A game that can only be played by one person.

## 1.6 References

- *IEEE Std 830-1998 - IEEE Recommended Practice for Software Requirements Specifications.*

- *BSG Software Project Management Plan, 2020*

## 1.7 Overview

This software requirement document is planned according to IEEE Std. 830-1998. The SRS document is consisting of three main parts: introduction, overall description, and specific requirement section.

First part of the document gives explanation to the project and answers what is objectives, scope of this project. Second part of the project informs the audience about the project's overall description and explains the main elements which have influence in system requirements.

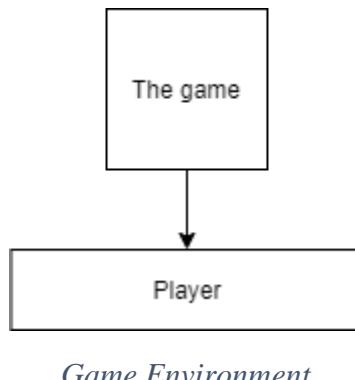
## 2. Overall Description

This part aims to give a full and overall description of the game.

### 2.1 Product Perspective

The game makes full use of the powerful tools that is available at the moment. Unity will be used as the main game engine. It offers very fast game development speeds. Unity follows agile game creation and facilitates quick prototyping and continuous releases. Easy and quick import process of resources used in a game leads to an optimized unified Assets pipeline and supports most image, audio, video and text formats. It also has an excellent editor which supports JavaScript and C# scripts. Superior support for de-bugging and tweaking as all the game variables are displayed during gameplay which enables systematic tweaking and de-debugging processes at runtime. Unity has vast number of communities and marketplaces which offer wide built components for sound, physics, rendering, controls etc. Multi-platform unity game development and deployment platforms for Consoles, Desktops, Browsers and Mobiles.

The game will be standalone executable platform, which will work on large amount of desktop devices. This platform will be used to as the main platform that hosts all the other functions a user can interact with. This game will be used to simulate a software company. Environment of the game is simple. There is only one type of user which can interact with the system.



## 2.2 System Interface

The game will not and has not be integrated into any other web application and or other type of application to work properly.

## 2.3 User Interface

The game will make use of a complex and pre-designed interface which will be explained and showed in depth in the upcoming topics of this document. Users must install the standalone executable file to access the user interfaces. No further registration is needed. On the hardware side, it is recommended that the user has a screen that is capable of at least 1280×720-pixel, preferably 1920×1080-pixel resolution. The design of the game is made with these constraints in mind.

## 2.4 Hardware Interfaces and Requirements

The game itself is a lightweight desktop application. Therefore, following minimum hardware specifications are required to run. As input devices, all users must have a keyboard and mouse. Displays with touch feature can also be used instead of a mouse as an input device.

OS: Windows 7+	
Processor	2.4 GHz Dual Core Processor
Memory	2 GB RAM
Graphics	512 MB dedicated video RAM, Shader Model 4.0
DirectX	Version 11
Storage	At least 1 GB available space
OS: Mac OS X 10.9+	
Processor	2.4 GHz Dual Core Processor
Memory	2 GB RAM
Graphics	512 MB dedicated video RAM, Shader Model 4.0
Storage	At least 1 GB available space

## 2.5 Software Interfaces

The game will not have a feature that requires a server-side server but to run correctly, as a requirement of Unity game engine, all devices running of Windows operating systems must install DirectX software before playing the game. DirectX is bundle that includes a set of standard commands and functions that Unity may use when compiling the game to control video playbacks, sound effects, peripheral input devices such as keyboards and mouse.

## 2.6 Memory Constraints

The game will be a light-weight application. Therefore, at least 2GB of system memory will be enough for the game to run smoothly. At least 1GB of available storage space is also required for the assets and store other necessary data about the game.

## 2.7 Operations

The game allows user to interact in multiple operations. These operations are:

- Using the skill tree
- Accessing the news
- Accessing the stock exchange
- Do marketing
- Manage products
- Manage employees
- Manage bank account
- Manage the office space
- Develop software

To perform above mentioned actions, users need to give input to the game on various points. These inputs will be processed and handled by the game logic to offer a realistic simulation.

## 2.8 Game Functions

There is only one type of user in the game, therefore above-mentioned actions are also valid on this topic.

### 2.8.1 User

- Using the skill tree
- Accessing the news
- Accessing the stock exchange
- Do marketing
- Manage products
- Manage employees
- Manage bank account
- Manage the office space
- Develop software

## 2.8.2 User characteristics

There is only one type of user in the game. This user does not need to follow a specific characteristic. Therefore, no further classifications about the user classes are necessary.

## 2.9 Constraints

### 2.9.1 Regulatory policies

The game will follow the “Allowing game developers and publishers to follow European regulation” of the European Games Developer Federation published in 2020. Therefore, following data will be collected:

- Data required for tax reporting,
- Access to location data for VAT/digital sales tax reporting

### 2.9.2 Hardware limitation

There are no limitations to the game. Users can have the best hardware available to them and get best performance out of the game.

### 2.9.3 Language

The game will be designed and created in English but there will be a possibility in the future to add new languages to it. This will be achieved by pushing update releases to users.

### 2.9.4 Reliability

The game shall work correctly on any supported device and operating system. Any crash or failure within the game will instantly be sent over to developers to be fixed. Therefore, necessary permissions need to be obtained from the user during the installation.

### 2.9.5 Safety

The game will not collect any data about the user that is considered to be a security issue if leaked.

## 2.10 Assumptions and Dependencies

The game will be created using Unity, therefore some depended on packages will be installed alongside the game to ensure a smooth gameplay. These dependencies will be stated in the software manifest during development. An example is given below.

```
1.      {
2.      "name": "com.test.dependency",
3.      "displayName": "Dependency",
4.      "version": "0.0.1",
5.      "unity": "2018.2",
6.      "description": "Base project for another package to depend
on",
7.      "dependencies": { }
8.      }
```

## 2.11 User documentation

Users will be informed using a user documentation or a user assistance document. User documentation refers to the documentation for the game provided to the end users. This document will be designed to assist end users to use the game. There will be two types of user documentation. Help files will provide low level details about the game that can be accessed from inside the game and online support which will be on a website providing larger amount of information.

## 3. Specific Requirements

### 3.1 External Interface Requirements

The following sections focuses on external interface requirements. These sections contain data about the functionalities of the software system. Users interface is designed to be simple. All scenes and windows will be improved in the main game.

#### 3.1.1 Main Screen



This is the main screen or user interface of the game, so when the user logs into the game they will be presented with this menu. All the action of the game and controlling of the overall game will be done through this menu. The buttons below lead to or presents other sub-menus. For instance, when user clicks the marketing button, they will be presented with menu below.



Name of the Item	Marketing
Description of Purpose	Allows players to open Marketing Window
Source of Input	Mouse



Name of the Item	Employee
Description of Purpose	Allows players to open Employee Window
Source of Input	Mouse



Name of the Item	News
Description of Purpose	Allows players to open News Window
Source of Input	Mouse



Name of the Item	Office
Description of Purpose	Allows players to open Office Window
Source of Input	Mouse



Name of the Item	Products
Description of Purpose	Allows players to open Products Window
Source of Input	Mouse



Name of the Item	Skill Tree
Description of Purpose	Allows players to open Skill Tree Window
Source of Input	Mouse



Name of the Item	Software Development
Description of Purpose	Allows players to open Software Development Window
Source of Input	Mouse



Name of the Item	Stock Exchange
Description of Purpose	Allows players to open Stock Exchange Window
Source of Input	Mouse



Name of the Item	Bank
Description of Purpose	Allows players to open Bank Window
Source of Input	Mouse

### 3.1.2 Marketing Window



This menu as the name suggest is for marketing, where user will make all the marketing decisions. The user is given three options, these options are as seen in the picture TV, Social Media and Billboards. The users can opt for all the three options but will have to select and setup one at a time therefore not simultaneously.



Name of the Item	Close
Description of Purpose	To close Marketing Window
Source of Input	Mouse

TV

Name of the Item	TV Button
Description of Purpose	To open Tv Window
Source of Input	Mouse

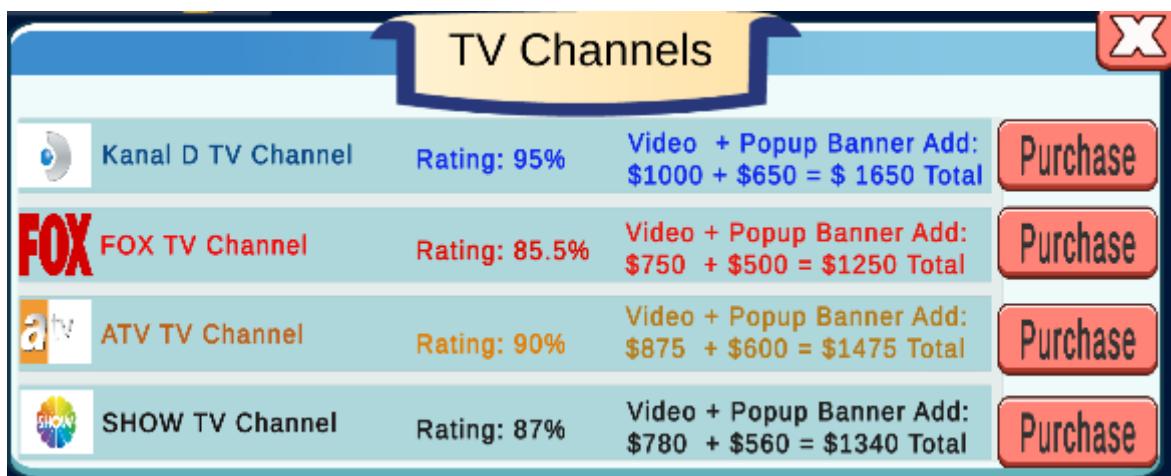
Social Media

Name of the Item	Social Media Button
Description of Purpose	To open Social Media Window
Source of Input	Mouse

Bilboards

Name of the Item	Billboards Button
Description of Purpose	To open Billboards Window
Source of Input	Mouse

### 3.1.3 TV Channels Window



This window is sub menu within TV menu which was one of the modes of marketing in this game. So, when the user clicks the TV button, they will see this menu where they will be able to select the TV Channel where they want to their ads to appear. There are certain criteria's that user can consider when making that decision.

X

Name of the Item	Close
Description of Purpose	To close TV Channels window
Source of Input	Mouse

Purchase

Name of the Item	Purchase Button
Description of Purpose	To Purchase TV Channel Ad package
Source of Input	Mouse

### 3.1.4 Products Window

Rank	Producer	Name	Type	Revenue	Monthly Revenue
#1	BeatDance	Taktak	Mobile App	9.97B	254.26M
#2	TenTen	PubB	Game	4.97B	74.04M
#3	InnerGame	BetweenUs	Game	750.84M	16.29M

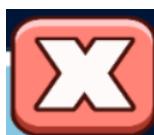
In Product Window, player can achieve product related options. She/he is able to see products' name, the producer company, type of the product, total revenue of the product and monthly revenue of the product. Clicking on Revenue allows the listing to be made according to the products' revenue. Clicking on Monthly Change allows the listing to be made according to the products' monthly revenue.

**Revenue**

Name of the Item	Revenue Button
Description of Purpose	Transmits the system that the list will be made according to the revenues of the products
Source of Input	Mouse

**Monthly Revenue**

Name of the Item	Monthly Revenue Button
Description of Purpose	Transmits the system that the list will be made according to the monthly revenues of the products
Source of Input	Mouse



Name of the Item	Close
Description of Purpose	Closes the news window
Source of Input	Mouse

### 3.1.5 News Window

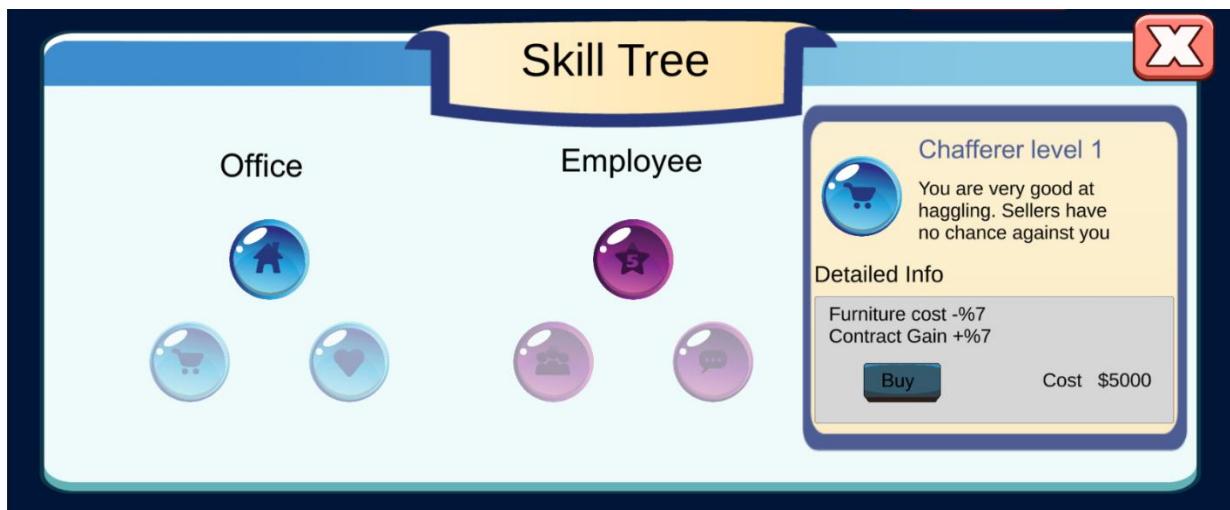


In news window, player will be able to see news from the software industry. Sometimes there will be news about players own company. Sometimes they will be about big IT firms around the world.



Name of the Item	Close
Description of Purpose	Closes the news window
Source of Input	Mouse

### 3.1.6 Skill Tree Window



Skill tree window will allow the player to achieve skills. Player can buy a skill and get bonuses from skills. The Skill tree has programmed and designed to make the player's gaming experience more fun and easier.



Name of the Item	Buy Button
Description of Purpose	Allows the player to buy skills
Source of Input	Mouse



Name of the Item	Office skill 1
Description of Purpose	Allows the player to see office skill 1 details
Source of Input	Mouse



Name of the Item	Office skill 2
Description of Purpose	Allows the player to see office skill 2 details
Source of Input	Mouse



Name of the Item	Office skill 3
Description of Purpose	Allows the player to see office skill 3 details
Source of Input	Mouse



Name of the Item	Employee skill 1
Description of Purpose	Allows the player to see employee skill 1 details
Source of Input	Mouse



Name of the Item	Employee skill 2
Description of Purpose	Allows the player to see employee skill 2 details
Source of Input	Mouse



Name of the Item	Employee skill 3
Description of Purpose	Allows the player to see employee skill 3 details
Source of Input	Mouse

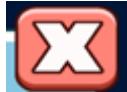
### 3.1.7 Stock Exchange Window

1 Stock Exchange				X
Rank	Company	Brand Value	Change % (Monthly)	
#1	Apple	\$352 billion	+1.4%	
#2	Megasoft	\$327 billion	+1.3%	
#3	Boogle	\$324 billion	+0.5%	
#4	Tencent	\$151 billion	+1.5%	
#5	BSG Company	\$148 billion	+1.4%	Balance-Sheet

In stock exchange window, the player will be able to see his own firms value and values of companies around the world. In this window the player will see firms names, their brand values, their monthly changes and their ranks. The ranking list will contain top 10 company from world and the players place in the world IT industry. Stock exchange window gets update every month.

Balance-Sheet

Name of the Item	Balance Sheet Button
Description of Purpose	To enter the balance sheet window
Source of Input	Mouse



Name of the Item	Close
Description of Purpose	Closes the stock exchange window
Source of Input	Mouse

### 3.1.8 Balance Sheet Window

Balance- Sheet			
Assets		Liabilities	
Cash	1 450 000,00	Purchase of Inventory	40 000,00
Gains from Contracts	100 000,00	Advertising	10 000,00
Gains from Features	0,00	Management and Staff	80 000,00
Inventory	450 000,00	Training	30 000,00
Total Current Assets	2 000 000,00	Taxes	21 000,00
Total Assets	1.819.000	Total Liabilities	181 000,00

Balance sheet window shows players to balance sheet of their company. Balance sheet window contains assets, liabilities and total assets. Player can see their spending and can understand how accurate their investment is. Assets panel contains Cash, Gains from Contracts, Gains from Features, Inventory and Total Current Assets. Liabilities panel contains Purchase of Inventory, Advertising, Management and Staff, Training, Taxes and Total Liabilities. Balance sheet gets reset every month.

Name of the Item	Close
Description of Purpose	Closes the balance sheet window
Source of Input	Mouse

### 3.1.9 Software Development Window



User can earn money by two activities. Firstly, user can create its own product. After initial creation user can enhance that product by performing various actions to improve its quality. Secondly, user can take on contracts from other companies.

In this window, user can create a new product, enhance an already created product or take on a contract from other companies to earn money. These functions can be achieved by clicking the three buttons 'Create New Product', 'Enhance Product' and 'Take on Contracts'. Above shown window is the window that pops open when the user clicks on Develop Software button in the main screen. As default, Crate new product menu is shown. User can switch between these menus by clicking the respecting buttons.

**Create New Product**

Name of the Item	Create New Product
Description of Purpose	Opens the product creation menu
Source of Input	Mouse

**Enhance Product**

Name of the Item	Enhance Product
Description of Purpose	Opens the product enhancement menu
Source of Input	Mouse

**Take on Contracts**

Name of the Item	Take on Contracts
Description of Purpose	Opens the contracts menu
Source of Input	Mouse

**Create Product**

Name of the Item	Create Product
Description of Purpose	Saves the inputs and creates a new product
Source of Input	Mouse

### 3.1.10 Create New Product

Software Name:

Software Type:

Platform:

Required coding skills: 5  
 Required art skills: 4  
 Required design skills: 8  
 Estimated time: 10 days

You can start enhancing the product once it is finished

In this window, user can create a new product. User has to specify the name of the product that they want to create, the type of software and the platform on which this product will run. By choosing these options, required coding, art, design skills and estimated time of completion will be shown.

Software Name:

Name of the Item	Software name input
Description of Purpose	Allows user to enter the name of the software
Source of Input	Mouse and keyboard

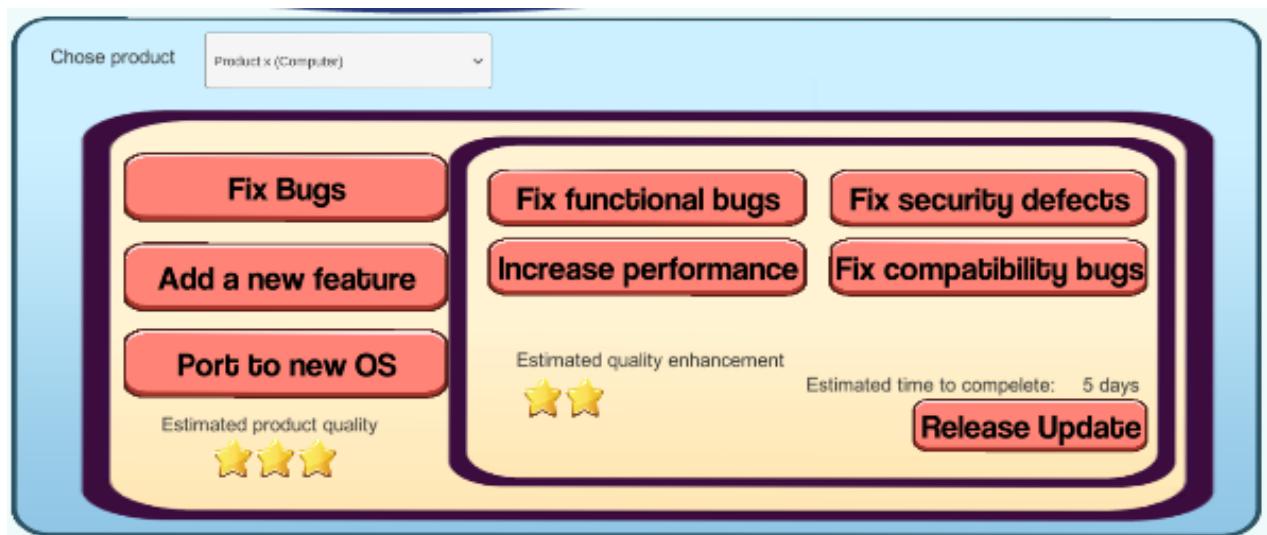
Software Type:

Name of the Item	Software type dropdown input
Description of Purpose	Allows user to select a type of software
Source of Input	Mouse and keyboard

Platform:

Name of the Item	Software platform dropdown input
Description of Purpose	Allows user to select a platform of software
Source of Input	Mouse and keyboard

### 3.1.11 Enhance Software



User can choose to enhance the created product. This can be achieved by three types of enhancement. By fixing bugs, by adding a new feature and by porting the product to a new operating system. Each type of enhancement can be done using clicking buttons on the left side of the Software Enhancement window which pops open when 'Enhance Product' is clicked. User also has to choose the product they wish to be enhanced on the left top of the mentioned window.

Chose product	Product x (Computer)	Name of the Item	Product choosing dropdown input
		Description of Purpose	Allows user to select a product
		Source of Input	Mouse and keyboard

Fix Bugs	Name of the Item	Fix Bugs
	Description of Purpose	Adds bug fixing enhancement
	Source of Input	Mouse

Add a new feature	Name of the Item	Add a new feature
	Description of Purpose	Adds a new feature to the product
	Source of Input	Mouse

Port to new OS	Name of the Item	Fix Bugs
	Description of Purpose	Adds bug fixing enhancement
	Source of Input	Mouse

### 3.1.12 Fix Bugs

User can choose to fix bugs on the created product to enhance its quality. This can be achieved by 4 activities: fixing functional bugs, increase performance, fixing security defects and fixing compatibility bugs all of which can be done by clicking the respective button.

**Fix functional bugs**

Name of the Item	Fix functional bugs
Description of Purpose	Adds fixing of functional bugs enhancement
Source of Input	Mouse

**Increase performance**

Name of the Item	Increase performance
Description of Purpose	Adds performance enhancement
Source of Input	Mouse

**Fix security defects**

Name of the Item	Fix security bugs
Description of Purpose	Adds fixing of security bugs enhancement
Source of Input	Mouse

**Fix compatibility bugs**

Name of the Item	Fix compatibility bugs
Description of Purpose	Adds fixing of compatibility bugs enhancement
Source of Input	Mouse

### 3.1.13 Add a new feature!



User can choose to add a new feature to the chosen product. This can be achieved by adding an online feature, adding a social network feature or allowing users to upload files to the system of the chosen product. These enhancements can and will be changed accordingly to the chosen product.

**Add online feature**

Name of the Item	Add online feature
Description of Purpose	Adds an online feature to the product
Source of Input	Mouse

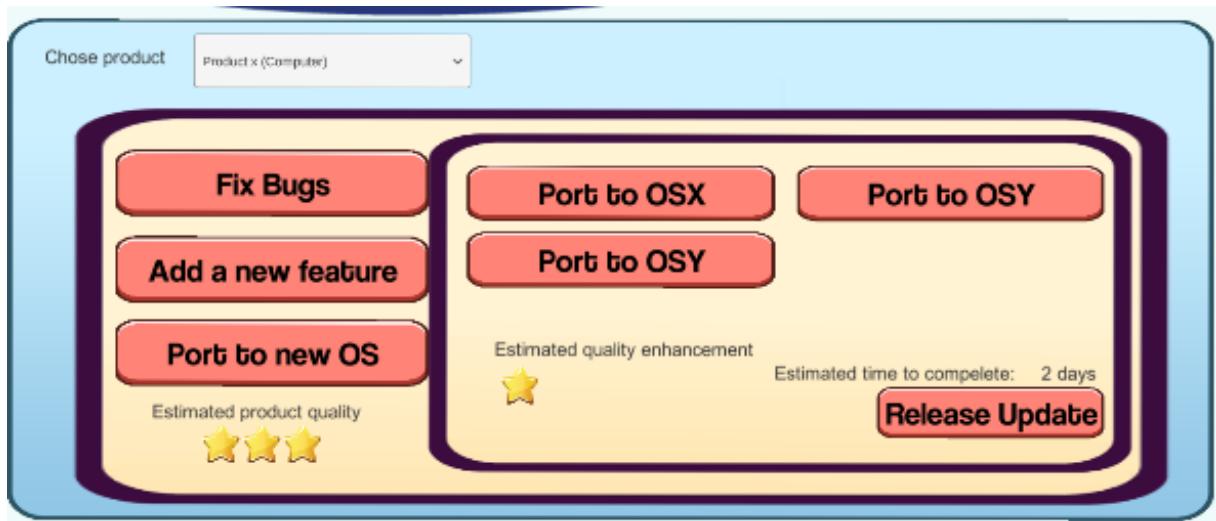
**Social network feature**

Name of the Item	Add social network feature
Description of Purpose	Adds a social networking feature to the product
Source of Input	Mouse

**Allow file upload**

Name of the Item	Allow file upload
Description of Purpose	Allows users to upload files
Source of Input	Mouse

### 3.1.14 Port to new OS



User can choose to port the chosen product to a new operating system to broaden the market of the product. This can be achieved by clicking on the chosen OS type and releasing an update.

Port to OSX

Name of the Item	Port the product to OSX
Description of Purpose	Ports the chosen product to the chosen operating system
Source of Input	Mouse

### 3.1.15 Take on Contracts

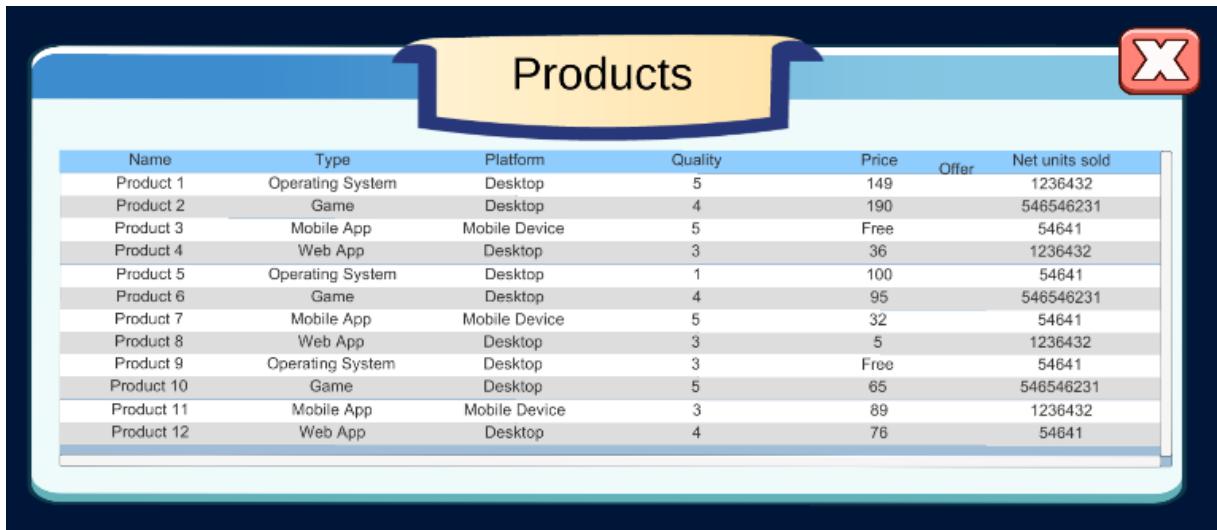
Company	Contract type	Type	Time	Offer	Accept contract
Company X	Operating System	Desktop	5 days	149	Accept contract
Company Y	Game	Desktop	4 days	169	Accept contract
Company Z	Mobile App	Mobile Device	5 days	189	Accept contract
Company X	Game	Desktop	4 days	269	Accept contract
Company X	Operating System	Desktop	5 days	149	Accept contract
Company Y	Game	Desktop	2 days	169	Accept contract
Company Z	Mobile App	Mobile Device	5 days	189	Accept contract
Company X	Game	Desktop	2 days	269	Accept contract
Company X	Operating System	Desktop	5 days	149	Accept contract
Company Y	Game	Mobile Device	4 days	169	Accept contract
Company Z	Mobile App	Mobile Device	5 days	189	Accept contract
Company X	Game	Desktop	2 days	269	Accept contract

User can choose to make money by taking on contracts from other companies. These contracts can range from all the possible types of products the user can create in the game. Companies will post contract ads, stating the contract type, type of device on which the product needs to run, time of completion of the contract and the offer. User can click on the accept button to take on the contract.

Accept contract

Name of the Item	Accept contract
Description of Purpose	Allows user to take on a contract
Source of Input	Mouse

### 3.1.16 Products Window



Name	Type	Platform	Quality	Price	Offer	Net units sold
Product 1	Operating System	Desktop	5	149		1236432
Product 2	Game	Desktop	4	190		546546231
Product 3	Mobile App	Mobile Device	5	Free		54641
Product 4	Web App	Desktop	3	36		1236432
Product 5	Operating System	Desktop	1	100		54641
Product 6	Game	Desktop	4	95		546546231
Product 7	Mobile App	Mobile Device	5	32		54641
Product 8	Web App	Desktop	3	5		1236432
Product 9	Operating System	Desktop	3	Free		54641
Product 10	Game	Desktop	5	65		546546231
Product 11	Mobile App	Mobile Device	3	89		1236432
Product 12	Web App	Desktop	4	76		54641

In the products window, user can see all of the products its company has created over time. This window gives information about the name of the product, type of the device the products runs, the platform on which the product runs, the quality of the product, price of the product and the net units sold until the that day.

### 3.1.17 Escape Window



Escape window let players to stop the game. When players hit escape button in keyboard escape window appears and game stops. In this window players can resume their game, save their game and load their previous saves. Player can also return main menu and quit the game if they want. If players quit game or return main menu, they will lose their unsaved progress.

**Resume**

Name of the Item	Resume
Description of Purpose	Allows players to resume their game
Source of Input	Mouse

**Save Game**

Name of the Item	Save Game
Description of Purpose	Allows players to save their game
Source of Input	Mouse

**Load Game**

Name of the Item	Load Game
Description of Purpose	Allows players to load their previous saves
Source of Input	Mouse

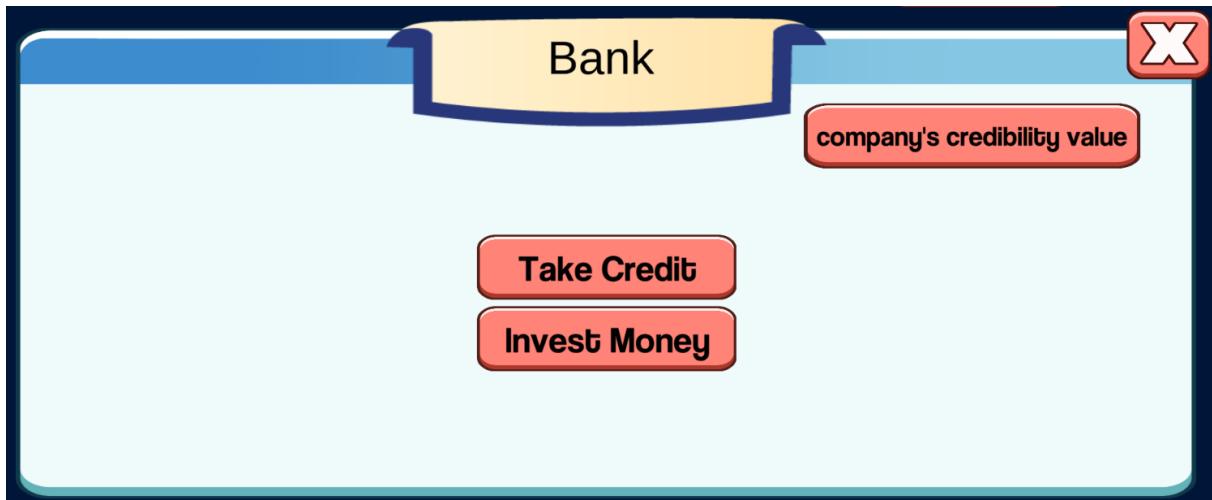
**Return to Main Menu**

Name of the Item	Return to Main Menu
Description of Purpose	Allows players to return to the main menu
Source of Input	Mouse

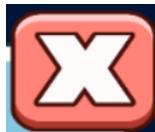
**Quit**

Name of the Item	Quit
Description of Purpose	Allows players to quit the game
Source of Input	Mouse

### 3.1.18 Bank Window



In Bank Window, user/player can achieve bank related options such as Take Credit Button and Invest Money Button. Also, there will be a data with shows the player's company credibility value. This value will affect credit interest. If company's credibility value is high the player could take credit with low interest rate. This value will be created with the values in the company's balance sheet. In addition, the value of customer satisfaction which is created via feedbacks from the company's works may also affect this value.



Name of the Item	Close
Description of Purpose	Closes the news window
Source of Input	Mouse

**Take Credit**

Name of the Item	Take Credit Button
Description of Purpose	Opens Take Credit Window
Source of Input	Mouse

**Invest Money**

Name of the Item	Invest Money Button
Description of Purpose	Opens Invest Money Window
Source of Input	Mouse

### 3.1.19 Take Credit Window



In Take Credit Window, player can achieve take credit related options. Firstly, the player will enter the desired amount of money. Clicking on Approve-Amount Button tells the system that the player is willing to take these amounts of money from bank. Secondly, player will enter the desired time period (shown with months) which is the player's loan repayment period. After entering the time period, player clicks Approve-Period Button. Clicking on this button tells the system the player is determined the time period. Blue area which is down left shows total amount to be paid and the monthly amount to be paid. If player accepts these conditions, he/she clicks on Take-Credit Button and these action causes the followings: System adds player's desired amount of money to the current balance. System starts the repayment process.

*Enter amount...*

Name of the Item	Amount Input Field
Description of Purpose	Allows the user to enter the desired amount of money
Source of Input	Mouse and Keyboard

*Enter month...*

Name of the Item	Time Period Input Field
Description of Purpose	Allows the user to enter the desired period of time
Source of Input	Mouse and Keyboard

**Approve Amount**

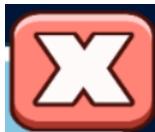
Name of the Item	Approve Amount Button
Description of Purpose	Transmits the system the player is approved the amount
Source of Input	Mouse

## Approve Period

Name of the Item	Approve Period Button
Description of Purpose	Transmits the system that the player approved the period
Source of Input	Mouse

## Take Credit

Name of the Item	Approve Take Credit Button
Description of Purpose	Transmit the system that the player took the credit
Source of Input	Mouse



Name of the Item	Close
Description of Purpose	Closes the news window
Source of Input	Mouse

### 3.1.20 Invest Money Window



In Invest Money Window, player can achieve invest money related options. Firstly, the player enters the desired amount of money. Clicking on Approve-Amount Button tells the system that the player is willing to give these amounts of money to bank. Secondly, player enters the

desired time period (shown with months) which shows when money will be withdrawn. After entering the time period, player clicks Approve-Period Button. Clicking on this button tells the system that the player is determined the time period. Blue area which is down left shows total amount to be obtained after the period. If player accepts these conditions, he/she clicks on Invest-Money Button and these action causes the followings: System subtracts player's desired amount of money from current balance. System starts the invest money process.

*Enter amount...*

Name of the Item	Amount Input Field
Description of Purpose	Allows the user to enter the desired amount of money
Source of Input	Mouse and Keyboard

*Enter month...*

Name of the Item	Time Period Input Field
Description of Purpose	Allows the user to enter the desired period of time
Source of Input	Mouse and Keyboard

**Approve Amount**

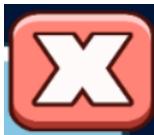
Name of the Item	Approve Amount Button
Description of Purpose	Transmits the system the player is approved the amount
Source of Input	Mouse

**Approve Period**

Name of the Item	Approve Period Button
Description of Purpose	Transmits the system that the player approved the period
Source of Input	Mouse

## Invest Money

Name of the Item	Invest Money Button
Description of Purpose	Transmits the system that the player approved Invest Money process.
Source of Input	Mouse



Name of the Item	Close
Description of Purpose	Closes the news window
Source of Input	Mouse

### 3.1.21 Social Media Platforms Window



Picture above is a sub-menu which is presented when clicked on the social media button. Social media is the second mode of marketing and these are the options shown in the picture are what users can opt for, the cost of each option is given below their respective icons.



Name of the Item	Close
Description of Purpose	To close Social Media window
Source of Input	Mouse

**Buy**

Name of the Item	Buy Button
Description of Purpose	To buy the platform where you want to give your ad
Source of Input	Mouse

### 3.1.22 Billboards Window



The menu above pops up when the user clicks on the Billboard button. Here in this menu user are given three option to pick from as shown in the picture. Option presents the busiest street across the Turkey region, so user can pick the option that best suit the product that they want to advertise and the type of audience they want to reach.



Name of the Item	Close
Description of Purpose	To close Billboards window
Source of Input	Mouse

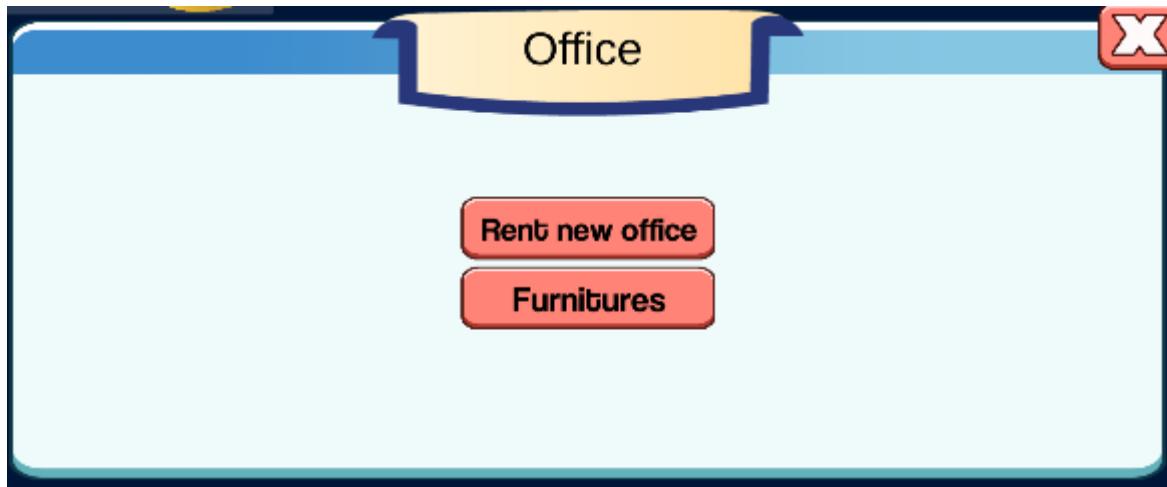


Name of the Item	Office Button
Description of Purpose	To buy billboard to hoist ad
Source of Input	Mouse



Name of the Item	Office Button
Description of Purpose	To open the Office window
Source of Input	Mouse

### 3.1.23 Office Window



When the button just above this picture is clicked this window above pops up. This menu allows user to perform office related function which are to either rent new office or buying new furniture.



Name of the Item	Close
Description of Purpose	To close Office window
Source of Input	Mouse



Name of the Item	Rent new Office Button
Description of Purpose	To open the Rent new office window
Source of Input	Mouse



Name of the Item	Furnitures Button
Description of Purpose	To open Furnitures window
Source of Input	Mouse

### 3.1.24 Furnitures Window



When the furniture button is clicked the menu with various furniture is presented as shown in the picture above. Now here user can buy new furniture which in turn affect the user's productivity, some of the option are visible in the picture and beside these options there will be many more. Users can either buy one furniture or they can purchase it in bulk depending on their needs. The green arrow shows by how much the productivity of the employee will increase.



Name of the Item	Close
Description of Purpose	To close Furniture window
Source of Input	Mouse



Name of the Item	Buy Button
Description of Purpose	To buy furniture
Source of Input	Mouse

### 3.1.25 Rent a New Office Windows



When user selects the rent new office, they are given three region options to pick from and each of these three options pertain a sub-menu within themselves. Through renting new office user can enhance their productivity beside attaining new capabilities and unlocking new achievements there by increasing their valuation ultimately.



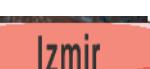
Name of the Item	Close
Description of Purpose	To close Offices window
Source of Input	Mouse

Ankara

Name of the Item	Ankara Button
Description of Purpose	To open Ankara Offices window
Source of Input	Mouse



Name of the Item	Istanbul Button
Description of Purpose	To open Istanbul Offices window
Source of Input	Mouse



Name of the Item	Izmir Button
Description of Purpose	To open Izmir Offices window
Source of Input	Mouse

**Offices**

**Ankara**

**Small Office**

Rent: \$20,000 mthly  
Size: Small  
Capacity: 30-50  
Overall Rating: 7.5

**Rent**

**Medium Office**

Rent: \$50,000 mthly  
Size: Medium  
Capacity: 50-100  
Overall Rating: 8.7

**Rent**

**Large Office**

Rent: \$100,000 mthly  
Size: Large  
Capacity: 100-700  
Overall Rating: 9.3

**Rent**

The above picture is sub-menu within Ankara button, so the menu presents three type of office option within Ankara region. The option varies in terms of their size, rent, capacity and overall rating, so, therefore user can take that into account when making the decision about renting the new office.

**X**

Name of the Item	Close
Description of Purpose	To close Offices window
Source of Input	Mouse

**Rent**

Name of the Item	Rent Button
Description of Purpose	To Rent Office
Source of Input	Mouse

**Offices**

**Istanbul**

**Small Office**

Rent: \$35,000 mthly  
Size: Small  
Capacity: 30-50  
Overall Rating: 7.3

**Rent**

**Medium Office**

Rent: \$75,000 mthly  
Size: Medium  
Capacity: 50-100  
Overall Rating: 8.8

**Rent**

**Large Office**

Rent: \$125,000 mthly  
Size: Large  
Capacity: 100-700  
Overall Rating: 9.5

**Rent**

The above picture is sub-menu within Istanbul button, so the menu presents three type of office option within Istanbul region. The option varies in terms of their size, rent, capacity and overall rating, so, therefore user can take that into account when making the decision about renting the new office.



Name of the Item	Close
Description of Purpose	To close Offices window
Source of Input	Mouse

Rent

Name of the Item	Rent Button
Description of Purpose	To Rent Office
Source of Input	Mouse



The above picture is sub-menu within Izmir button, so the menu presents three type of office option within Izmir region. The option varies in terms of their size, rent, capacity and overall rating, so, therefore user can take that into account when making the decision about renting the new office.

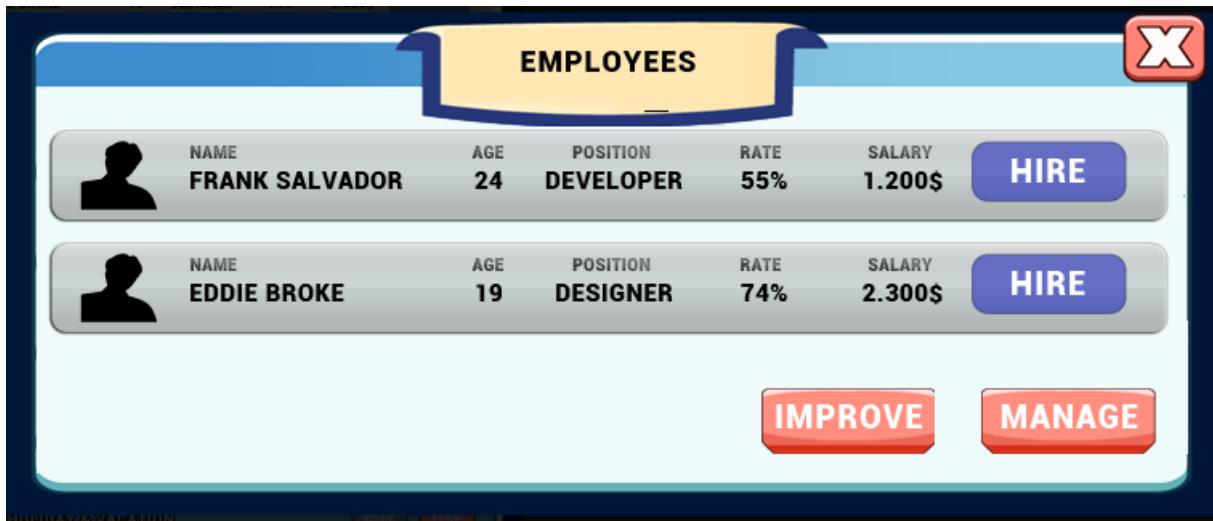


Name of the Item	Close
Description of Purpose	To close Offices window
Source of Input	Mouse

Rent

Name of the Item	Rent Button
Description of Purpose	To Rent Office
Source of Input	Mouse

### 3.1.26 Employees Window



In Employees Window, player will be able to hire new employees for the work-office. The player can also view all employees looking for jobs in the software industry by age, position, rate and salary. Player can sort employees by age, rate, position and salary. In addition, player will be able to access employee improve and manage window from this scene.



Name of the Item	Close
Description of Purpose	Closes the news window
Source of Input	Mouse



Name of the Item	Hire button
Description of Purpose	Hire to new employee
Source of Input	Mouse

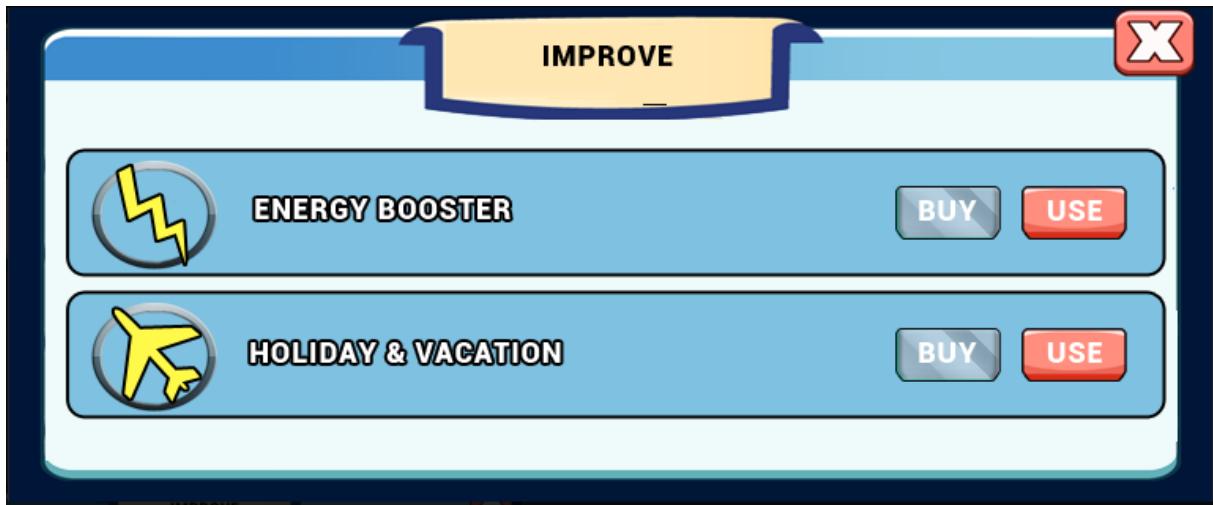


Name of the Item	IMPROVE button
Description of Purpose	Opens Take Improve Window
Source of Input	Mouse



Name of the Item	Manage button
Description of Purpose	Opens Take Manage Window
Source of Input	Mouse

### 3.1.27 Improve Window



In Improve Window, the player can buy or use items and booster for own employees. The items and booster are to temporarily or permanently increase employee's rate. If the player wants to increase own employee's rate, the player buys the items or booster and then clicks the use button to goes the manage window so that player can use those items or booster for own employees.



Name of the Item	Energy Booster
Description of Purpose	Increase the rate of the employee
Source of Input	Mouse



Name of the Item	Holiday & Vacation
Description of Purpose	Increase the rate of the employee
Source of Input	Mouse

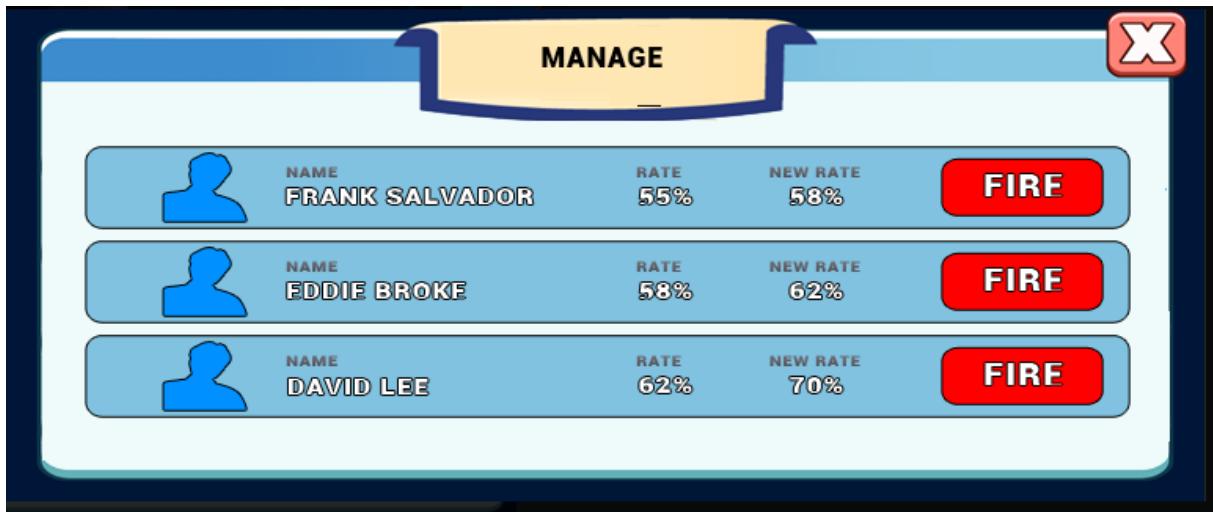


Name of the Item	Buy Button
Description of Purpose	Allows the player to buy items or booster
Source of Input	Mouse

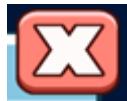


Name of the Item	Use Button
Description of Purpose	Allows the player to use items or booster
Source of Input	Mouse

### 3.1.28 Manage Window



In Manage Window, the player fire to own employees. The fired employees will not appear in this window. The player or user will be able to hire new employees. When the player hires a new employee appears on this screen.



Name of the Item	Close
Description of Purpose	Closes the news window
Source of Input	Mouse



Name of the Item	Fire Button
Description of Purpose	Allows Players to fire their certain employee
Source of Input	Mouse

### 3.1.29 Hardware Interfaces

Hardware interfaces defined in second part of the SRS document. There are no other specific hardware interfaces are required.

### 3.1.30 Software Interfaces

Software interfaces defined in second part of the SRS document. There are no other specific software interfaces are required.

## 3.2 Design Constraints

In BSG Software, we plan to design according to the practice recommended for IEEE Standard 1016-1998 Software Design Explanations.

### 3.3 Software system attributes

There are a number of attributes of software that can serve as requirements. It is important that required attributes be specified so that their achievement can be objectively verified. The following items provide a partial list of examples. These are also known as non-functional requirements or quality attributes.

These are characteristics the system must possess, but that pervade or crosscut the design. These requirements have to be testable just like the functional requirements.

#### 3.3.1 Reliability

Player or user accounts will be held in the database, so if an error occurs player will not affect from this action. Player accounts keep information about their previous and current games information. For this reason, it is important to keep the data in database.

#### 3.3.2 Availability

Offline game has little to no server failure risk because of data will be saved automatically as you progress through the section. To avoid this situation, important measures will be taken, such as establishing checkpoints and some regular backups. So, the application will always have a stable state that we present to the users.

#### 3.3.3 Security

Business Simulation Game (BSG)'s security will be ensured to all players. No personal information will be shared with or sold to any other third-party companies.

#### 3.3.4 Maintainability

All code will be completely archived. Each function will be remarked with preconditions and postconditions. All program records will incorporate remarks concerning initiation and date of last change. The code should be modular, to allow future modifications. Foreseen refreshes incorporate changes to the arrangements of items and their depictions utilized during the game. This ought to be put away in a different information document, as opposed to inserted in the program code.

#### 3.3.5 Portability

It is the ability to compile a game developed with Unity in accordance with different platforms (PC, Mac, Web, iOS, Android, Windows Phone, PlayStation, Xbox etc.) without the need for

any infrastructure changes. In this way, a game prepared for PC can be made running for Mac with one click. Business Simulation Game (BSG), it will be accessible from all devices with a window operating system.

### 3.4 Performance Requirements

- Maximum startup time should not exceed 15 seconds.
- All windows should open within 0.05 seconds.
- 95% of overall response time of any task should not exceed 1.5 seconds.
- 90% of the game should be bug-free on release date.
- The finished product should be ready to be ported into other platforms with the above given performance requirements.
- All bugs should be reported and be notified to the developers.
- Any debugging activity should not exceed a week to complete.
- All bugs should be reported as an issue on the software's GitHub page.

ANKARA YILDIRIM BEYAZIT UNIVERSITY  
BUSINESS SCHOOL  
DEPARTMENT OF MANAGEMENT INFORMATION  
SYSTEMS

**Business Simulation Game**

**Software Design Description**

*Acquirer:*

*Asst. Prof. İhsan Tolga Medeni*

*Research Associate Dilan Özcan*

*Prepared by*

*Mustafa Balta*

*Enes Eyüboğlu*

*Berke Can Yaman*

*İnanç Haluk Aksoy*

*Ahmed Qureshi*

## Table of Contents

1. Version History .....	4
2. Introduction.....	5
Purpose .....	5
Scope.....	5
Definitions, Acronyms and Abbreviations.....	5
3. References.....	6
4. Use Cases.....	6
1. Main Screen Use Case .....	6
View Office Use Case .....	7
Furniture Use Case.....	8
Office Use Case .....	9
2. View Marketing Use Case.....	10
TV Use Case.....	11
Social Media Use Case .....	12
Billboards Use Case.....	13
3. View News Use Case .....	14
4. View Balance Sheet Use Case.....	15
5. View Stock Exchange Use Case .....	16
Balance Sheet Use Case .....	17
6. Press Esc Button Use Cases .....	18
Resume Use Case .....	19
Save Game Use Case .....	20
Load Game Use Case.....	21
Return to Main Menu Use Case .....	22
Quit Use Case .....	23
7. View Skill Tree Use Case .....	24
Office Skill 1 Use Case .....	25
Office Skill 2 Use Case .....	26
Office Skill 3 Use Case .....	27
Employee Skill 1 Use Case.....	28
Employee Skill 2 Use Case.....	29

Employee Skill 3 Use Case.....	30
8. Develop Software Use Case.....	31
Create new Product .....	32
Enhance Product.....	33
Take on Contracts .....	34
9. Products Use Case .....	35
10. View Employees Use Case .....	36
Hire Use Case .....	37
Improve Use Case .....	38
Manage Use Case.....	39
View Improve Use Case.....	40
Buy Use Case .....	41
“Use” Use Case.....	42
View Manage Use Case.....	43
Fire Use Case.....	44
11. View Products Use Case .....	45
Sort by Revenue Use Case.....	46
Sort by Monthly Revenue Use Case.....	47
12. View Bank Use Case .....	48
Take Credit Use Case.....	49
Invest Money Use Case .....	50
5. Data storage .....	51
JSON File.....	51
User Data Object.....	51
Company Data Object .....	52
Companies Data Object .....	53
Products Data Object .....	53
Contracts Data Object.....	54
Accepted Contracts Data Object.....	54

## 1. Version History

<b>Version</b>	<b>Implemented</b>	<b>Revision</b>	<b>Reason</b>
#	By	Date	
1.0	<i>BSG Team</i>	<i>22.01.2021</i>	<i>Initial version</i>

## 2. Introduction

### Purpose

The purpose of this document is to expose the use cases of the BSG project and, how the developers will save the game data. Thus, the contents related to the design are in this document. The purpose of the preparation of this document is to help the developers, instructors, and testers of the project.

### Scope

The fundamental goal of this project is to build a single player business simulation game. The targeted features of the project can be listed as below:

- To entertain the users of the game.
- To have well written document.
- To be able to handle or support large number of players.
- To run as smoothly as possible to maximize the gaming experience.
- To have music and sound effects that appeal to the genre and target users of the game.

As mentioned before or in the previous documents that the ultimate goal of player will be to become the most valuable out there in the stock market. Each related window that is presented to player which are as follows Software Development, Office, Bank, Skill Tree, News, Employee, Products, Marketing, and Stock Exchange are given in form of use case to highlight how player can take respective action in the game and moreover the methodology that is used for data is also present in this document.

### Definitions, Acronyms and Abbreviations

BSG: Business Simulation Game

SDD: Software Design Document

SPMP: Software Project Management Plan

SRS: Software Requirement Specification

Use Case: A use case is a explanation of how a person who actually uses that process or system will achieve an aim.

Use Case Diagram: The purpose of a use case diagram is to show the different pathways that a user might interact with a system.

UI: User Interface

IEEE: The Institute of Electrical and Electronics Engineers, Inc

Player: A person who plays the game.

Unity3D: A game engine which provide to design and develop a game for PC, mobile and console platforms.

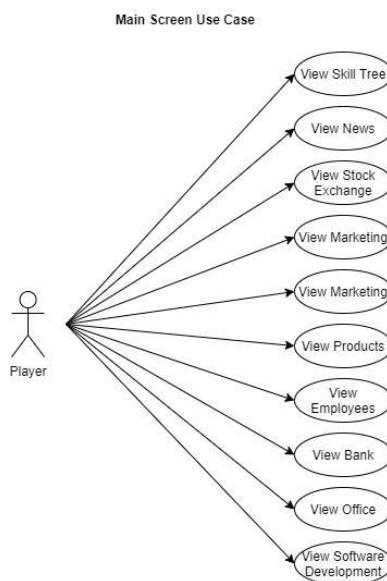
Single-Player: A game that can only be played by one person.

### 3. References

- *IEEE STD 1016-1998, IEEE Recommended Practice for Software Design Descriptions*
- *BSG Software Project Management Plan, 2020*
- *BSG Software Requirement Specification, 2020*

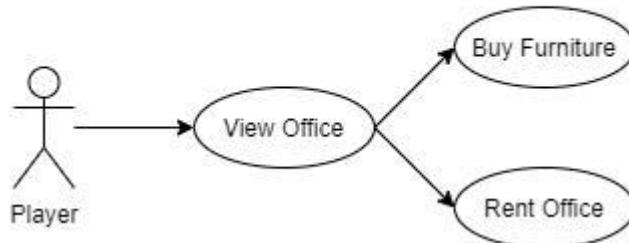
### 4. Use Cases

#### 1. Main Screen Use Case



## View Office Use Case

Office Window Use Case



<b>Use Case Name</b>	<b>View Office</b>
<b>Actor</b>	Player
<b>Description</b>	This use case describes how player can view the office window
<b>Trigger</b>	Player clicks the "Office" button in the Main Window.
<b>Preconditions</b>	Player is in Main Window.
<b>Normal Course</b>	<ul style="list-style-type: none"> <li>-Player clicks the "Office" button in the Main Window.</li> <li>-System activates Office Window and shows the further data or menu about the office</li> </ul>
<b>Alternative Course</b>	-
<b>Postcondition</b>	1. Player has viewed the Office
<b>Exceptions</b>	

2.1

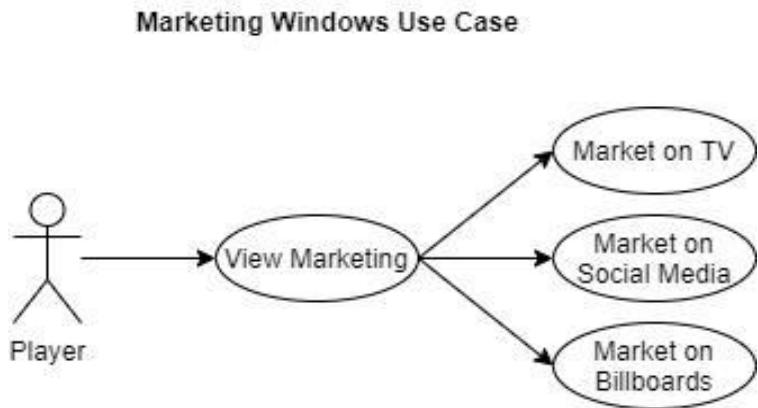
## Furniture Use Case

<b>Use Case Name</b>	<b>Buy Furniture</b>
<b>Actor</b>	Player
<b>Description</b>	This use case describes how player can buy the Furnitures
<b>Trigger</b>	Player clicks the "Furniture" button in the Office Window.
<b>Preconditions</b>	Player is in the Office Window.
<b>Normal Course</b>	<ul style="list-style-type: none"> <li>-Player scrolls the "Furniture" window to view all the available Furnitures</li> <li>-Player selects the Furnitures that they want to buy</li> <li>-click on the buy button beneath each furniture to purchase it</li> <li>-System confirms the purchase</li> <li>-cost of the furniture bought is deducted from the players balance</li> </ul>
<b>Alternative Course</b>	<ul style="list-style-type: none"> <li>-if the balance in the player account is not sufficient then the purchase is declined</li> </ul>
<b>Postcondition</b>	<ul style="list-style-type: none"> <li>-the selected furniture is purchased and can be viewed and placed in the office</li> </ul>
<b>Exceptions</b>	

## Office Use Case

<b>Use Case Name</b>	<b>Rent Office</b>
<b>Actor</b>	Player
<b>Description</b>	This use case describes how player can rent the Office
<b>Trigger</b>	Player clicks the "Office" button in the Office Window.
<b>Preconditions</b>	Player is in the Office Window.
<b>Normal Course</b>	<ul style="list-style-type: none"> <li>-Player selects the “region” from which to purchase office</li> <li>-then selects from the office option available in that region</li> <li>-by clicking the rent button, they can rent the office</li> <li>-system approves the transaction</li> <li>-the rent of the first month is deducted upfront from the balance</li> </ul>
<b>Alternative Course</b>	<ul style="list-style-type: none"> <li>-if the balance in the player account is not sufficient then system displays the message regarding that and does not allow the transaction to proceed</li> </ul>
<b>Postcondition</b>	<ul style="list-style-type: none"> <li>-the rented office can be viewed in the main screen</li> </ul>
<b>Exceptions</b>	

## 2. View Marketing Use Case



<b>Use Case Name</b>	<b>View Marketing</b>
<b>Actor</b>	Player
<b>Description</b>	This use case describes how player can view the Marketing window
<b>Trigger</b>	Player clicks the "Marketing" button in the Main Window.
<b>Preconditions</b>	Player is in Main Window.
<b>Normal Course</b>	<ul style="list-style-type: none"> <li>-Player clicks the "Marketing" button in the Main Window.</li> <li>-System activates Marketing Window and shows the further data or menu about the marketing</li> </ul>
<b>Alternative Course</b>	-
<b>Postcondition</b>	1. Player has viewed the Marketing
<b>Exceptions</b>	

## TV Use Case

<b>Use Case Name</b>	<b>Market on TV</b>
<b>Actor</b>	Player
<b>Description</b>	This use case describes how player can advertise or market on TV
<b>Trigger</b>	Player clicks the "TV" button in the Marketing Window.
<b>Preconditions</b>	Player is in the Marketing Window.
<b>Normal Course</b>	<ul style="list-style-type: none"> <li>-player selects one or multiple tv channel on which they want their ad to appear</li> <li>-click on the buy button proceed</li> <li>-system check the total cost against balance and approves</li> <li>-total cost is deducted from the player's balance</li> </ul>
<b>Alternative Course</b>	<ul style="list-style-type: none"> <li>-if the balance in the player account is not sufficient then system displays the message regarding that and does not allow for ads to be bought</li> </ul>
<b>Postcondition</b>	<ul style="list-style-type: none"> <li>-players desired ads appear on the selected TV channels</li> </ul>

## Social Media Use Case

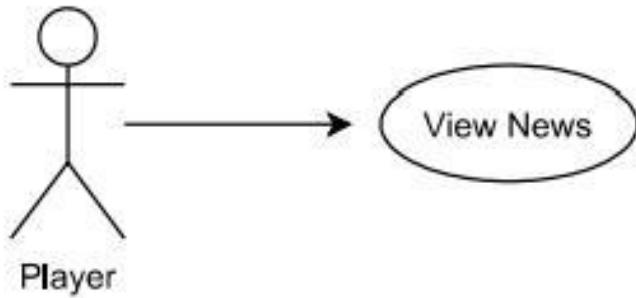
<b>Use Case Name</b>	<b>Market on Social Media</b>
<b>Actor</b>	Player
<b>Description</b>	This use case describes how player can advertise or Market on Social Media
<b>Trigger</b>	Player clicks the "Social Media" button in the Marketing Window.
<b>Preconditions</b>	Player is in the Marketing Window.
<b>Normal Course</b>	<ul style="list-style-type: none"> <li>-player views and selects the social media platform to give ads on</li> <li>-players click on the buy button below the icon to make the purchase</li> <li>-system check the total cost against balance and approves</li> <li>-total cost is deducted from the player's balance</li> </ul>
<b>Alternative Course</b>	<ul style="list-style-type: none"> <li>-if the balance in the player account is not sufficient then system displays the message regarding that and does not allow for ads to be bought</li> </ul>
<b>Postcondition</b>	<ul style="list-style-type: none"> <li>-players desired ads appear on the selected Social Media platforms</li> </ul>

## Billboards Use Case

<b>Use Case Name</b>	<b>Market on Billboards</b>
<b>Actor</b>	Player
<b>Description</b>	This use case describes how player can advertise or Market on Billboards
<b>Trigger</b>	Player clicks the " Billboards " button in the Marketing Window.
<b>Preconditions</b>	Player is in the Marketing Window.
<b>Normal Course</b>	<ul style="list-style-type: none"> <li>-player views and selects the billboards on which they want their ads to appear</li> <li>-players click on the buy button to make the purchase</li> <li>-system check the total cost against balance and approves</li> <li>-total cost is deducted from the player's balance</li> </ul>
<b>Alternative Course</b>	<ul style="list-style-type: none"> <li>-if the balance in the player account is not sufficient then system displays the message regarding that and does not allow for ads to be bought</li> </ul>
<b>Postcondition</b>	<ul style="list-style-type: none"> <li>-players desired ads appear on the selected Billboards</li> </ul>

### 3. View News Use Case

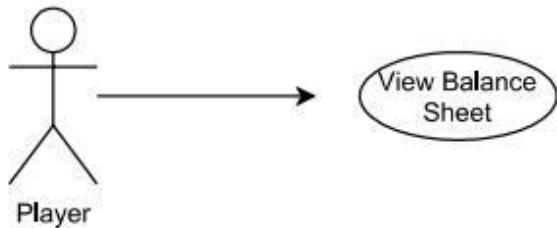
#### News Window Use Case



<b>Use Case Name</b>	<b>View News</b>
<b>Actor</b>	Player
<b>Description</b>	This use case describes how the player can view News Window.
<b>Trigger</b>	The player clicks the "News" Button.
<b>Preconditions</b>	The player is in the Main Screen.
<b>Normal Course</b>	<ul style="list-style-type: none"><li>- The player clicks the "News" Button.</li><li>- The system opens News Window for the player.</li></ul>
<b>Alternative Course</b>	-
<b>Postcondition</b>	-
<b>Exceptions</b>	

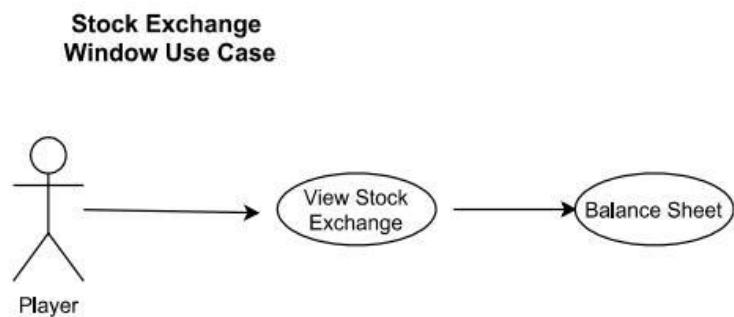
#### 4. View Balance Sheet Use Case

##### Balance Sheet Window Use Case



Use Case Name	View Balance Sheet
Actor	Player
Description	This use case describes how player can view Balance Sheet Window.
Trigger	The player clicks the "Balance Sheet" Button.
Preconditions	The player is in the Stock Exchange Window.
Normal Course	<ul style="list-style-type: none"><li>- The player clicks the "Balance Sheet" Button.</li><li>- The system opens Balance Sheet Window for the player.</li></ul>
Alternative Course	-
Postcondition	-
Exceptions	

## 5. View Stock Exchange Use Case

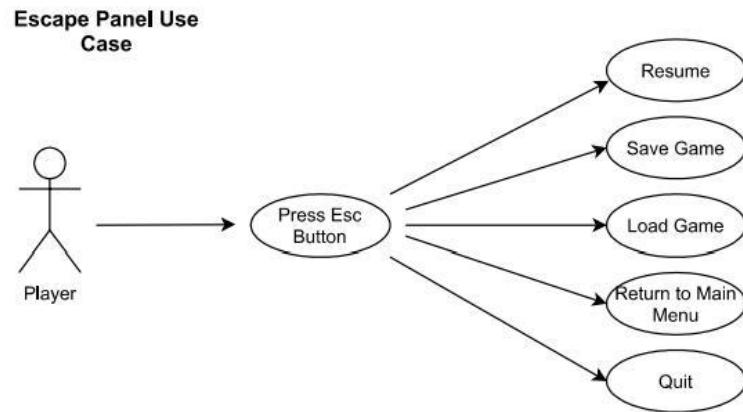


<b>Use Case Name</b>	<b>View Stock Exchange</b>
<b>Actor</b>	Player
<b>Description</b>	This use case describes how the player can view Stock Exchange Window.
<b>Trigger</b>	The player clicks the "Stock Exchange" Button.
<b>Preconditions</b>	The player is in the Main Screen.
<b>Normal Course</b>	<ul style="list-style-type: none"> <li>- The player clicks the "Stock Exchange" Button.</li> <li>- The system opens Stock Exchange Window for the player.</li> </ul>
<b>Alternative Course</b>	-
<b>Postcondition</b>	-
<b>Exceptions</b>	

## Balance Sheet Use Case

<b>Use Case Name</b>	<b>Balance Sheet</b>
<b>Actor</b>	Player
<b>Description</b>	This use case describes what happens when the player clicks "Balance Sheet" Button.
<b>Trigger</b>	The player clicks the "Balance Sheet" Button.
<b>Preconditions</b>	The player is in the Stock Exchange Window.
<b>Normal Course</b>	<ul style="list-style-type: none"><li>- The player clicks the "Stock Exchange" Button.</li><li>- The system opens Balance Sheet Window for the player.</li></ul>
<b>Alternative Course</b>	-
<b>Postcondition</b>	-
<b>Exceptions</b>	

## 6. Press Esc Button Use Cases



<b>Use Case Name</b>	<b>View Escape Panel</b>
<b>Actor</b>	Player
<b>Description</b>	This use case describes how the player can view Escape Panel.
<b>Trigger</b>	The player presses the "Escape" Button.
<b>Preconditions</b>	
<b>Normal Course</b>	<ul style="list-style-type: none"> <li>- The player clicks the "Escape" Button.</li> <li>- The system opens Escape Panel for the player.</li> </ul>
<b>Alternative Course</b>	-
<b>Postcondition</b>	<ul style="list-style-type: none"> <li>- The Game pauses.</li> </ul>
<b>Exceptions</b>	

## Resume Use Case

<b>Use Case Name</b>	<b>Resume</b>
<b>Actor</b>	Player
<b>Description</b>	This use case describes what happens when player clicks "Resume" Button in Escape Panel.
<b>Trigger</b>	The player clicks the "Resume" Button.
<b>Preconditions</b>	- The player is in the Escape Panel.
<b>Normal Course</b>	- The player clicks the "Resume" Button. - The system continues the game.
<b>Alternative Course</b>	-
<b>Postcondition</b>	-
<b>Exceptions</b>	

## Save Game Use Case

<b>Use Case Name</b>	<b>Save Game</b>
<b>Actor</b>	Player
<b>Description</b>	This use case describes what happens when player clicks "Save Game" Button in Escape Panel.
<b>Trigger</b>	The player clicks the "Save Game" Button.
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>- The player is in the Escape Panel.</li> </ul>
<b>Normal Course</b>	<ul style="list-style-type: none"> <li>- The player clicks the "Save Game" Button.</li> <li>- The system saves the progress of the player.</li> </ul>
<b>Alternative Course</b>	<ul style="list-style-type: none"> <li>-</li> </ul>
<b>Postcondition</b>	<ul style="list-style-type: none"> <li>- The player can load the certain game save now.</li> </ul>
<b>Exceptions</b>	<ul style="list-style-type: none"> <li>- The game can not be saved in the Main Menu</li> </ul>

## Load Game Use Case

<b>Use Case Name</b>	<b>Load Game</b>
<b>Actor</b>	Player
<b>Description</b>	This use case describes what happens when player clicks "Load Game" Button in Escape Panel.
<b>Trigger</b>	The player clicks the "Load Game" Button.
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>- The player is in the Escape Panel.</li> </ul>
<b>Normal Course</b>	<ul style="list-style-type: none"> <li>- The player clicks the "Load Game" Button.</li> <li>- The system loads the previous save of the game.</li> </ul>
<b>Alternative Course</b>	<ul style="list-style-type: none"> <li>-</li> </ul>
<b>Postcondition</b>	<ul style="list-style-type: none"> <li>-</li> </ul>
<b>Exceptions</b>	

## Return to Main Menu Use Case

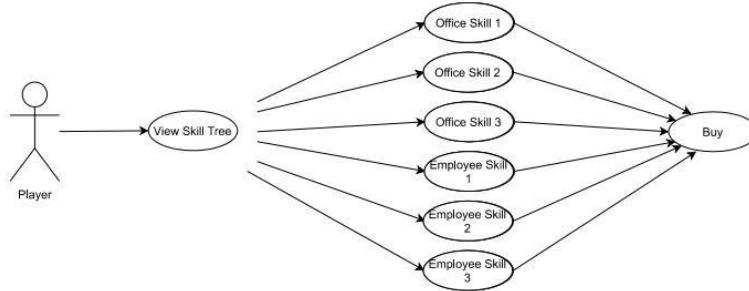
Use Case Name	Return to Main Menu
Actor	Player
Description	This use case describes what happens when player clicks "Return to Main Menu" Button in Escape Panel.
Trigger	The player clicks the "Return to Main Menu" Button.
Preconditions	- The player is in the Escape Panel.
Normal Course	- The player clicks the "Return to Main Menu" Button. - The system returns to the Main Menu. The player loses unsaved progress.
Alternative Course	-
Postcondition	-
Exceptions	

## Quit Use Case

<b>Use Case Name</b>	<b>Quit</b>
<b>Actor</b>	Player
<b>Description</b>	This use case describes what happens when player clicks “Quit” Button in Escape Panel.
<b>Trigger</b>	The player clicks the "Quit" Button.
<b>Preconditions</b>	- The player is in the Escape Panel.
<b>Normal Course</b>	- The player clicks the "Quit" Button. - The system goes off. The player loses unsaved progress.
<b>Alternative Course</b>	- The player may also quit game from Main Menu too.
<b>Postcondition</b>	-
<b>Exceptions</b>	

## 7. View Skill Tree Use Case

Skill Tree Window  
Use Case



<b>Use Case Name</b>	<b>View Skill Tree</b>
<b>Actor</b>	Player
<b>Description</b>	This use case describes how the player can view Skill Tree.
<b>Trigger</b>	The player clicks the "Skill Tree" Button.
<b>Preconditions</b>	- The player is in the Main Screen.
<b>Normal Course</b>	- The player clicks the "Skill Tree" Button. - The system opens Skill Tree Window for the player.
<b>Alternative Course</b>	-
<b>Postcondition</b>	-
<b>Exceptions</b>	

## Office Skill 1 Use Case

<b>Use Case Name</b>	<b>Office Skill 1</b>
<b>Actor</b>	Player
<b>Description</b>	This use case describes what happens when player clicks "Office Skill 1" Button in Skill Tree Window.
<b>Trigger</b>	The player clicks the "Office Skill 1" Button.
<b>Preconditions</b>	- The player is in the Skill Tree Window.
<b>Normal Course</b>	- The player clicks the "Office Skill 1" Button. - The system opens a window that contains detailed information about certain skill.
<b>Alternative Course</b>	-
<b>Postcondition</b>	- Detailed information window contains the "Buy" Button. Player now can buy Office Skill 1.
<b>Exceptions</b>	

## Office Skill 2 Use Case

<b>Use Case Name</b>	<b>Office Skill 2</b>
<b>Actor</b>	Player
<b>Description</b>	This use case describes what happens when player clicks “Office Skill 2” Button in Skill Tree Window.
<b>Trigger</b>	The player clicks the "Office Skill 2" Button.
<b>Preconditions</b>	- The player is in the Skill Tree Window.
<b>Normal Course</b>	- The player clicks the "Office Skill 2" Button. - The system opens a window that contains detailed information about certain skill.
<b>Alternative Course</b>	-
<b>Postcondition</b>	- Detailed information window contains the “Buy” Button. Player now can buy Office Skill 2.
<b>Exceptions</b>	

## Office Skill 3 Use Case

<b>Use Case Name</b>	<b>Office Skill 3</b>
<b>Actor</b>	Player
<b>Description</b>	This use case describes what happens when player clicks “Office Skill 3” Button in Skill Tree Window.
<b>Trigger</b>	The player clicks the "Office Skill 2" Button.
<b>Preconditions</b>	- The player is in the Skill Tree Window.
<b>Normal Course</b>	- The player clicks the "Office Skill 2" Button. - The system opens a window that contains detailed information about certain skill.
<b>Alternative Course</b>	-
<b>Postcondition</b>	- Detailed information window contains the “Buy” Button. Player now can buy Office Skill 3.
<b>Exceptions</b>	

## Employee Skill 1 Use Case

<b>Use Case Name</b>	<b>Employee Skill 1</b>
<b>Actor</b>	Player
<b>Description</b>	This use case describes what happens when player clicks "Employee Skill 1" Button in Skill Tree Window.
<b>Trigger</b>	The player clicks the "Employee Skill 1" Button.
<b>Preconditions</b>	- The player is in the Skill Tree Window.
<b>Normal Course</b>	- The player clicks the "Employee Skill 1" Button. - The system opens a window that contains detailed information about certain skill.
<b>Alternative Course</b>	-
<b>Postcondition</b>	- Detailed information window contains the "Buy" Button. Player now can buy Employee Skill 1.
<b>Exceptions</b>	

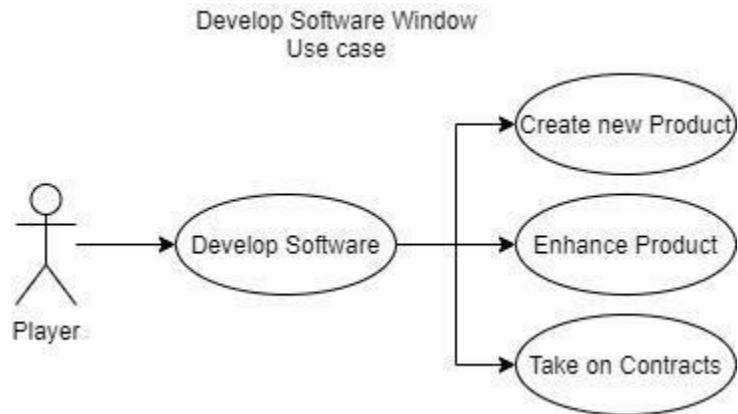
## Employee Skill 2 Use Case

<b>Use Case Name</b>	<b>Employee Skill 2</b>
<b>Actor</b>	Player
<b>Description</b>	This use case describes what happens when player clicks "Employee Skill 2" Button in Skill Tree Window.
<b>Trigger</b>	The player clicks the "Employee Skill 2" Button.
<b>Preconditions</b>	- The player is in the Skill Tree Window.
<b>Normal Course</b>	- The player clicks the "Employee Skill 2" Button. - The system opens a window that contains detailed information about certain skill.
<b>Alternative Course</b>	-
<b>Postcondition</b>	- Detailed information window contains the "Buy" Button. Player now can buy Employee Skill 2.
<b>Exceptions</b>	

## Employee Skill 3 Use Case

<b>Use Case Name</b>	<b>Employee Skill 3</b>
<b>Actor</b>	Player
<b>Description</b>	This use case describes what happens when player clicks "Employee Skill 3" Button in Skill Tree Window.
<b>Trigger</b>	The player clicks the "Employee Skill 3" Button.
<b>Preconditions</b>	- The player is in the Skill Tree Window.
<b>Normal Course</b>	- The player clicks the "Employee Skill 3" Button. - The system opens a window that contains detailed information about certain skill.
<b>Alternative Course</b>	-
<b>Postcondition</b>	- Detailed information window contains the "Buy" Button. Player now can buy Employee Skill 3.
<b>Exceptions</b>	

## 8. Develop Software Use Case



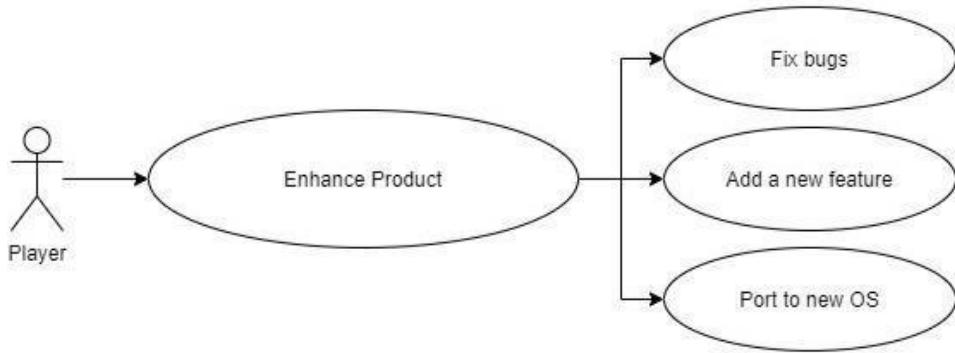
<b>Use Case Name</b>	<b>Develop Software</b>
<b>Actor</b>	Player
<b>Description</b>	This use case describes how the user interacts with the game if the user clicks on the “Develop Software” button.
<b>Trigger</b>	Player clicks the " Develop Software " button in the Main Window.
<b>Preconditions</b>	Player is in Main Window.
<b>Normal Course</b>	-Player clicks the " Develop Software " button in the Main Window. -System activates Develop Software Window and shows the options player can chose from.
<b>Alternative Course</b>	Player exits the window by clicking the X on the top right.
<b>Postcondition</b>	1. Player viewed options about software development”.
<b>Exceptions</b>	

## Create new Product



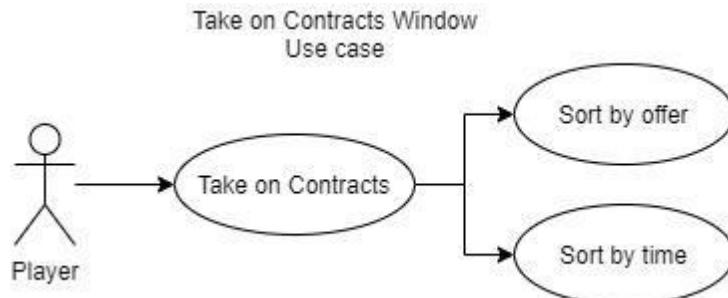
<b>Use Case Name</b>	<b>Create new Product</b>
<b>Actor</b>	Player
<b>Description</b>	This use case describes how the user interacts with the game if the user clicks on the “Create new Product” button on the Develop Software window.
<b>Trigger</b>	Player clicks the " Create new Product" button in the Software Development Window.
<b>Preconditions</b>	Player is in the Software Development Window.
<b>Normal Course</b>	<ul style="list-style-type: none"> <li>-Player clicks the " Create new Product " button in the Develop Software window.</li> <li>-System activates the “Create new Product Window” and shows the options player can chose from.</li> </ul>
<b>Alternative Course</b>	Player exits the window by clicking the X on the top right.
<b>Postcondition</b>	1. Player viewed options about developing a new product.
<b>Exceptions</b>	

## Enhance Product



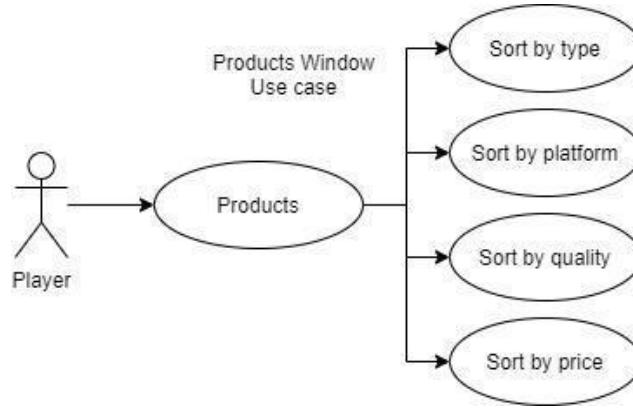
<b>Use Case Name</b>	Enhance Product
<b>Actor</b>	Player
<b>Description</b>	This use case describes how the user interacts with the game if the user clicks on the "Enhance Product" button on the Develop Software window.
<b>Trigger</b>	Player clicks the "Enhance Product" button in the Software Development Window.
<b>Preconditions</b>	Player is in the Software Development Window.
<b>Normal Course</b>	<ul style="list-style-type: none"> <li>-Player clicks the "Enhance Product" button in the Develop Software window.</li> <li>-System activates the "Enhance Product Window" and shows the options player can chose from.</li> </ul>
<b>Alternative Course</b>	Player exits the window by clicking the X on the top right.
<b>Postcondition</b>	<ol style="list-style-type: none"> <li>1. Player viewed options about developing a new product:           <ol style="list-style-type: none"> <li>a. Player viewed "Fix functional bugs" option</li> <li>b. Player viewed "Fix security defects" option</li> <li>c. Player viewed "Increase performance" option</li> <li>d. Player viewed "Fix compatibility bugs" option</li> </ol> </li> </ol>
<b>Exceptions</b>	-

## Take on Contracts



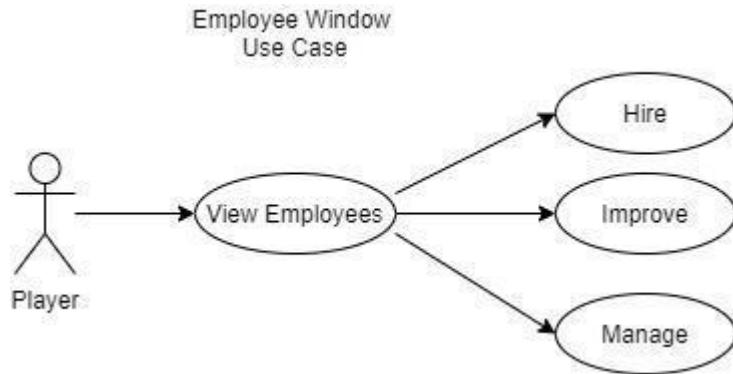
<b>Use Case Name</b>	<b>Take on Contracts</b>
<b>Actor</b>	Player
<b>Description</b>	This use case describes how the user interacts with the game if the user clicks on the “Take on Contracts” button on the Develop Software window.
<b>Trigger</b>	Player clicks the "Take on Contracts" button in the Software Development Window.
<b>Preconditions</b>	Player is in the Software Development Window.
<b>Normal Course</b>	<ul style="list-style-type: none"> <li>-Player clicks the "Take on Contracts " button in the Develop Software window.</li> <li>-System activates “the Take on Contracts Window” and shows the options player can chose from.</li> </ul>
<b>Alternative Course</b>	Player exits the window by clicking the X on the top right.
<b>Postcondition</b>	<ol style="list-style-type: none"> <li>1. Player viewed current offers of contracts by other companies.</li> <li>2. Player can sort by           <ol style="list-style-type: none"> <li>a. Offer</li> <li>b. Time</li> </ol> </li> </ol>
<b>Exceptions</b>	-

## 9. Products Use Case



<b>Use Case Name</b>	<b>Products</b>
<b>Actor</b>	Player
<b>Description</b>	This use case describes how the user interacts with the game if the user clicks on the “Products” button on the Develop Software window.
<b>Trigger</b>	Player clicks the " Products" button in the Main Window.
<b>Preconditions</b>	Player is in the Main Window.
<b>Normal Course</b>	<p>-Player clicks the " Products " button in the main menu.</p> <p>-System activates “the Products Window” and shows the options player can chose from.</p>
<b>Alternative Course</b>	Player exits the window by clicking the X on the top right.
<b>Postcondition</b>	<ol style="list-style-type: none"> <li>1. Player viewed current offers of products created by the company.</li> <li>2. Player can sort by             <ol style="list-style-type: none"> <li>a. Type</li> <li>b. Platform</li> <li>c. Quality</li> <li>d. Price</li> </ol> </li> </ol>
<b>Exceptions</b>	-

## 10. View Employees Use Case



<b>Use Case Name</b>	<b>View Employees</b>
<b>Actor</b>	Player
<b>Description</b>	This use case allows the player to view the employees who will work for the company, in addition the player will be able to develop and manage their own employees.
<b>Trigger</b>	Player clicks the "Employees" button in the Main Window.
<b>Preconditions</b>	Player is in Main Window.
<b>Normal Course</b>	-Player clicks the "Employees" button in the Main Window. -System activates Employees Window and shows the data about employees.
<b>Alternative Course</b>	-
<b>Postcondition</b>	1. Player viewed the employees.
<b>Exceptions</b>	

## Hire Use Case

<b>Use Case Name</b>	Hire
<b>Actor</b>	Player
<b>Description</b>	This use case is where player can hire employees that can be worked for the player company the employees in the Employees Window.
<b>Trigger</b>	Player clicks the "Hire" Button.
<b>Preconditions</b>	Player is in Employees Window.
<b>Normal Course</b>	-Player clicks the "Hire" Button in the Employee Window. -The system displays employees sorted by their monthly wages, from low to high.
<b>Alternative Course</b>	-
<b>Postcondition</b>	1. Player viewed and hired employees sorted by their monthly wages.
<b>Exceptions</b>	

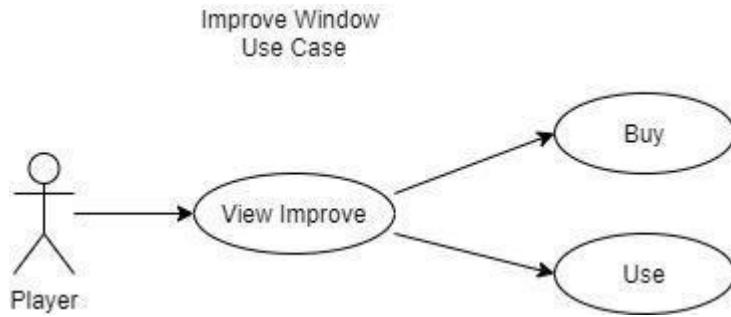
## Improve Use Case

<b>Use Case Name</b>	<b>Improve</b>
<b>Actor</b>	Player
<b>Description</b>	This use case describes player can buy or use items and booster for own employees
<b>Trigger</b>	Player clicks the "Improve" Button.
<b>Preconditions</b>	Player is in Improve Window.
<b>Normal Course</b>	<ul style="list-style-type: none"> <li>-Player clicks the "Improve" Button in the Employees Window.</li> <li>-The system lists the items and boosters used for its employees.</li> <li>-Player can increase the motivation of their employees with these items or boosters.</li> <li>-Player needs to purchase items or boosters before using.</li> <li>-Player purchases these items, they are added to the inventory and it decreases as the player uses them.</li> </ul>
<b>Alternative Course</b>	-
<b>Postcondition</b>	1. Player used and bought the items and boosters.
<b>Exceptions</b>	

## Manage Use Case

<b>Use Case Name</b>	Manage
<b>Actor</b>	Player
<b>Description</b>	In this use case, the player can view and manage their own employees in the Manage Window.
<b>Trigger</b>	Player clicks the "Manage" Button.
<b>Preconditions</b>	Player is in Employees Window.
<b>Normal Course</b>	-Player clicks the "Manage" Button in the Manage Window. - The system allows the player to fire their own employees
<b>Alternative Course</b>	-
<b>Postcondition</b>	1. Player viewed and fired to their own employees.
<b>Exceptions</b>	

## View Improve Use Case



<b>Use Case Name</b>	<b>View Improve</b>
<b>Actor</b>	Player
<b>Description</b>	This use case describes how player views Improve Window.
<b>Trigger</b>	Player clicks the "Improve" button in the Employees Window.
<b>Preconditions</b>	Player is in Employees Window.
<b>Normal Course</b>	-Player clicks the "Improve" button in the Employees Window. -System activates Improve Window and shows the items and boosters.
<b>Alternative Course</b>	-
<b>Postcondition</b>	1. Player viewed Improve Window.
<b>Exceptions</b>	

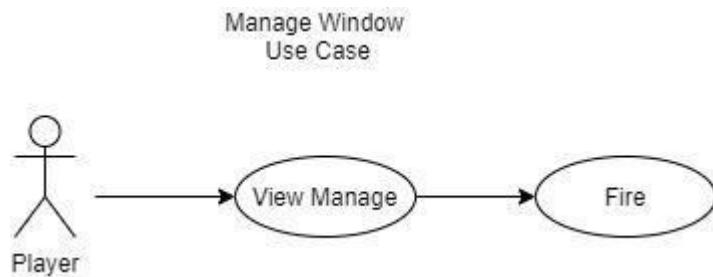
## Buy Use Case

<b>Use Case Name</b>	Buy
<b>Actor</b>	Player
<b>Description</b>	This use case allows the player to get boosts and boosters for their employees.
<b>Trigger</b>	Player clicks the "Buy" button in the Improve Window.
<b>Preconditions</b>	Player is in Improve Window.
<b>Normal Course</b>	<ul style="list-style-type: none"> <li>-Player clicks the "Buy" button in the Improve Window.</li> <li>- If the player has enough money, player can buy that items or boosters.</li> <li>- The main money the player has is reduced to the amount of the items or boosters player received.</li> <li>- The system will show the items or boosters purchased by the player in their inventory for later use.</li> </ul>
<b>Alternative Course</b>	<ul style="list-style-type: none"> <li>- If the money the player has is not enough to buy that item, player will not be able to own that item.</li> </ul>
<b>Postcondition</b>	<ul style="list-style-type: none"> <li>- Items or boosters purchased by the player are added to their inventory.</li> </ul>

## “Use” Use Case

<b>Use Case Name</b>	Use
<b>Actor</b>	Player
<b>Description</b>	This use case allows the player to use their own items or boosters.
<b>Trigger</b>	Player clicks the "Use" button in the Improve Window.
<b>Preconditions</b>	Player is in Improve Window.
<b>Normal Course</b>	<ul style="list-style-type: none"><li>-Player clicks on the " Use " Button in the Improve Window.</li><li>-Player uses their items or booster to increase employee morale and enthusiasm.</li><li>-Each items or booster can only be used once.</li><li>-Used items will be deleted from the inventory.</li></ul>
<b>Alternative Course</b>	-
<b>Postcondition</b>	-System uses the items or booster owned by the player.

## View Manage Use Case

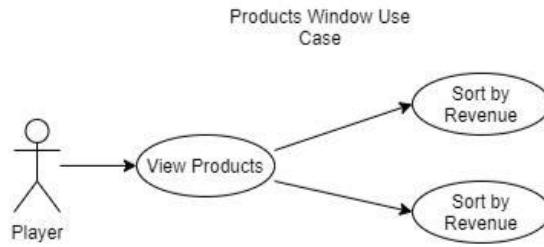


<b>Use Case Name</b>	<b>View Manage</b>
<b>Actor</b>	Player
<b>Description</b>	This use case allows you to fire employees owned by the player.
<b>Trigger</b>	Player clicks the "Fire" button in the Manage Window.
<b>Preconditions</b>	Player is in Manage Window.
<b>Normal Course</b>	<ul style="list-style-type: none"> <li>-Player clicks on the "Fire" button in the Manage Window.</li> <li>-The selected player is removed from the Manage Window.</li> <li>-The player can then hire a new one instead of the employee she/he fired.</li> </ul>
<b>Alternative Course</b>	-
<b>Postcondition</b>	-The system removes the employee the player wants from the list.

## Fire Use Case

<b>Use Case Name</b>	Fire
<b>Actor</b>	Player
<b>Description</b>	This use case shows how the player fire their own employee.
<b>Trigger</b>	Player clicks the "Fire" button in the Manage Window
<b>Preconditions</b>	Player is in Manage Window.
<b>Normal Course</b>	-Player clicks on the " Fire" button in the Manage Window. -The system enables the player's own and chosen employee firing function.
<b>Alternative Course</b>	-
<b>Postcondition</b>	- The system removed the employee chosen by the player.

## 11. View Products Use Case



<b>Use Case Name</b>	<b>View Products</b>
<b>Actor</b>	Player
<b>Description</b>	This use case describes how player views details about products which are built by the companies represented in Stock Exchange.
<b>Trigger</b>	Player clicks the "Products" button in the Main Window.
<b>Preconditions</b>	Player is in Main Window.
<b>Normal Course</b>	-Player clicks the "Products" button in the Main Window. -System activates Products Window and shows the data about products.
<b>Alternative Course</b>	-
<b>Postcondition</b>	1. Player viewed the products.
<b>Exceptions</b>	

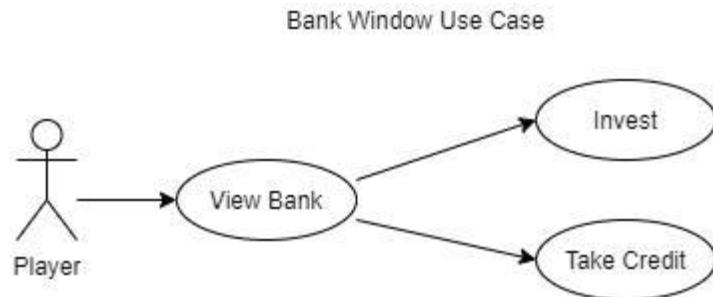
## Sort by Revenue Use Case

Use Case Name	Sort by Revenue
Actor	Player
Description	This use case describes how player sorts the products according to the revenue in the Products Window.
Trigger	Player clicks the "Revenue".
Preconditions	Player is in Products Window.
Normal Course	-Player clicks the "Revenue" title in the revenue column. -System ranks products by Revenue Value, from the highest revenue to lowest one.
Alternative Course	-
Postcondition	1. Player viewed the products sorted by Revenue.
Exceptions	

## Sort by Monthly Revenue Use Case

Use Case Name	Sort by Monthly Revenue
Actor	Player
Description	This use case describes how player sorts the products according to the monthly revenue in the Products Window.
Trigger	Player clicks the "Monthly Revenue".
Preconditions	Player is in Products Window.
Normal Course	-Player clicks the "Monthly Revenue" title in the monthly revenue column. -System ranks products by Monthly Revenue Value, from the highest revenue to lowest one.
Alternative Course	-
Postcondition	1. Player viewed the products sorted by Monthly Revenue.
Exceptions	

## 12. View Bank Use Case



<b>Use Case Name</b>	<b>View Bank</b>
<b>Actor</b>	Player
<b>Description</b>	This use case describes how player views Bank Window.
<b>Trigger</b>	Player clicks the "Bank" button in the Main Window.
<b>Preconditions</b>	Player is in Main Window.
<b>Normal Course</b>	-Player clicks the "Bank" button in the Main Window. -System activates Bank Window and shows options related to Bank.
<b>Alternative Course</b>	-
<b>Postcondition</b>	1. Player viewed Bank Window.
<b>Exceptions</b>	

## Take Credit Use Case

<b>Use Case Name</b>	<b>Take Credit</b>
<b>Actor</b>	Player
<b>Description</b>	This use case describes how player takes credit from Bank.
<b>Trigger</b>	Player clicks the "Take Credit" button in the Bank Window.
<b>Preconditions</b>	Player is in Bank Window.
<b>Normal Course</b>	<ul style="list-style-type: none"> <li>-Player clicks the " Take Credit" button in the Bank Window.</li> <li>-System activates Take Credit Window.</li> <li>-Player enters the desired amount of money to the input field and clicks on “Approve Amount” button.</li> <li>- Player enters the desired period of time to the input field and clicks on “Approve Period” button.</li> <li>-System shows the player total amount to be paid and the monthly amount to be paid to bank after take credit process approved by the player.</li> <li>-Player clicks on green “Take Credit” button.</li> <li>-System starts repayment process.</li> </ul>
<b>Alternative Course</b>	<ul style="list-style-type: none"> <li>-If player’s desired amount and time period are not acceptable by the bank, the bank refuses the player.</li> </ul>
<b>Postcondition</b>	<ul style="list-style-type: none"> <li>-Player’s desired amount of money added to the Current Balance.</li> <li>-System started repayment process.</li> </ul>

## Invest Money Use Case

<b>Use Case Name</b>	<b>Invest Money</b>
<b>Actor</b>	Player
<b>Description</b>	This use case describes how player deposits his/her money in Bank.
<b>Trigger</b>	Player clicks the "Invest Money" button in the Bank Window.
<b>Preconditions</b>	Player is in Bank Window.
<b>Normal Course</b>	<ul style="list-style-type: none"> <li>-Player clicks on the " Invest Money " button in the Bank Window.</li> <li>-System activates Invest Money Window.</li> <li>-Player enters the desired amount of money (which to be given to Bank as deposit) to the input field and clicks on “Approve Amount” button.</li> <li>- Player enters the desired period of time to the input field and clicks on “Approve Period” button.</li> <li>-System shows the player the amount of money which to be obtained after the approved period is passed.</li> <li>-Player clicks on green “Invest Money” button.</li> <li>-System starts Invest Money process.</li> </ul>
<b>Alternative Course</b>	-
<b>Postcondition</b>	<ul style="list-style-type: none"> <li>-System subtracts player’s desired amount of money from current balance.</li> <li>-System started Invest Money process.</li> </ul>

## 5. Data storage

In our project, we do not need a relational database to store the data we need and use in the game. Instead, we will be using JSON files to store data. JSON is a lightweight data-interchange format. It is easy for computers to parse and generate. JSON is based on a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages. For these mentioned purposes, we have chosen to use JSON in our project.

JSON files consists of three types of structures

- Objects
- Arrays
- Values

Unity offers unique classes to take full advantage of JSON type data storage solutions. In our project JsonUtility class will be used to read and store data. An example of this JSON file is given below.

JSON File

```
{  
  "gameData": {  
    "currentDate": "21-01-2021",  
    "userData": { ...  
    },  
    "company": { ...  
    },  
    "companies": [ ...  
    ],  
    "products": [ ...  
    ],  
    "contracts": [ ...  
    ],  
    "acceptedContracts": [ ...  
    ]  
  }  
}
```

This JSON file consists of one object called “gameData” which hosts other objects each of them storing necessary data about a specific part of the game.

User Data Object

```
"userData": {  
    "username": "Elon Tusk",  
    "skills": {  
        "chafferer": 1,  
        "decorator": 2,  
        "sensei": 3,  
        "business": 4  
    }  
},
```

“userData” object stores values about the user and its skills.

## Company Data Object

```
"company": {  
    "name": "BSG Inc.",  
    "currentValue": 148000000000,  
    "previousValue": 140000000000,  
    "officeType": 3,  
    "bankData": {  
        "currentBalance": 12343245,  
        "deposit": 20000,  
        "debt": 0,  
        "credibility": 0.76  
    },  
    "employees": [  
        {  
            "name": "Frank Salvador",  
            "age": 24,  
            "position": "Developer",  
            "rate": 55,  
            "salary": 1200  
        }  
    ]  
},
```

“company” object stores values about the company of the user.

## Companies Data Object

```
"companies": [
  {
    "companyName": "Epple",
    "currentValue": 350000000000,
    "previousValue": 325000000000
  }
],
```

“companies” object stores values about the companies generated randomly when a user first begins with a game.

## Products Data Object

```
"products": [
  {
    "name": "Doors",
    "producer": "BSG Inc.",
    "type": "Operating System",
    "platform": "Desktop",
    "quality": 5,
    "price": 150,
    "netUnitsSold": 123456654
  }
],
```

“products” object stores values about the products created in the game.

## Contracts Data Object

```
"contracts": [  
  {  
    "id": 1,  
    "company": "Megasoft",  
    "type": "Operating System",  
    "time": 5,  
    "offer": 65987  
  }  
,
```

“contracts” object stores values about the available contracts to the users which were randomly generated.

## Accepted Contracts Data Object

```
"acceptedContracts": [1, 2]
```

“acceptedContracts” object stores values about the contracts that have been accepted by the user.