BIG CENG THEORY

Software Requirements Specifications

Middle East Technical University

25.11.2011

INDEX

1. Introduction

This document is a software requirement specification for educational game called “Physics For Fun”. This is the temporary name of the game. Original name will be given to game after the detailed design according to the content of the problems. In this document we will firstly introduce the purpose and the scope of this document. Secondly, we will give an overall description of the game. After the general introduction, we will state specific requirements, data models and behavioral models with their descriptions in consecutive parts. Finally, we will introduce our plan for the game.

1.1. Problem Definition

The problem that leads us to develop an educational game is that high school students have some difficulties in learning physics. Therefore, making physics lessons more associated with real life and visualized is quite important for them when understanding basic subjects. In order to achieve this goal, creating a basis for creative thinking, reasoning and curiosity are key points of this problem. Besides these points, fun part of the learning is also a significant issue. Considering teenagers are spending most of their times on playing computer games rather than studying which also bothers their parents, integrating physics into the game can be very helpful at solving this problem. Also when properly used, games are the most inspirational elements in learning and personal progress. They help children create their own world and explore their environments. However, there are not rich content educational games in the game market of Turkey. This is the reason why we are developing an educational game for high school children.

1.2. Purpose

The purpose of this document is to give complete description of all functions and features of “Physics For Fun” in order to help decrease time and effort, that practitioner spend on developing, and create a basis for interfaces, functionalities and design. This document will be quite helpful for creating a basis between customer and supplier about the software product so that they can agree on the functions and features of the software to be developed.

1.3. Scope

The software product which we will develop is an educational game. This game will include physics subjects which are formally in the syllabus of first year of high schools. Each level of the game will focus on a different physics concept will be designed for the purpose of gaining student creative thinking, reasoning, questioning and reinvention. After developing the game, it will be available in the web page [1] of The Ministry of National Education under the name of lesson supporter games and it will be used in the tablet PCs which will be provided to children with the “Fatih Project” [2].

1.4. User and Literature Survey

In game market there are a lot of examples for educational games. Examining them is really important in order to find their inspirational elements to develop and/or find their weaknesses to avoid when we develop “Physics For Fun”. Here are the some good examples that can summarize the current educational game market:

Magic Pen [3]: This game is a level based game about balance and motion. User can draw any shape and attach them together in order to move a ball to reach the flag. Levels are very easy to pass at the beginning but getting really harder at each step. Despite of being very funny this game is lack of educational elements. It only covers simple machines. Therefore, this game is not suitable for our purpose except being addictive. We are targeting high school students but we can make them play our game by making it more addictive like this one.

Zoo Transportation [4]: This game is again level based whose goal is transporting animals to zoo without any harm is done to them. This game has the weakest fiction among all the games we searched. This game also focused on only one subject which we do not desire. The only element we like in this game is so sympathetic with animals.

MaddeyiTanıyalım 3 and KaldırmaKuvveti 2 [5] [6]: These are highly educational purposed games. They are based on question-answer matching which can be found boring by most of the students. They also have very irrelevant interface with questions: “In or Out?” format is opened after each correctly answered question. Only good thing in this game is using current contest format in the game.

The Transistor Recycler [7]: This game depends only on memorizing the objects which includes transistors or not. This game is really far away from being educational but a good implementation of making someone remembers something.

The intended audience of our product, as we mentioned in previous parts, is first grade high school students. We chose them in order to make this game create a basis in physics instead of making them learn directly a physics subject. Since the basic subjects are seen in first grade we chose their syllabus.

1.5. References

[1] Fatih Projesi.(n.d.) Retrieved from http: http://fatihprojesi.meb.gov.tr/site/

[2] Eğitim Bilişim Ağı.(n.d.) Retrieved from http: http://eba.meb.gov.tr/

[3] Magic Pen.(n.d.) Retrieved from http://www.learn4good.com/games/kids/physics_based_puzzles.htm

[4] Zoo Transport.(n.d.) Retrieved from http: http://www.physicsgames.net/game/Zoo_Transport.html

[5] KaldırmaKuvveti 2 EğitimAmaçlıOyunlar.(n.d.) Retrieved from http://www.sosyaldersleri.com/oyun.php?id=466&bas=4&turkodu=eslestirme&frontpage=0

[6] MaddeyiTanıyalım 3 EğitimAmaçlıOyunlar.(n.d.) Retrieved from http://www.sosyaldersleri.com/oyun.php?id=598&bas=4&turkodu=varmisinyokmusun&frontpage=0

[7] The Transistor Recycler.(n.d.)Retrieved from http://www.nobelprize.org/educational/physics/transistor/

1.6. Overview

In the following part, we will give information about overall description of the software, introducing product perspective, product functions, user characteristics and constraints, assumptions and dependencies. In the third part, we will introduce specific requirements. After third part we will state data model and descriptions, behavioral model and description and our plan for the “Physics For Fun” in consecutive chapters.

2. Overall Descriptions

In this part the general factors that affect the software and its requirements will be explained without details. The detailed information will be given in section 3.

2.1. Product Perspective

The block diagram shows the general structure of the product. There will be one user that is in interaction with the game. The game contains all the information that is needed for our product. When user wants to play the game, he/she can either continue to the previous game or start a new game. If user wants to start a new game, he/she has option to select an avatar or start the game immediately without selecting an avatar. When avatar is not selected, a default avatar will be assigned.

Moreover, the user can also get some help about the game or exit the game without playing.

The game contains some levels that the user has to accomplish a set of tasks and tries to solve different problems in order to reach a final point. While playing, the user will be able to get a map that is provided by the game. The map will help the user to see his/her progress during the game.

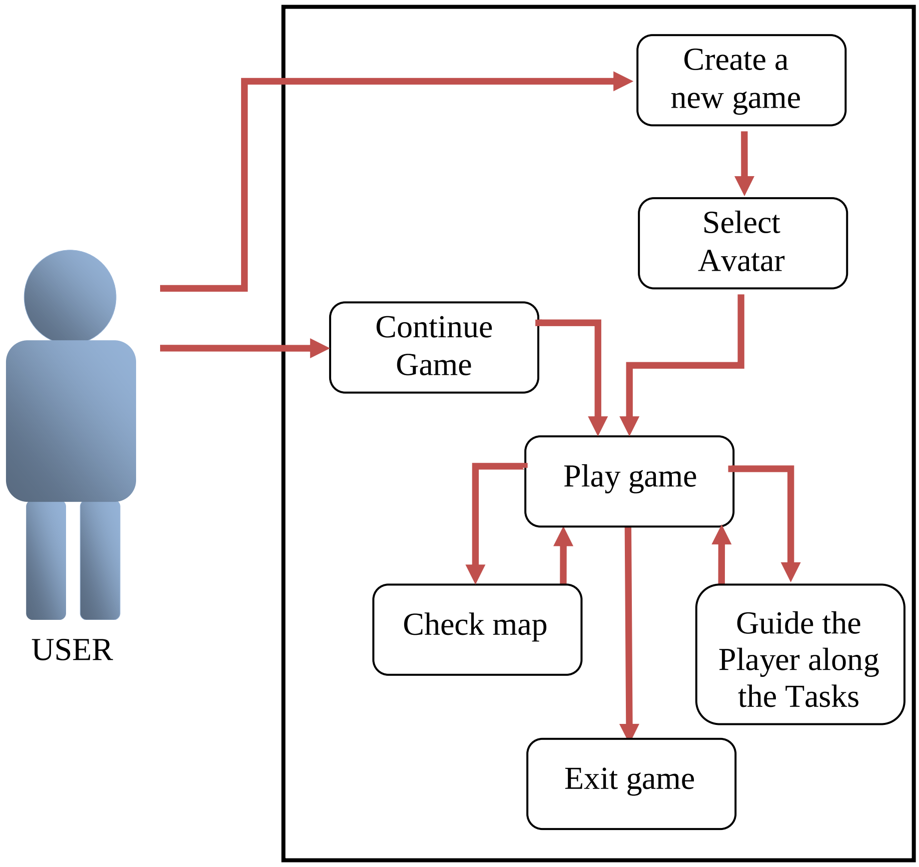


Figure-1 Product Functions

2.2. Product Functions

In this part, the systems functions that will be used for our system will be explained.

- “Create a New Game” function enables user to start a new game after selecting some options from the main menu.

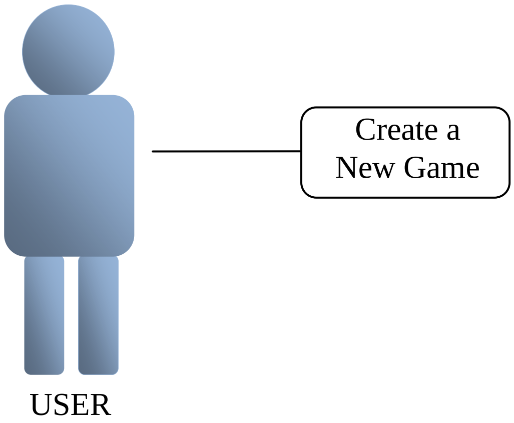


Figure-2 Create a new game use case

- “Play Game” function enables user to start the game.

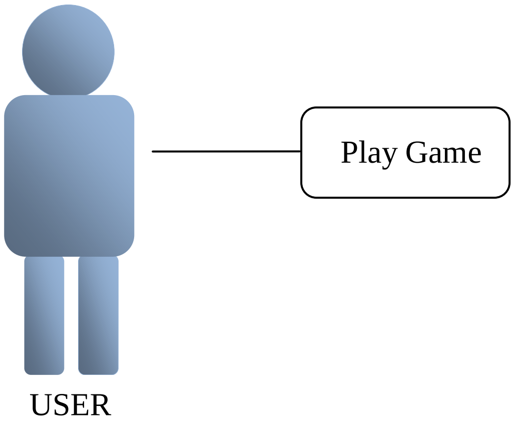


Figure-3 Play game use case

- “Select Avatar” function enables user to start select an avatar before starting to play the game.

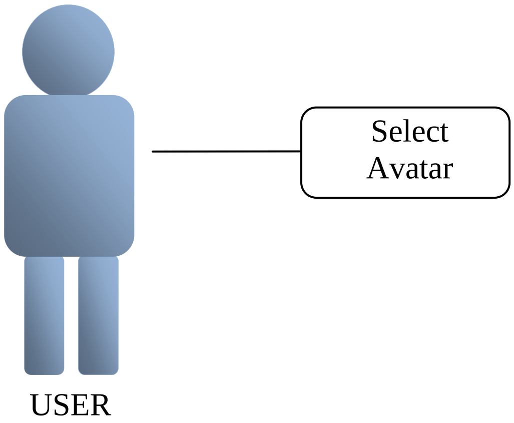


Figure-4 Select avatar use case

- “Check Map” function enables user to see his/her progress during the game by giving information about how many tasks are remained to complete the level and in which year the player is in.

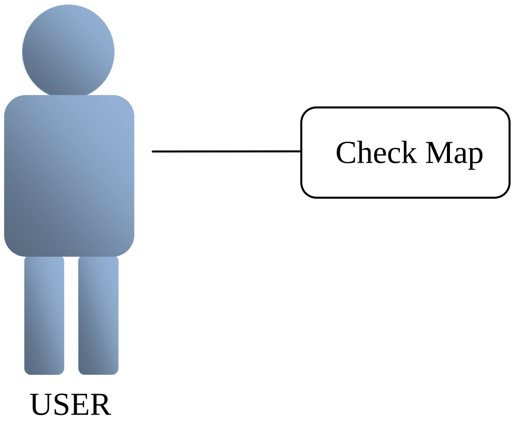


Figure-5 Check map use case

- “Exit Game” function enables user to save his/her game and exit from the system.

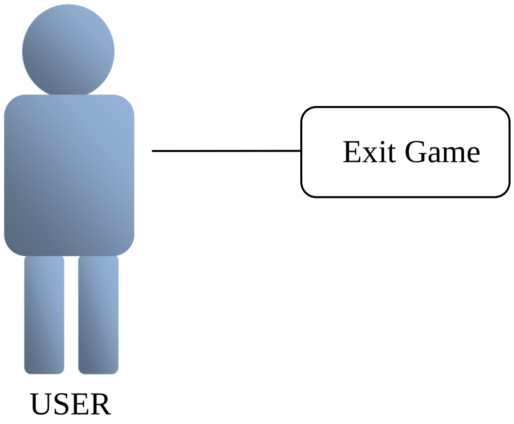


Figure-6 Exit game use case

- “Continue Game” function enables user to continue to play the game from where he/she was stayed.

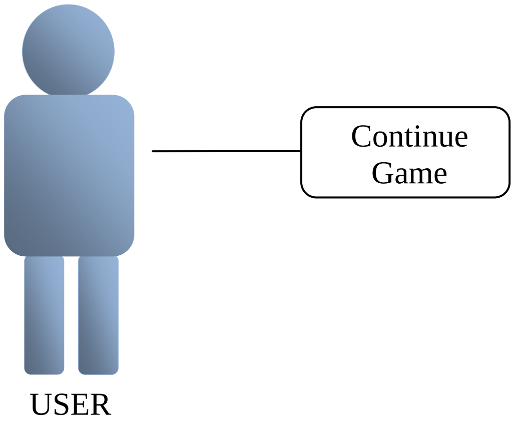


Figure-7 Continue game use case

- “Guide the Player along the Tasks” function enables user to get some help from a character during the game.

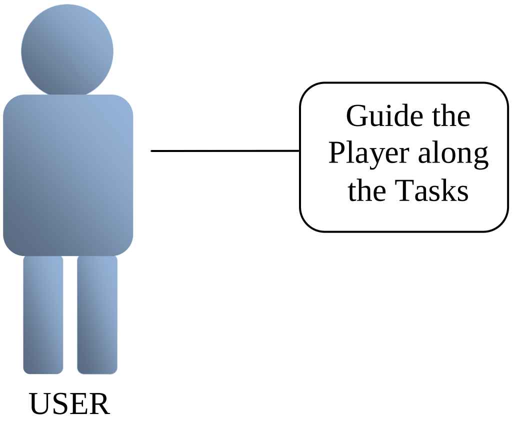


Figure-8 Guide the Player along the tasks use case

2.3. Constraints, Assumptions and Dependencies

The game will be used only with one user. No other users can connect to the game at the same time. If the player does not do any action during a predefined time interval, the user will be out of the game immediately. The number of avatars that the user will be able to select before starting to the game will be limited. The game will be in Turkish. No other language will be provided to the user. Since the game will be first used for “Fatih Project”, it will only be used in tablets. We assume that the operating system that will be used for this tablet will be Android OS.02 and “Unity3D” will be used to develop the software.

3. Specific Requirements

3.1. Interface Requirements

3.1.1. User interfaces

In our software there are five interfaces: game, map, menu, help and avatar selection screens.

Menu screen: In the menu screen there will be 4 buttons that player can go to avatar selection screen, help screen, map screen and game screen. These buttons will include appropriate icons on them within a harmonic design on a later decision. The purpose of this design is that this project needs to be appealing to teenagers and make them want to play this game.

Help screen: In this screen there will be the summary of the game and the information of how the game can be played. For example: the actions in order to move the character will be shown clearly in the screen.

Map screen: This screen will simply display a map with task information. It will help user to understand which level and task he/she is in the game. This screen will be dynamic and change with every task solving and level passing.

Avatar selection screen: This screen is providing users a choice on their avatars. There will be predefined avatars side by side in the screen so that user can select the one he/she likes and be more connected to the game. The avatars' design need to be very attractive to teenagers because these avatars will represent them.

Game screen: This screen is the one that the player will see during the game playing except another non game screen is opened for a while. The whole game story will appear in this screen according to the player's actions and the story together. The most important feature of this screen is it will be highly designed for the purpose of making teenagers enjoy during solving a physics-based problem.

3.1.2. Hardware interfaces

There will be no hardware interfaces included in our project.

3.1.3. Software interfaces

Developed game is intended to be distributed to all the first grade high school students under "Fatih project" which is mentioned before. Since Fatih project is planning to provide them tablet pcs, game will be developed for tablet pcs. The detailed specifications that the tablets will have has not determined yet. For now, we are assuming that the project will be developed in Unity3D and will be an application for an Android operating system.

3.2. Functional Requirements

3.2.1. Create a New Game

3.2.1.1. Background Information

When the players open the game, after completing and determining some other options, they will be able to create a new game based on the options they previously declared.

3.2.1.2. Description

Primary Actor	Players
Goal in Context	The aim of this function is to create a game based on the options which was previously declared by the player.

3.2.1.3. Normal Flow of Events

1. Player opens the game.
2. After determining some options, which will later be further explained in this document, game is created based on those options.
3. Player starts the game created.

3.2.2. Play Game

3.2.2.1. Background Information

That is one of the main functions of the system. Function which will be mostly used is that one. After the game is created by the system, players will be able to start playing the game.

Game will consist of levels which players will have to accomplish a set of tasks and solve various problems related to the topic to get through. To maintain the flow of the events in order so that player can see the bigger picture and feel like the story is processing, there will be animations prepared between levels.

We divided those videos into three categories based on where they are going to be have:

1. Introductory Video
2. Level-transition Videos
3. Finish Video

Introductory video is the one which opens when the game begins. It includes a little story behind the game, and what is expected from the player to accomplish tasks and finish the game.

Level-transition video will also have the same informational purpose, as well as player will see the outcomes his/her accomplished task and the player will be assigned a new task and will take crucial information regarding to that mission. This videos will be highly important since they will help to keep players in the game and will give the sense of success.

Finish video is the one that will be displayed in the end of the game in the case that the player completed all the levels successfully. This video needs to be very complementary to the players and makes them see themselves as heroes and heroines. The reason for this is to make students more enthusiastic about physics by making them feel happy by solving physics problems.

In order to complete each level, players will be assigned a task so that they will get an intuition about the physics-related topics, so the task will be prepared accordingly.

3.2.2.2. Description

Primary Actor	Players
Goal in the Context	This function is crucial since it will enable players to actually play the game, and make the transitions between levels.

3.2.2.3. Normal Flow of Events

1. Player starts the game which was previously created by the system.
2. Introductory movie is shown and player gets the necessary information regarding to the story of the game and is assigned to his/her first task to complete.
3. Player completes solves the problems and completes each level successfully and sees the level-transition videos as he/she goes along.
4. After completing each and every level, finish video is shown and players are directed to the main menu.

3.2.2.4. Alternate Flows of Events

3.2.2.4.1. Alternate Flow – 1

1. Player starts the game which was previously created by the system.
2. Introductory movie is shown and player gets the necessary information regarding to the story of the game and is assigned to his/her first task to complete.
3. Player fails in one of the levels.

3.2.2.4.2. Alternate Flow – 2

1. Player starts the game which was previously created by the system.
2. Player will be able to skip the videos which are shown in between the levels.
3. Player completes all the levels.
4. Finish video will not be able to be skipped, player will watch it.
5. Player will finish the game and redirected to the main menu.

3.2.3. Select Avatar

3.2.3.1. Background Information

To make the player more connected to the character created in the story, and so more connected to the game and make him/her want to play the game more, there will be a section to choose an avatar between previously-determined and created ones. However, player will not be able to change this avatar during the game, this option will only be available at the start of the game. Later on, player will have to stick to his/ her avatar. If he/she wants to change the avatar, then he/she will have to start the game from the very first level. The purpose of this is to make the story more consistent.

Player will also be able to enter a name to his/her avatar. Later on, this name will be used during the game-playing and in the stories. If player does not want to use this function, a default avatar with a default name will be assigned.

3.2.3.2. Description

Primary Actor	Players
Goal in the Context	To make the players have the option to choose their own avatars with the names desired.
Trigger	Personalization of the character, thus the game.

3.2.3.3. Normal Flow of Events

1. At the start of the game, player will be directed to a menu.
2. Player chooses “Select Avatar” option.
3. Avatar options will be shown to the player and he/she chooses one of them.
4. A name box will appear on the screen and player will write a name he/she desires to have his/her character to have in the game.
5. Player starts the game created.

3.2.3.4. Alternate Flows of Events

3.2.3.4.1. Alternate Flow – 1

1. Player is in the menu.
2. Instead of choosing an avatar, he/she directly starts the game.
3. The default avatar is assigned.
4. Game starts.

3.2.3.4.2. Alternate Flow – 2

1. Player is in the menu.
2. Player chooses the option to select an avatar.
3. Player chooses an avatar between the choices.
4. Player forgets to enter a name and wants to start the game.
5. An error message is displayed on the screen to inform the user that he/she has to enter a name for his/her avatar.
6. Player enters the name.
7. Game is created and player starts the game.

3.2.4. Check Map

3.2.4.1. Background Information:

During the game at each level, player will be supplied a map from which he/she will be able to see the progress in the level. Those maps will be unique to each level, and in those maps, player will be able to get various information, like how many more tasks does he/she has to accomplish before completing the current level and going on the next one, which levels did he/she completed so far in the game, etc.

To see this map, player will have to click a button that will be supplied on the side and stay there during the whole game, unless user is either on the menu, or watching a video. Player will be able to go back and forth as many times as he/she wants between the game scene and this map, as it is not critical to see where he/she is in the map.

When player select to see the map, an aerial viewpoint of the environment of the current level will appear on the screen. On this view, there will be checkpoints. On each and every one of these check points, player will be assigned a new task. After checking the map, taking the necessary information, to continue the game, on the side there will also be a button that player will have to click to return the actual scene of the game.

Number of maps created will be equal to the number of levels. In the case of decision to add more levels to the game, news maps will be supplied.

Player will not be able to see previous maps from the levels he/she completed, however the necessary information will be supplied, so there will be no need to see the previous ones.

3.2.4.2. Description

Primary Actor	Player
Goal in the Context	To enable users to keep track of things what they have accomplished so far, and see how many more problems they have to solve to complete current level.
Trigger	To motivate the user by showing how much they've accomplished so far and to have an information mechanism for the user.

3.2.4.3. Normal Flow of Events

1. Player plays the game.
2. Player wants to see the map, and clicks the button on the side.
3. The map, which is updated for each and every level, appears on the screen.
4. Player gets the necessary information.
5. Player clicks the button on the side to resume the game.
6. Main scene of the game appears on the screen.
7. Player continues to play the game.

3.2.5. Exit Game

3.2.5.1. Background Information

Player can leave the game by clicking exit game button which also enable player save this game. Since there will be many levels and many tasks in those levels, we do not expect players to complete the whole game in one go, so players will be able to save the game to play later. If there is a game already saved, that will be overwritten.

In the case of leaving a game before completing all levels, players will be asked if they want to save the game and then quit. Further actions will be taken by the system according to the answer came from the player.

3.2.5.2. Description

Primary Actor	Player
Goal in the Context	To enable user exit the game with or without saving game
Trigger	Getting rid of the limitation for the players having to complete game at once, and starting the game at each time.

3.2.5.3. Normal Flow of Events

1. Player clicks the “Return to Main Menu” button which is placed on the side.
2. Main Menu appears on the screen.
3. Player clicks the “Exit Game” button.
4. Player is asked whether he/she wants to save the game.
5. Player clicks the “Save Game” button.
6. If there is already a saved game, player gets informed about the situation and asked about what to do.
7. System takes actions accordingly.
8. Game is saved and closed.

3.2.5.4. Alternate Flows of Events

3.2.5.4.1. Alternate Flow of Events – 1

1. Player clicks the “Return to Main Menu” button which is placed on the side.
2. Main Menu appears on the screen.
3. Player clicks the “Exit Game” button.
4. Player is asked whether he/she wants to save the game.
5. Player clicks the “Don’t Save” button.
6. Game is closed without save game.

3.2.5.4.2. Alternate Flow of Event-2

1. Main Menu is on the screen.
2. Player clicks on the “Save Game” button.
3. However, he/she has not played a game yet, so there is no game to save, so game is closed.

3.2.6. Continue Game

3.2.6.1. Background Information

As our system will be able to save a game, this function will enable player to continue saved ones. If player chooses to continue a previously saved game, with the scene, map and the avatar state which is recorded before, game will be created so that player will be able to continue where he/she left.

If player chooses to start a new game, nothing will happen to the saved one, unless player wants to save a new game.

3.2.6.2. Description

Primary Actor	Player
Goal in the Context	To continue the previously-saved games.

3.2.6.3. Normal Flow of Events

1. Player opens the game.
2. Main Menu appears on the screen.
3. Player wants to continue to game which he/she saved the last time, therefore clicks the “Continue Game” button.
4. System creates the game accordingly.
5. Player starts playing.

3.2.6.4. Alternate Flows of Events

1. Player clicks the “Return to Main Menu” button which is placed on the side.
2. Main Menu appears on the screen.
3. Player clicks to quit the game without saving the current game.
4. Player is asked if he/she wants to save the game.
5. System takes actions accordingly.
6. Game is closed.

3.2.7. Guide the Player Along the Tasks

3.2.7.1. Background Information

Although there will be informative videos shown before and after every level, in those levels, some tasks may be way complicated for players to figure out what he/she will do in that task. In that situation, a character which is also embedded into the story of the game will help the players.

That character will either explain what he/she is expected to do in that task or what kind information player can use in order to complete current task, since the purpose of this whole game is to help students understand the topics which can be done better if players go further in the game.

3.2.7.2. Description

Primary Actor	Player
Goal in the Context	In order to guide the players and prevent them from losing themselves in the games.
Trigger	Some tasks are pretty hard to figure out even for some student who is academically very successful.

3.2.7.3. Normal Flow of Events

1. Player opens the game and starts playing.
2. The characters are shown from a box placed at top-right corner of the screen.
3. As player needs some guidance, this character gives some information and hints during the game.

3.3 Non-functional Requirements

In this part performance requirements and design constraints will be explained.

3.3.1 Performance requirements

The game will be developed on tablets and the operating system that these tablets are assumed to use is Android OS.02. Since the game is a part of the “Fatih Project”, the tablets that will be used have not been determined yet. Therefore, the performance of the game will be determined after making a decision about which tablet to use. However, we assume that when user wants to start the game, system will respond in 0.1 seconds.

3.3.2 Design constraints

As we mentioned before, we will use Android OS.02 for our system. Therefore, the game will be developed using “Unity3D”.because it is a good way to design a game on Android platform.

4. Data Models and Description

4.1. Data Description

The complete data model of our project can be seen in Figure 9. Main data objects in “Physics For Fun” are Character and Game objects because the game is around the character and the story of the game. Non-Player, Item, Character, SavedGame, Map, and Task objects are associated with Game object. Also Character object is associated with Task and Item objects so that the player will be in contact with the tasks using items to pass levels. SavedGame object is simply used for storing the current game’s information to save that game. Following parts will introduce these data objects and their attributes in detail.

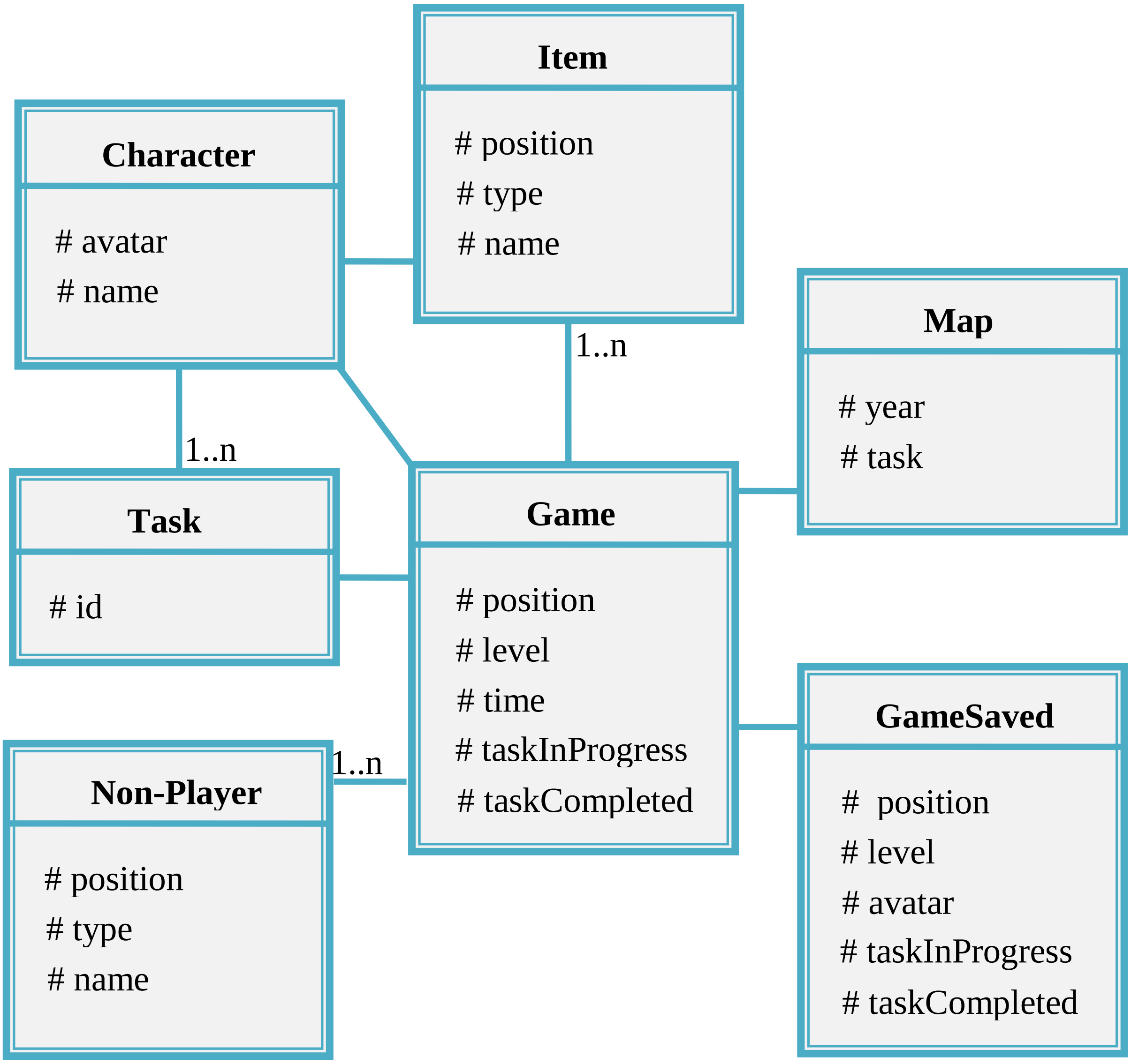


Figure-9 The Complete Data Model

4.1.1. Data Objects

In this part we will provide information for all data objects in the project with their function, semantic and attributes.

Character: This data object is representing the leading character of the game that is the center element and controlled by the user.

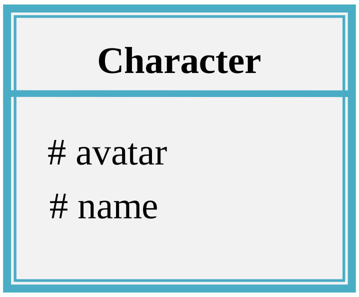


Figure-10 Character object

avatar: holds the avatar information of the leading character that is selected by the player.

name: holds the name of the leading character that is entered by the player.

Non-Player: This data object is representing the all other characters of the game which are not leading character and independent of the user.

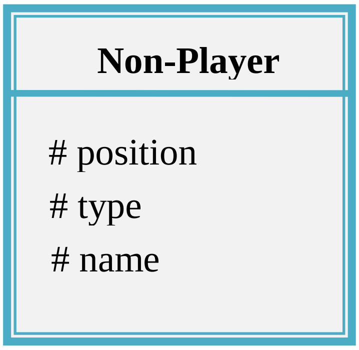


Figure-11 Non-player object

position: holds the position information of the character designed in the game.

type: holds the type information of the character designed in the game.

name: holds the name information of the character designed in the game.

Item: This data object is representing all the items that can be seen in the game which are going to be helpful for user to solve problems.

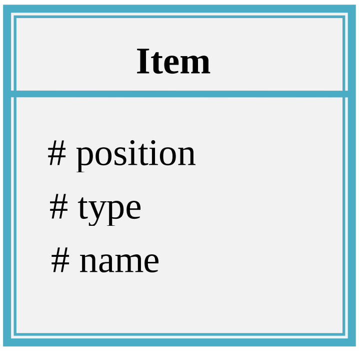


Figure-12 Item object

position: holds the position information of the item designed in the game.

type: holds the type information of the item designed in the game.

name: holds the name information of the item designed in the game.

GameSaved: This data object is for storing necessary information in order to save the game and play it later.

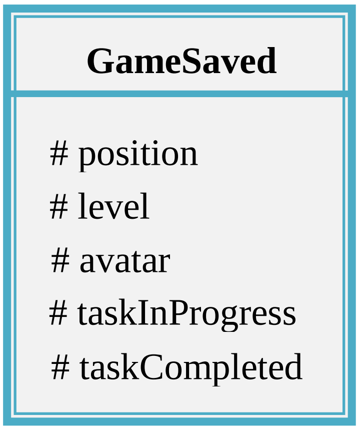


Figure-13 GameSaved object

position: holds the position information of the character in the game.

level: holds the level which the character exited and saved the game at.

avatar: holds the avatar information of the character designed in the game.

taskInProgress: holds the task which the character exited and saved the game at.

tasksCompleted: holds the tasks which the character completed before saving the game.

Game: This data object is representing the whole game and includes all the data objects.

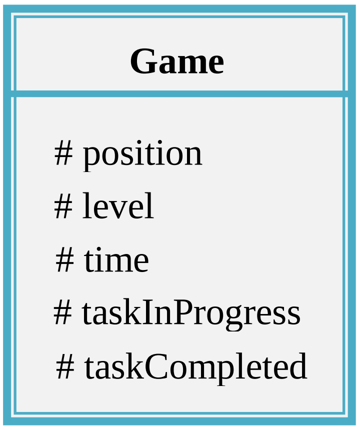


Figure-14 Game object

position: holds the position information of the character in the game.

level: holds the level information which the character is at in the game.

time: holds the time that the player spend on a level which he/she is currently at.

taskInProgress: holds the task which the character is at in the game.

tasksCompleted: holds the tasks which the character completed until the current task in the game.

Map: This data object is representing the map screen of the game in order to show user where he/she is currently.

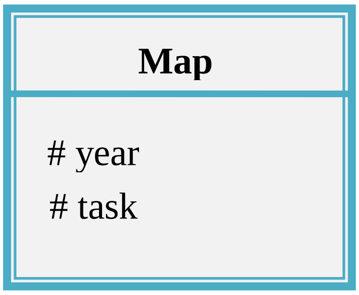


Figure-15 Map object

year: holds the year information of the game according to the levels.

task: holds the task information which the character is at in the game.

Task: This data object is representing each quest in the tasks of the game.

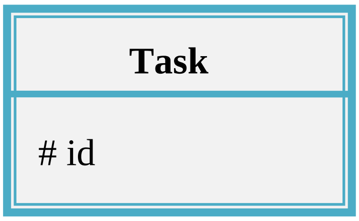


Figure-16 Task object

id: holds the id information of the quest according to the tasks.

4.1.2. Relationships

In this section we will introduce the relationships between all the data objects described above.

Character-Game: Game object has only one Character object in the game.

Non-Player-Game: Game object may have one or more Non-Player objects representing the characters in the game environment.

Item-Game: Game object may have one or more Item objects representing that there can be one or more items available in the game environment.

SavedGame-Game: Game object has only one SavedGame object.

Map-Game: Game object has only one Map object that is dynamically changed.

Task-Game: Game object may have one or more Task objects.

Character-Task: Character object may have one or more Task objects to pass in order to proceed.

Character-Item: Character object may have one or more Item objects to use in order to solve problems.

4.1.3. Complete Data Model

In this section, we provide a complete data model by an entity relationship diagram including all the relationships between data objects. This diagram can be seen in Figure 17.

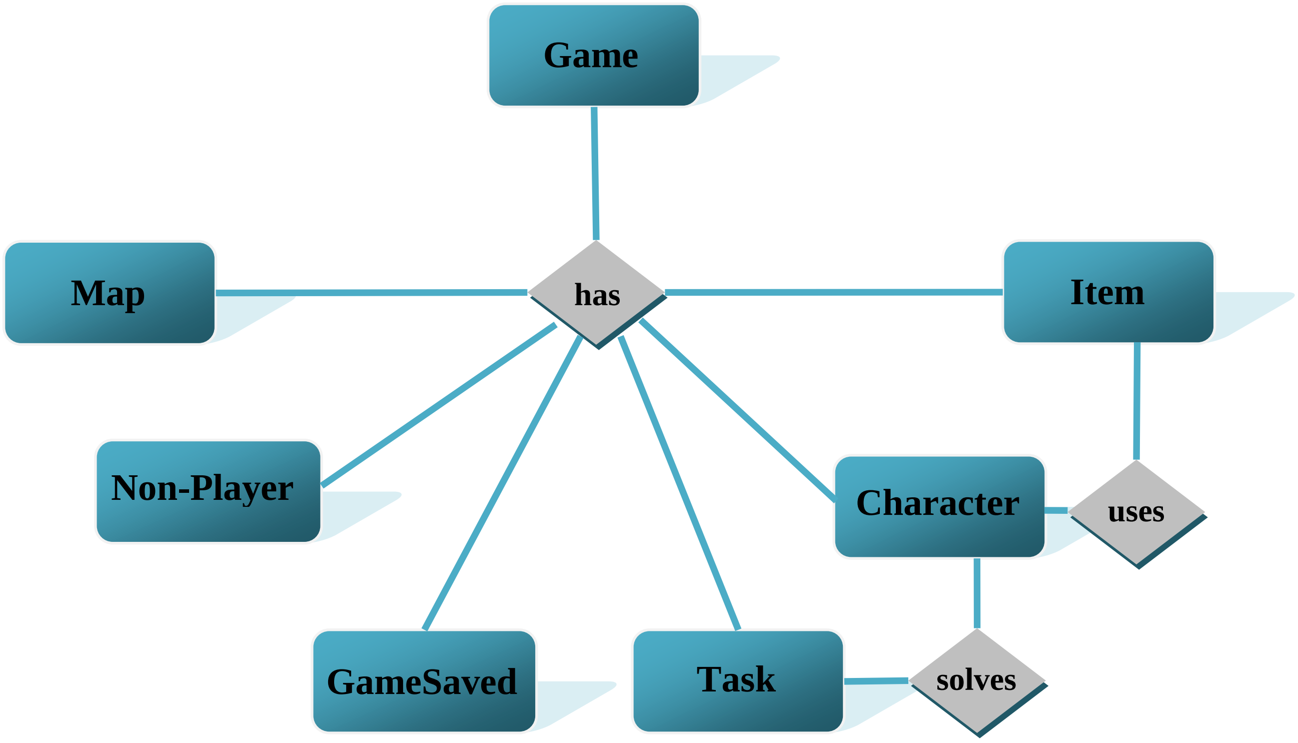


Figure-17 Entity relationship diagram

4.1.3. Data Dictionary

avatar: is a visual object in the game that represents the player. Players can separate themselves from other characters in the game by this visual. The most important part is that the avatar is under the control of the player.

character: means the player in the game environment, the hero/heroin of the game.

task: is the each problem need to be solved by the user in levels. There may be more than one task in one level.

level: is the each step of the game that the player has to pass in order to succeed.

Non-Player: means every other characters which are not under the control of the player in the game environment, the friendly ones or the enemy ones.

Item: means all the objects in the game environment available for player to use in solving the problems, for example; bridge, pen, tree etc.

5 Behavioral Model and Description

5.1 Description for Software Behavior

The game starts with a main menu. User has two choices in this situation, creating a new game or continuing the game which is saved last time. If user chooses new game, he/she can choose an avatar or start to play game without choosing avatar, and then a default avatar will be chosen. If user prefers to continue game, he/she start to play directly.

During the game, user confronts some task according to the story of game. When a task is given to user, achievement of task is required to continue the game. User will be guided according to the tasks in this period of the game. In any time of the game, user can click to map, which will be unique to each level, so get various information from the map. Then he/she can return to game and resume playing. When user wants to exit game, he/she clicks to exit button, and asked whether he/she want to save this game. So user can exit game with or without saving the game.

5.2 State Transition Diagrams

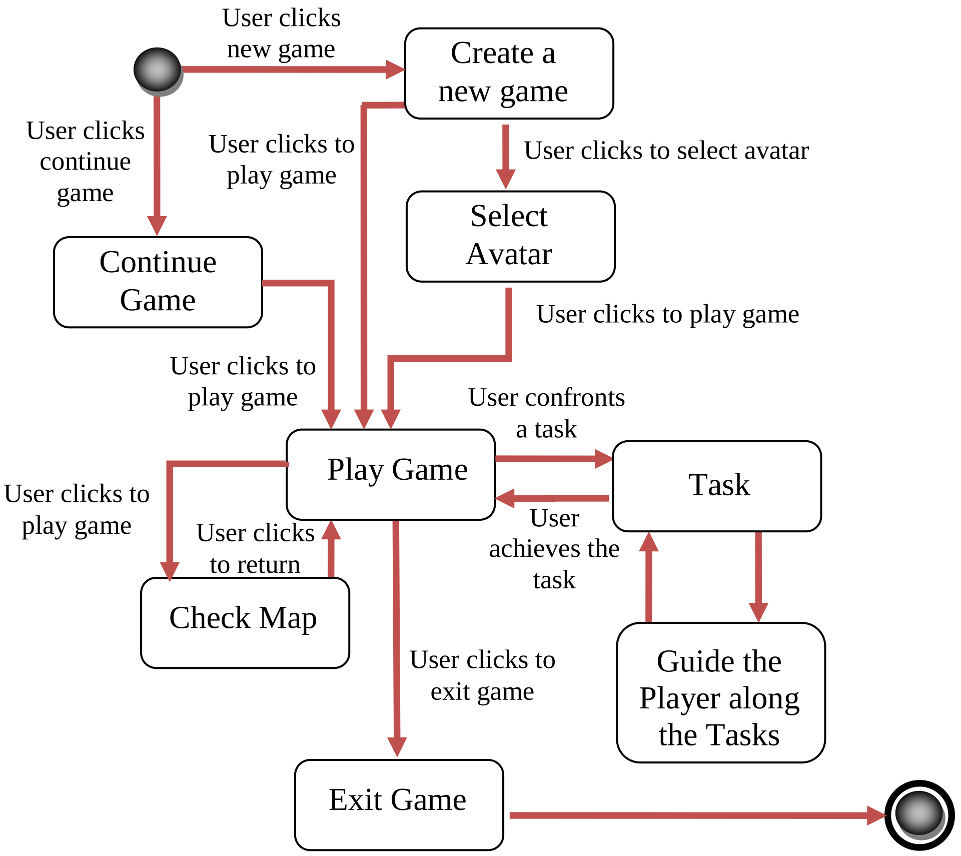


Figure-18 State transition diagram

6. Planning

6.1 Team Structure

Ezgi Nazlıbilek - Researcher, Game Module Developer, Software Engineering
Özgün Çiçek - Researcher, Game Module Developer, Public Relations
Mehmet Akçay - Researcher, Game Module Developer, Public Relations
Berat Ceren Doğankaya - Researcher, Game Module Developer, Software Engineering

Since we do not have any kind of database module in our game, it was not necessary for us to assign a database module developer to one of the group members. Throughout this game development, there will be a lot of communication between us and "words2inspire" charity, schools, designers and so on, so we assigned two of our group members to be in charge of Public Relations. Their duty will be to organize, and get feedback from our exterior environment on a regular basis.

Although we are going to create a game, our aim is mainly to educate 9th-grade students, in order to achieve this goal, there has to be an on-going research process which will include looking at different educational methodologies, looking at the trends among teenagers to make them more into playing the game, keeping ourselves updated about the subjects of curriculum, and so on. As a result, we assigned all of our group members as Researchers.

As a group, we do not have a leader. We wanted to have horizontal hierarchy instead of a vertical one. Since every one of the group members know their duties and responsible enough to handle those, so far we did not need the existence of a leader. Besides, we determined a strict hour. Every week at that hour, we gather around and discuss about the process of our games and new ideas that can be innovational and beneficial regarding our process and the game we are going to develop.

6.2 Estimation (Basic Schedule)

As seen in the Gantt chart in Figure-20 , the plan is to provide a detailed design report and a prototype demo to show in the presentations before the first semester ends. Next semester we are planning to make detailed implementation and system testing until June.

6.3 Process Model

The Waterfall Process Model which can be seen in Figure-19 is chosen for our project. Therefore, after the software requirements specification, we will write initial and detailed design reports and then start to develop the project.

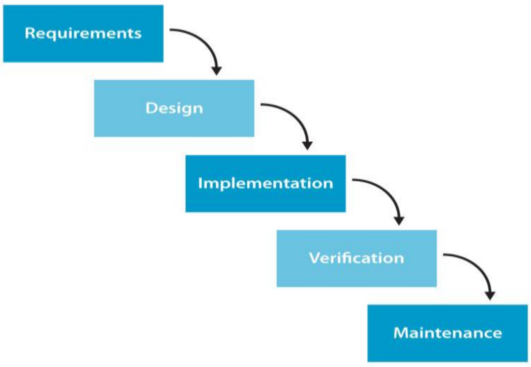
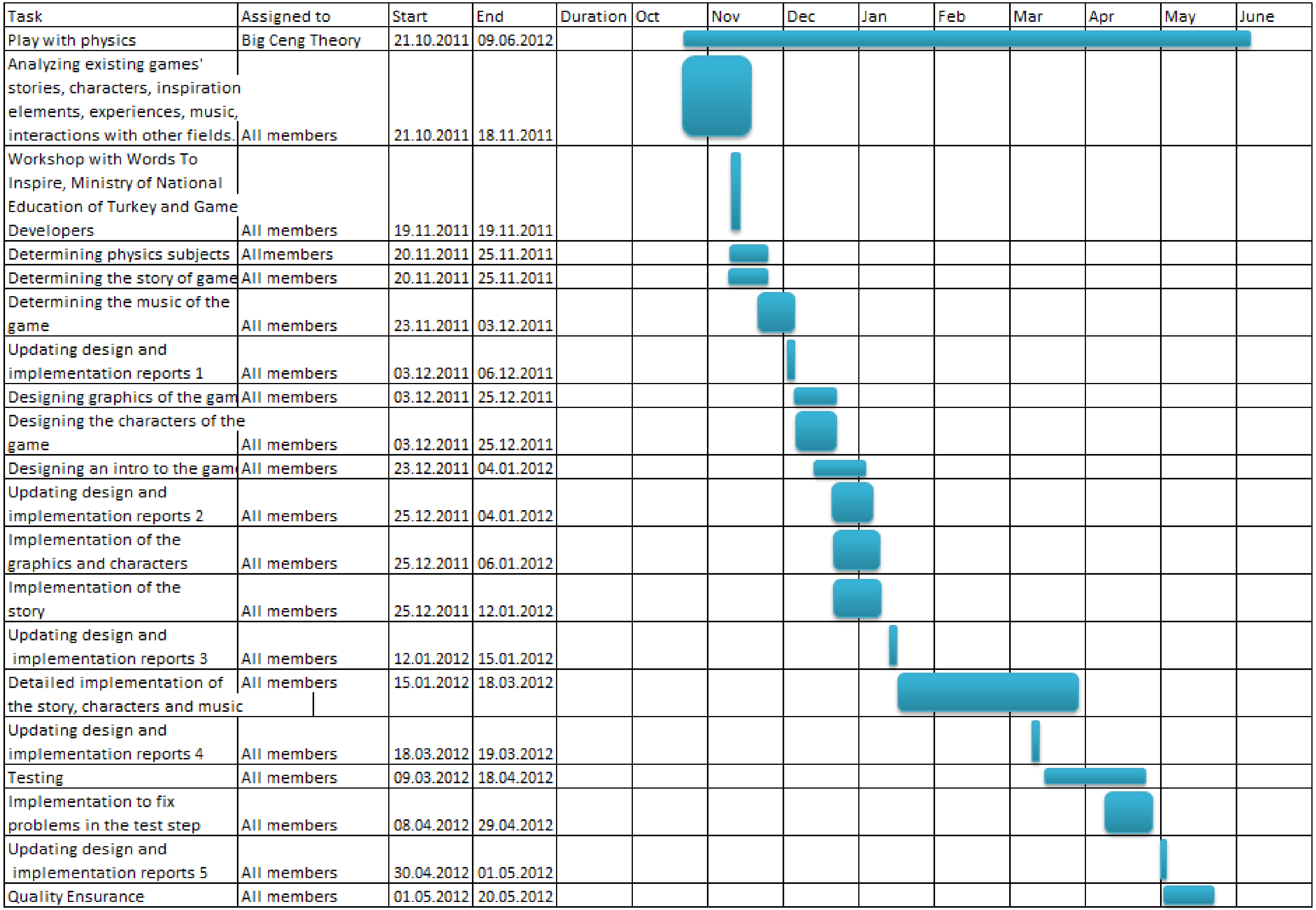


Figure-19 Waterfall process modal



Task	Assigned to	Start	End	Duration	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	June
Play with physics	Big Ceng Theory	21.10.2011	09.06.2012										
Analyzing existing games' stories, characters, inspiration elements, experiences, music, interactions with other fields.	All members	21.10.2011	18.11.2011										
Workshop with Words To Inspire, Ministry of National Education of Turkey and Game Developers	All members	19.11.2011	19.11.2011										
Determining physics subjects	Allmembers	20.11.2011	25.11.2011										
Determining the story of game	All members	20.11.2011	25.11.2011										
Determining the music of the game	All members	23.11.2011	03.12.2011										
Updating design and implementation reports 1	All members	03.12.2011	06.12.2011										
Designing graphics of the gam	All members	03.12.2011	25.12.2011										
Designing the characters of the game	All members	03.12.2011	25.12.2011										
Designing an intro to the gam	All members	23.12.2011	04.01.2012										
Updating design and implementation reports 2	All members	25.12.2011	04.01.2012										
Implementation of the graphics and characters	All members	25.12.2011	06.01.2012										
Implementation of the story	All members	25.12.2011	12.01.2012										
Updating design and implementation reports 3	All members	12.01.2012	15.01.2012										
Detailed implementation of the story, characters and music	All members	15.01.2012	18.03.2012										
Updating design and implementation reports 4	All members	18.03.2012	19.03.2012										
Testing	All members	09.03.2012	18.04.2012										
Implementation to fix problems in the test step	All members	08.04.2012	29.04.2012										
Updating design and implementation reports 5	All members	30.04.2012	01.05.2012										
Quality Ensurance	All members	01.05.2012	20.05.2012										

Figure-20 Gantt Chart

7. Conclusion

This Software Requirement Specification document is prepared to give requirement details of the project “Physics For fun”. First, general description of the game is given. Then, researches conducted about market are introduced. In the main part, all the functional, non-functional and interface requirements, data and behavioral models are stated in a detailed manner. Finally, planning that will be used during the development of the software is given. This document will be helpful at constituting a basis for design and development of the game to be developed.