**Learning through advertorial educational game**

**“The Adventure of a New Mother”**

**BY**

**Imran Hossain Shoikoth**

**ID: 1311063**

A REPORT PRESENTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

FROM THE DEPARTMENT OF

COMPUTER SCIENCE AND ENGINEERING



**INDEPENDENT UNIVERSITY, BANGLADESH**

**Autumn 2017**

**Supervisor: Prof. Farruk Ahmed**

All rights reserved. This work may not be reproduced in whole or in part, by photocopy or other means, without the permission of the author.

© Imran Hossain Shoikoth 2017

**Department of Computer Science & Engineering  
Independent University, Bangladesh**

Abstract

In Bangladesh, high proportions of infant deaths (two-thirds) and deaths among children aged less than five years (38%) occur in the neonatal period. Although most of these deaths occur at home due to preventable causes, little is known about routine domiciliary newborn-care practices and care-seeking for neonatal illness. As an initial step in strategic planning for the implementation of interventions in Bangladesh to improve neonatal outcomes, we created a **RPG role playing game** named “**The Adventure of a New Mother**”. In my game I created an adventuring way to teach mother’s how to take care to her baby. In each level of the game the mother (In-game) need’s to take care to baby and all on a sudden baby will be affected with different type’s disease then she need to contact to another character of our game named doctor. Doctor will tell her how what to do why this is happing, which vaccine need to give in which time period and many more. . The RPG-style mechanics and story mode making this game more interactive and interesting for the players. In the process, mothers will learn the maternal issues subconsciously while playing the games.

Attestation

I understand the nature of plagiarism, and I am aware of the University’s policy on this. I certify that this is an original work by me during. However, following internationally accepted academic guideline of using others written work and / or software (in the form of code) in my University project is properly cited if used in any part of this work.

|  |  |  |  |
| --- | --- | --- | --- |
| Signature: |  | Signature: |  |
| Name: |  | Name: |  |

Evaluation Committee

|  |  |
| --- | --- |
| Signature: |  |
| Name: |  |
| Supervisor: |  |
|  |  |
| Signature: |  |
| Name: |  |
| Internal: |  |
|  |  |
| Signature: |  |
| Name: |  |
| External: |  |
|  |  |
| Signature: |  |
| Name: |  |
| Head of CSE: |  |

Acknowledgments

The successful completion of this report might never be possible in time without the help some person whose inspiration and suggestion made it happen. First of all, I want to thank my faculty advisor and Supervisor **Prof. Farruk Ahmed, Professor** of School of Engineering & Computer Science for helping me completing my report on **Role-playing Educational Game as a Learning tool “The Adventure of a New Mother”.**

I would also like to thank my advisor **Aunnoy K Mutasim,** Junior Lecturer, Department of Computer Science and Engineering, Independent University, Bangladesh and **Dr. Shabareen Tisha**, Lecturer, School of Public Health, Faculty and Research Associate, Independent University, Bangladesh helped me by providing informative instructions. I was closely attached with them during my project. Without them, completion of this paper would have been very difficult.

I am also thankful to my parents, and friends who were also supportive in these six months of journey.

And finally, I also express my sincere gratitude to all those who participated to prepare the report.

Table of Contents

.

Abstract i

List of Publications ii

Attestation iii

Evaluation Committee iv

Acknowledgments v

Table of Contents vi

List of Figures viii

1 Introduction 1

1.1 Background and Context 2

1.2 Scope and Objectives 3

1.3 Achievements 4

2 State-of-The-Art (The literature review) 5

3 Software Requirements Specification of “The Adventure of a New Mother” 11

3.1 Introduction 11

3.1.1 Purpose of this Chapter 11

3.1.2 Document Conventions 11

3.1.3 Scope of this Document 11

3.2 General Description 12

3.2.1 Product and Business Perspective of the Game 12

3.2.2 System Environment 12

3.2.3 Quality Function Deployment of “The Adventure of a New Mother” 13

3.2.4 User Story of Our Game 14

3.3 Specific Requirements 15

**3.3.1.1** **User Interfaces** 15

**3.3.1.2** **Hardware Interfaces** 15

**3.3.1.3** **Software Interface** 15

**3.3.1.4** **User Characteristics for the System** 16

3.4 Analysis Model of Our Game Project 16

3.4.1 Scenario Based Model 16

**3.4.1.1** **Use Case Scenario** 17

**3.4.1.2** **Use Case Diagram with Use Case Description** 18

**3.4.1.3** **Activity Diagram** 25

**3.4.1.4** **Swimlane Diagram** 28

3.4.2 Data Model 32

3.4.3 Behavioral Model 32

**3.4.3.1** **State Diagram** 33

4 Implementation of the Proposed System 35

4.1 Product Design Terms 35

4.1.1 User Experience (UX) 35

4.1.2 Backend Programming 35

4.2 Assumptions and Dependencies 35

4.2.1 Construction of the Game 36

4.2.2 Integration with Android 36

4.3 Key Resource Requirements of the Project. 36

4.4 Implementation Tools Required 38

4.5 Implementation Code Example 39

5 Testing of “The Adventure of new Mother” 42

5.1 Test Case 1 42

5.2 Test Case 2 42

5.3 Test Case 3 43

5.4 Test Case 4 43

6 Discussion and Result 44

7 Some Snapshots 46

8 Conclusion 52

8.1 Summary 52

8.2 Evaluation 52

8.3 Future Work 52

References 53

Appendix 1 55

Appendix 2 – User guide 97

Appendix 3 – Installation guide 98

List of Figures

Fig 1: User Interection

Fig 2: Level 0 for Game

Fig 3: Level 1 for Game UX

Fig 4: Level 2.1(Play) for Game UX

Fig 5: Level 2.2(Options) for Game UX

Fig 6: Activity Diagram for “New Game” module of “Play” (Fig 3)

Fig 7: Activity Diagram for “Resume Game” module of “Play” (Fig 3)

Fig 8: Activity Diagram for “Exit Game” module of “Play” (Fig 3)

Fig 9: Activity Diagram for “Show Controls” module of “Options” (Fig 4)

Fig 10: Activity Diagram for “Change Sound/ Music Volume” module of “Options” (Fig 4)

Fig 11: Activity Diagram for “Story” module (Fig 1)

Fig 12: Activity Diagram for “Quit” module (Fig 1)

Fig 13: Swimlane Diagram for “New Game” module of “Play” (Fig 3)

Fig 14: Swimlane Diagram for “Resume Game” module of “Play” (Fig 3)

Fig 15: Swimlane Diagram for “Exit Game” module of “Play” (Fig 3)

Fig 16: Swimlane Diagram for “Show Controls” module of “Options” (Fig 4)

Fig 17: Swimlane Diagram for “Change Sound/ Music Volume” module of “Options” (Fig 4)

Fig 18: Swimlane Diagram for “Story” module (Fig 1)

Fig 19: Swimlane Diagram for “Quit” module (Fig 1)

Fig 20: top level state diagram

Fig 21: top level state diagram

Fig-22: Opening Scene 1

Fig-23: Opening Scene 2

Fig-24: Bedroom Scene

Fig-25: Dinning Room Scene

Fig-26: Bathroom Scene

Fig-27: kitchen Scene

Fig-28: Outdoor Scene 1

Fig-29: Outdoor Scene 2

Fig-30: hospital Scene

Fig-31: Bath Scene

# Introduction

The aim of this paper is to investigate the influence of a role-playing educational mobile game on the learning activities of girls, women and most importantly pregnant women. The “game” concept represents a structured or semi-structured activity with goals that players try to achieve and a set of rules control its operation. The associated concept of "play" relates to the interaction with the game itself. A game can be instantiated for learning, entertainment or recreation as it involves mental (and sometimes physical) stimulation and develop practical skills a game force the user, much more than other media, to decide, to choose and to define priorities.

With this aim, we first need to clarify the concept and varieties of mobile games; uses of them and success and failure of their intentions. Mobile games are similar to computer games that are constructed also with the help of programs.

Before the days of dual-core processors and wireless internet providers, the mobile phone had only the most basic of games by the end of the 20th century. Many consider Snake, first seen in 1997, to be the original mobile game. This simple 2D game involved directing an ever-growing snake around the screen to pick up items. While exceedingly plain, it began a craze for mobile gaming that would continue on indefinitely. The advent of wireless application protocol (WAP) technology was a significant development for mobile gaming. Not only did this service allow users to download basic games onto their handsets, it enabled multiplayer support. Early WAP titles usually took the shape of text adventures - which were a great fit for mobiles of this era - though it was titles like Jamdat's multiplayer combat fest *Gladiator* and nGame's hacker sim *DataClash* that pushed the boundaries in the monochrome age. From this point, portable technology advanced rapidly and the mobile gaming gained serious momentum. But the biggest shake up has to be in 2007 when on January 9th the iPhone touchscreen was announced, and consumer frenzy was born. This excellent piece of technology worked well and with the launch of the App Store in July 2008, the mobile games industry had a platform that enabled consumers to buy their favourite games directly to their phones via iTunes.

There are many categories of mobile games but top Mobile Game Categories are Strategy, Trivia, Adventure, Family, Role Playing , Simulation, Action, Puzzle, Educational and Arcade according a report by Apsalar’s Big Data Lab. **Role-playing Games** leads a player to a fantasy world, where the player can think that s/he is the main character and makes decisions based on the story line of the game and **Educational games** are used to teach people certain subjects e.g. vocabulary, mathematics, science and ICT. Educational games try to make learning a topic fun and interactive and engage the learner in the subject. This type of games can teach both adults and children. One of most popular RPG and Educational game is “The Sims game Series (2000-2017)”. Other Popular games are “Pokemon game Series (1996-2016)”, “The Legends of Zelda(1986-2017)”, “Warcraft”, “Final Fantasy”, “Doom and Destiny Advanced”, “Pixel Dungeon”, “Battleheart”, “ZENONIA 2” etc. All this software and apps are in English, a few of the popular games can be played in other languages, as for example, French, German, Italian, Spanish, etc. (Wikipedia, 2017). This research finds no RPG based educational game in Bengali language.

In “**The Adventure of a New Mother”** game, user start game after the baby born, and she/he have to perform specific tasks regarding maternal issues and baby care, to complete each mission to get to the end, also apart from the main mission users have to perform some regular tasks. In this process, they will get to know about unknown maternal and child care facts.

The first commercially available role-playing game, [Dungeons & Dragons](https://en.wikipedia.org/wiki/Dungeons_%26_Dragons) (*D&D*), was published in 1974 by Gygax's [TSR](https://en.wikipedia.org/wiki/TSR,_Inc.). TSR marketed the game as a niche product. Gygax expected to sell about 50,000 copies. After establishing itself in boutique stores it developed a cult following.

Logo (1967) is perhaps a strange inclusion on a list full of games, as it is actually a programming language, but its early application in education and use as a fun way to teach programming and mathematical concepts earn it a solid place on any list of foundational computer programs in education. Students will primarily remember Logo through its use of a turtle-shaped icon, which could be moved and altered. Through inputting commands, essentially very basic programming codes, students could use the turtle to draw geometric shapes, from circles to stars to spirals. While Logo’s use peaked during the mid-1980s, it was nonetheless pivotal in the development of educational programs, teaching a generation of kids that programming wasn’t only accessible, it could also be fun.

## Background and Context

Deaths of newborns within 28 days of birth are a major barrier to improving the survival of children aged less than five years (under-five children) in developing countries. Neonatal deaths now account for more than two-thirds of all deaths in the first year of life and for about half of all deaths in under-five children. Bangladesh has a neonatal mortality rate of 41 per 1,000 livebirths, and neonatal deaths account for about half of deaths of under-five children. Therefore, appropriate interventions are crucial for improving the health of under-five children in Bangladesh and to help achieve the global target of reducing under-five mortality by two-thirds. Information on the timing and causes of neonatal deaths can help direct appropriate interventions.

According to the World Health Organization (WHO), preterm birth accounts for 30% of global neonatal deaths, sepsis or pneumonia for 27%, birth asphyxia for 23%, congenital abnormality for 6%, neonatal tetanus for 4%, diarrhea for 3%, and other causes for 7% of all neonatal deaths. However, these estimates are based on limited datasets as most births and neonatal deaths occur in the home or outside formal health settings in developing countries. Game can be an appropriate tool as it used to create awareness.

We know how to prevent unnecessary newborn mortality. Quality care around the time of childhood including simple affordable steps like ensuring early skin-to-skin contact, exclusive breastfeeding and extra care for small and sick babies can save thousands of lives every year. Nearly half of all under-five deaths are associated with undernutrition. My game “The Adventure of a New Mother” will help to create awareness to help prevent this high dead rate. Thorough out the game player will get a lot’s of key information’s like impotency of breastfeeding under 6 month, symptoms of different common disease like diarrhea, poliomyelitis, ham etc. This not boring text book style learning fully adventuring task given model with lots of voice, funny animation and many more.

## [Scope](http://www.cs.stir.ac.uk/~kjt/research/conformed.html) and Objectives

This Report describes all the requirements for the project. The purpose of this research is to provide a virtual image for the combination of both structured and unstructured information of our project “The Adventure of a new, Mother”. This is a Role-playing game on the Android platform. The player will progress through levels which require completion of list of tasks, though the game Learning and performing timely via branching pathways. The episodic structure of the game facilitates the pace of the story. We demonstrate the action flow between inputs, script, display (output). We are working mainly with story, levels, object, animation, graphics, scripts, game engine facilities.

## Achievements

1. Now we know much more about game engines. How it works? The properties, objects and others.

2. We know how a model is constructed and how it is animated.

3. The main thing is that as a software engineer, skill and expertise to create a SRS document and an overall software product report is now better than before.

4. Co-Operation between group members.

5. Develop communication skills

6. Growing creative thinking and imagination capability.

# State-of-The-Art (The literature review)

**Literature Review in Game-Based Learning**

**Introduction**

The first computer games appeared in the fifties of the 20th century - since then, their development has proceeded at a vertiginous speed. It was almost impossible to assume that they will become one of the dominant social phenomena, and that, in the last decade of the 20th century, the industry of computer games generate more revenue from the film industry.

The popularity of computer games led to thinking about their application in education. Games became an integral part of modern society. They are the ideal platform for presenting new content and new technology - a lot of people play computer games and accept them as a normal form of entertainment. Research shows that not only youth who play games - big part of the playing population are adults. According to the report of the American Association for entertainment software (ESA - Entertainment Software Association) in 2006. The 69 percent of the U.S. population plays video or computer games. The average age of players is 33, and 25 percent are older than 50. Men make up 62% of the population.

Electronic games are a new mass medium, with its characteristics, as compared to the now traditional media such as books, television, film or music. In contrast to all existing media, games have the opportunity to interact, allowing the user to actively participate, not just passively receive information. That is why the last few years we can see more and more use of computer games for education. Educational games are social, card, or computer games that are specifically designed to teach people about a certain subject, expand concepts, reinforce development, understand an historical event or culture, or assist in the development of certain skills.

**Educational Games**

The paper focuses on educational computer games [3]. They combine education and entertainment in a concept known as edutainment. This concept usually assumes that users provide lessons in a fun environment recognizable: television, computer games, movies, music, websites, multimedia software and so on.

It is known that computer simulation has long been used to train civilian and military pilots. Modern flight simulation games are so highly developed that greatly surpassed the commercial software, especially in the quality of sound, graphics, and the degree of realism in general. In Germany is already working on a program very similar to computer games that simulate driving a car in the city and on the open roads. The advantage of these flights and car-simulation is that the represent a cheap way to simulate incidents and enhance response by participants. Disadvantage is that in comparison to modern computer games they reminiscent to clumsy attempts at 3D animation of the twentieth century.

There are indications that even a shooting game in first person in some areas can be used as an educational tool. Some armies take them to be cost-effective way to supplement tactical knowledge of military and anti-terrorist unit, as well as orientation and coordination skills. Many social and computer games that are not intended to be strictly educational in themselves carry a significant educational aspect. Monopoly teaches us basic principles of market economy, Microsoft's Age of Empires series, Total War games, and Civilization teach political economy, history, military theory, and even sociology and ecology. There are also many games that teach management: Transport Tycoon (management of urban and intercity transport), Railroad Tycoon (managing railway), Rollercoaster Tycoon (managing amusement park), Sim City I-IV (management of the modern city), Football Manager (managing a football club) and many others.

It is significant to mention the educational games for kids. Sales of classic toys shown a downward trend from year to year, and the kids all spend more time next to the screen. Thus, this genre of computer games is becoming extremely important. Children's educational games are educational in the true sense of the word. There are games for all ages, from those that resemble picture books in electronic format, to games like The Sims to help teenagers to cope with and resolve problems in the real world.

We decided to name the classification given by Chris Crawford, in his book 'The art of computer games' [4], published for the first time back in the 1970th year. It is a classification by type of game, and essentially cover the actual games today.

**Educational games effects on people**

Treating attention deficit, resulting in improved scanning and tracking (Larose *et al*. 1989). It was found beneficial if the game becomes progressively more difficult as attention wanes (Pope and Bogart 1996).

Providing cognitive–attentional distraction (eg in treating pain and nausea). Used, for example, with patients suffering from the effects of a stroke or burns or with children undergoing chemotherapy (Vasterling *et al*. 1993).

Treating schizophrenia (Samoilovich *et al*. 1992) and promoting and increasing motor skills (Sietsema *et al*. 1993).

Computer games have also been useful in supporting analysis into the development of attention in children: PlayStation 2 games were found useful at Bangor University in assessing children’s competence at visual processing (Kirriemuir 2002).

An attractive element of the gaming experience as a learning tool is that it provides opportunities for continued practice because negative consequences are not typically associated with failure. Rather, failure serves as an integral part of the learning experience (Gee, 2009; Groff, Howells, & Cranmer, 2010; Ke, 2009; Klopfer, Osterweil, & Salen, 2009). This encourages players to improve through repeated practice either by advancing within a game or replaying parts of a game. Failure with limited consequence, agency, and choice are seen as critical elements of a true gaming experience. That said, in the context of education where a game might become a required activity tied to real consequences, there could be a diminution in these key elements that may lead students to be less inclined to practice and realize some of the benefits of gaming.

Although a player’s actions may demonstrate learning within the game environment, less is known about whether such learning can be applied or transferred to a different context. For example, Gee (2005) describes how the game World of Warcraft reflects key 21st century skills such as individual specialization within cross-functional teams working collaboratively to meet goals. Although this type of specialization and collaboration is important within the game, it is still unclear how much these behaviors transfer outside of the game world. Of course, there are some situations in which you would not expect behavior from a game to transfer (e.g., jet skiing simulation games), and games cannot be adapted for every possible learning situation (Nagle, 2001).

The use of quiz games has also led to positive results in long-term student retention (ie ensuring they complete a course) by attracting higher student interest than traditional classroom approaches (Randel et al. 1992).For example, in training environments such as the Naval Training Systems Centre in Orlando, Florida, computer-based versions of board games such as Serious Pursuit were adapted to cater for service personnel

whose jobs required a pre-existing knowledge base for certain tasks. This prompted development of GameShell, a software program to house question and answer databases. When these games were used there

was better retention. This was attributed to more focused attention, because the students enjoyed the approach (Ricci 1994).

Learning progressions are frequently used in education. In traditional classroom settings, a student that does not master a concept could be left with a gap in their knowledge foundation that challenges later attempts to build to more complex concepts. In contrast, digital games inherently force the player to master a concept in order to advance (e.g., the double jump with a dash in mid-air to get across the pit of lava). Players are able to repeat the same scenario until they master this concept. The same philosophy could extend to the use of digital games in education. A student cannot, in essence, unlock Algebra until a prerequisite knowledge of previous skills has been mastered. This mastery-based learning, however, may require students to invest ample time in learning each skill before moving to the next.

These scenarios also imply that a student has some curricular choice and control over their learning. This sense of agency and autonomy for the learner is important. The most common error in online education activities is a failure to provide the learner with an appropriate level of agency. Agency refers to the learner’s ability to interact with the material and feelings of belongingness and socio-emotional support in the situation (Jalongo, 2007).

Simulation games have been used in schools to enhance children’s spatial abilities and general cognitive development, with both boys and girls performing equally well (de Lisi and Wolford 2002), while

Jayakanthan (2002) reports that versions of strategy games like Sim City have been used in schools to encourage learning in subjects such as geography. Simulation games have also been used in business

environments, for example in teaching administration skills. Off-the-shelf games simulations such as Doom II have been used in conjunction with free tools downloaded from the internet to provide cost-effective military training, for example where real-world environments or locations may be unavailable to troops.

The combination of interactivity within a familiar and yet novel situation, with clear and agreed aims for learning, proves very effective (Kirriemuir 2002). For example, a game with a job-relevant context has encouraged young adults who lacked the basic skills to perform their jobs to engage

in intensive instruction. The game served to guide their learning and to elicit performance via visual stimuli, motion sequence and audio feedback (Brownfield and Vik 1983).

However, learning does not just end with the game. Debriefing is critical to using games in education (Lederman & Fumitoshi, 1995), as it provides the connection between learning in the game and applying those skills to other contexts. Teachers can facilitate the transfer of skills by leading pre- and post-game discussions which connect the game with other things students are learning in class (Ash, 2011). Students can be encouraged to share different ways of approaching a problem. Based on a review of 17 studies focused on game design, Ke (2009) concluded that instructional support features are necessary in order for the lessons learned in computer games to transfer to other contexts. Video games can be used to create deeper learning experiences for students, but they do not provide the entire experience. Games work best when coupled with effective pedagogy (Squire, 2002). As such, Steinkueler & Chmiel (2006) suggest that games will not replace teachers and classrooms, but they might replace some textbooks and laboratories.

Simulation games have been found to be most effective in encouraging discovery learning where the system provides two kinds of instructional support: learner-requested background information and elaborate

system-initiated advice (Leutner 1993). However, the role of teacher mediation remains important, in explaining or augmenting the game (Lawry 1994; Kirriemuir 2002). For example, task cards were used with

games, requiring learners to describe their strategies and to provide tips to others, thereby stimulating reflection and writing skills (Kaptelinin and Cole 2001). Working with sections, rather than the whole game, may be more useful to particular learning objectives. This means the teacher must know the content behind the titles and understand controls, menus and skill levels of the game, and this requirement thus increases

teacher workload (Kirriemuir 2002).

# Software Requirements Specification of “The Adventure of a New Mother”

This chapter covers the requirements specification of our game “The Adventure of a New Mother”. It includes the specification of this documentation with general description, specific requirements, and analysis of models. It also includes changes management of this requirement specification in case of any change.

## Introduction

In this section, the documentation of this report is specified. It specifies the document convention, document scope and also provides a suggestion for the readers of the document.

### Purpose of this Chapter

This Software Requirements Specification (SRS) part is intended to give a complete overview of our Project the game “The Adventure of a New Mother” including the action flow, initial user interface and story therein. The SRS document details all features upon which we have currently decided with reference to the manner and importance of their implementation.

### Document Conventions

This document will freely interchange the pronoun “we” with the team’s acronym. As the development team is responsible for the SRS document, no ambiguity arises from its usage. There is a clear distinction, however, between the use of the words “player/gamer” and “character.” The “player” is the human being interacting with the game in the real world, while the “character” is the in-game player avatar being manipulated.

### Scope of this Document

This Software Requirements Specification (SRS) describes the functional and nonfunctional requirement for the project. As we said before the purpose of this research is to provide a virtual image for the combination of both structured and unstructured information of our project “The Adventure of a New Mother”.

Project “The Adventure of a New Mother” was conceived by the 2 of our team members as having an anticipated development cycle greater than the length of the semester. The team wishes to carry on the project until its completion. The game will continue to grow until we feel it satisfactory for open-source distribution.

## General Description

This section includes the perspective of our product and the system environment it requires. It specifies the QFD (Quality Function Deployment) of our game and also the User Story of it.

### Product and Business Perspective of the Game

Software product development is a paradigm shift from routine application maintenance and support in the software industry. Development a game/software product from scratch is a significant challenge for any organization. It requires considerable investments in terms of effort and cost and also confirms client involvement, knowledge about client market (example: Google play).

### System Environment

Gamer

Input Manager

(touchpad/game pad)

Script

(Compile)

Renders

(Display)

Fig 1: User Interection

Gamer can interact with system by giving input (press key to start game) to the system. System give those inputs to script, if any change occurs (if the value is changed) this object sends to renders to display the things (a character can change its place).

### Quality Function Deployment of “The Adventure of a New Mother”

Quality Function Deployment is a technique that translates the needs of the customer into technical requirements for software/game. It concentrates on maximizing customer satisfaction from the Software engineering process. With respect to our project the following requirements are identified by a QFD.

* + Normal Requirements.
  + Expected Requirements.
  + Exciting requirements.

**Normal Requirements**

Normal requirements consist of objectives and goals that are stated during the meeting with the actor/gamer/relevant people. Normal requirements of our project are: -

1. User friendly efficient and lucrative system.

2. Minimum maintenance cost (may be graphics definition).

3. Availability of expected requirements within the PC/mobile configuration.

4. Easy to operate.

5. They observe our game as this is build with professional manner.

6. The game with measured coding, professional thinking.

**Expected Requirements**

These requirements are implicit to the system and may be so fundamental that the actor/gamer/ relevant people does not explicitly state them. Their absence will be a cause for dissatisfaction.

1. Develop system within limited cost.

2. Maximum high definition.

3. Minimum hardware requirements which is relevant for this game.

4. Design whole system with efficient manner.

**Exciting requirements**

These requirements are for features that go beyond the customer's expectations and prove to be very satisfying when present:

1. We may provide some cheat codes.

2. Maximum high regulation with minimum hardware.

3. We may provide an international player rank list.

4. Easy to update.

### User Story of Our Game

“The Adventure of a new Mother” is a RPG educational game. It is a multi-platform game which is supported by web player, android phone, IOS and windows also.

After running the game, the UX view of the game will appear on the screen. The term UX means User Experience which is used to explain all aspects of a person’s experience with a system. However, then the gamer can directly select “Start” from the “Main Menu” and start playing the game. A “Story” is also provided with the game to understand the game objective. However, after starting a level the player will find helpful tips on the side of the screen and he/ she can follow it and enjoy the game.

The story behind the game is about a woman who recently became a new mother, now she will face some maternal and child care issues, she has to overcome those. The objective of the game is completing each task which will be regarding the child care with the help of a non-player character “Doctor Apa”, who will help the mother by giving important medical information. When a new information provided to the character, a new task regarding that information will be given and user have to perform those tasks. The timeline of the game is six months, so user will be provided only those months child care information, the game will end in sixth month.

## Specific Requirements

This section covers the project external requirements of our game and also indicates the user characteristics for this project.

* + 1. **External Interface Requirements of the Game**

We will disscuss the external interface requirements in this section

#### **User Interfaces**

Every game must have a menu so is can be user friendly enough and gamers can easily fulfill their need. Menu is also an important thing while creating the SRS document section. In this SRS document part; we have used the menu snapshots in the user manual part. These snapshots are based on the menu of the game.

#### **Hardware Interfaces**

“The Advancer of a new mother” is a mobile gaming application designed specifically for the Android platform and is functional on both mobile smart phones and tablets. Gaming application data is stored locally on the game engine elements. “The Advancer of a new mother” has been developed for Android developed Version and all subsequent releases. Now the Android platform is graphically adaptable with a 2-dimensional graphics library and a 3-dimensional graphics library based on OpenGL ES 2.0 specifications as well as hardware orientation, scaling, pixel format conversion and accelerated 3D graphics.

#### **Software Interface**

*“The Adventure of a new mother”* has been developed using a series of game development tools.

**Working tools and platform**

* **Javascript**
* **Jquery**
* **Enchantjs**
* **Illustrator CC2014**
* **Photoshop CS6**
* **PhotoScape**
* **Adobe Phonegap**
* **Sublime Text 3**

#### **User Characteristics for the System**

There is only one user at a time in this software and the user interacts with the game (system) in different manner. So, Gamer is the only one who communicates with the system through playing the game. And this gamer can be any person. The primary requirement is that, the gamer must read the playing procedure provided by us.

## Analysis Model of Our Game Project

This section describes the Software Requirements Specification of our project by analyzing the proper models of requirement engineering.

### Scenario Based Model

This Model depicts how the user interacts with the system and the specific sequence of activities that occur as the software is used.

#### **Use Case Scenario**

The following table summarizes the use cases of the system. We have created the use cases based on the UX view (mentioned in “User Story Part”) of the game. The swim lane diagram connects UX with background programming which are the two important views of a game SRS

|  |  |  |
| --- | --- | --- |
| **Level – 0** | **Level – 1** | **Level – 2** |
| “The Adventure Of a new Mother” | Play | New Game |
| Resume Game |
| Exit Game |
|  |
| About us |  |
| Information about the Developers |
|  |
| Option | Show Control |
| Change Sound/ Music Volume |
| Story | View Story |
| Quit | exit |

#### **Use Case Diagram with Use Case Description**

System

player

Fig 3: Level 1 for Game UX

Fig 2: Level 0 for Game UX

Player

Player

Fig 4: Level 2.1(Play) for Game UX

This Diagram of Level 2.1(Fig 3) leads us to the “Play” module of the use cases. These use case descriptions are given here –

**Play**

**Use case: New Game**

**Primary Actors:** Any one playing the game

**Goal in context:** To start a new game

**Precondition:**

1. System supports the game configuration
2. The file has been triggered to run and the game screen has appeared

**Triggers:** The player needs to start a new game

**Scenario:**

1. Go to the main menu of the game
2. Click new game button
3. New game is loaded on system

**Exception:** Game crushed

**Priority:** Essential, must be implemented

**When Available:** First increment

**Use case: Resume Game**

**Primary Actors:** Any one playing the game

**Goal in context:** To resume game from previous play

**Precondition:**

1. Game was played before
2. Game supports to have a checkpoint to start from

**Triggers:** Need to resume game

**Scenario:**

1. Go to the main menu of the game
2. Click the resume game button
3. Game is loaded from the last checkpoint

**Exception:**

1. Level cannot be loaded
2. Game crushed

**Priority:** Essential, must be implemented

**When Available:** First increment

**Use case: Exit Game**

**Primary Actors:** Any one playing the game

**Goal in context:** To exit from the game level

**Precondition:** A game level is being played

**Triggers:** Player needs to exit from the game level

**Scenario:**

1. Press game pause
2. When Pause Menu appears, click Return to Menu button
3. Game is exited and Title screen appears

**Priority:** Essential, must be implemented

**When Available:** First increment

Fig 5: Level 2.2(Options) for Game UX

Player

This Diagram of Level 2.2(Fig 4) connects with the “Option” module of the use cases. These use case descriptions are given here –

**Options**

**Use case: Show Controls**

**Primary Actors:**  Any one playing the game

**Goal in context:** To know the controls of playing the game

**Precondition:** Game provides control information

**Triggers:**  Player needs to know the controls to play the game

**Scenario:**

1. Go to the main menu
2. Click the Options button
3. When Option menu appears click the show control button
4. Game controls are being showed

**Exception:** No control information

**Priority:** Essential, must be implemented

**When Available:** First increment

**Use case: Change Sound/ Music Volume**

**Primary Actors:** Any one playing the game

**Goal in context:** To change the sound or music volume

**Precondition:** Player is allowed to change volume of game

**Triggers:** Player has a need to change volume of the game

**Scenario:**

1. Go to the main menu
2. Click on Options button
3. Click on Music/ Sound Slider and change the value
4. Music or Sound Volume is changed

**Exception:** System is in mute mode, cannot increase volume

**Priority:** Expected

**When Available:** Second increment

There is another module for “Quit” in Figure 1 which is the Level 1 of Use Case Diagram. The Use Case for it is given here –

1. **Quit**

**Use case: Quit**

**Primary Actors:** Any one playing the game

**Goal in context:** To Exit from the Game Process

**Precondition:** Player has entered in the game process

**Triggers:** Player needs to exit from the game

**Scenario:**

1. Go to the main menu
2. Click Quit button
3. Game is exited

**Exception:** Something went wrong. Cannot exit now.

**Priority:** Essential, must be implemented

**When Available:** First increment

#### **Activity Diagram**

**Go to Main Menu**

**Click New Game**

**Level-1 loaded**

Fig 6: Activity Diagram for “New Game” module of “Play” (Fig 3)

**Go to Main Menu**

**Click Resume Game**

**Last Played Level loaded**

Fig 7: Activity Diagram for “Resume Game” module of “Play” (Fig 3)

**Press Pause Game**

**Pause Menu Appears**

**Click Exit Game**

**Game Exited**

Fig 8: Activity Diagram for “Exit Game” module of “Play” (Fig 3)

**Go to Main Menu**

**Click Options**

**Option Menu Appears**

**Click Show Controls**

**Controls Showed**

Fig 9: Activity Diagram for “Show Controls” module of “Options” (Fig 4)

**Go to Main Menu**

**Click Options**

**Option Menu Appears**

**Set Value on Volume Slider**

**Volume Changed**

Fig 10: Activity Diagram for “Change Sound/ Music Volume” module of “Options” (Fig 4)

**Go to Main Menu**

**Click Story**

**Game Story Played**

Fig 11: Activity Diagram for “Story” module (Fig 1)

# 

**Go to Main Menu**

**Click Quit**

**Game Exited**

Fig 12: Activity Diagram for “Quit” module (Fig 1)

#### **Swimlane Diagram**

Background Programming

UX

**Go to Main Menu**

**Click New Game**

**Level-1 Loaded**

Fig 13: Swimlane Diagram for “New Game” module of “Play” (Fig 3)

Background Programming

UX

Background Programming

UX

**Go to Main Menu**

**Click Resume Game**

**Last Played Level loaded**

Fig 14: Swimlane Diagram for “Resume Game” module of “Play” (Fig 3)

**Pause Menu Appears**

**Game Exited**

Background Programming

UX

**Press Pause Game**

**Click Exit Game**

Fig 15: Swimlane Diagram for “Exit Game” module of “Play” (Fig 3)

Background Programming

UX

**Go to Main Menu**

**Options Menu Appears**

**Click Options**

**Click Show Controls**

**Controls Showed**

Fig 16: Swimlane Diagram for “Show Controls” module of “Options” (Fig 4)

Background Programming

UX

**Go to Main Menu**

**Options Menu Appears**

**Click Options**

**Set Value on Volume Slider**

**Volume Changed**

Fig 17: Swimlane Diagram for “Change Sound/ Music Volume” module of “Options” (Fig 4)

Background Programming

UX

**Go to Main Menu**

**Click Story**

**Game Story Played**

Fig 18: Swimlane Diagram for “Story” module (Fig 1)

Background Programming

UX

**Go to Main Menu**

**Click Quit**

**Game Exited**

Fig 19: Swimlane Diagram for “Quit” module (Fig 1)

### Data Model

I have used client-Side storage called local storage. Before HTML5, application data had to be stored in cookies, included in every server request. Local storage is more secure, and large amounts of data can be stored locally, without affecting website performance.

Unlike cookies, the storage limit is far larger (at least 5MB) and information is never transferred to the server.

Local storage is per origin (per domain and protocol). All pages, from one origin, can store and access the same data.

Local storage is very easy to use. This is a key-value model database.

A Local storage, or key-value database is a simple database that uses an associative array (think of a map or dictionary) as the fundamental data model where each key is associated with one and only one value in a collection. This relationship is referred to as a key-value pair.

In each key-value pair the key is represented by an arbitrary string such as a filename, URI or hash. The value can be any kind of data like an image, user preference file or document. The value is stored as a blob requiring no upfront data modeling or schema definition.

The storage of the value as a blob removes the need to index the data to improve performance. However, you cannot filter or control what’s returned from a request based on the value because the value is opaque.

In general, local storage stores have no query. They provide a way to store, retrieve and update data using simple get, put and delete commands; This are the 3 functions



1. localStorage.setItem("key", "value");

2. localStorage.getItem("key");

3. localStorage.removeItem("key");

### Behavioral Model

The Behavioural indicates how software will respond to external events or stimuli. There are two ways to show these responses. One is state diagram and the other is sequence. Usually state diagram can be made in two ways, one is creating a state diagram for each class and the other is to create a state diagram for the whole system. As we don’t have any class, for this is not an object-oriented game, we have followed the later one. We used the modules of the use case scenario to create the state diagram. And to lessen complexity we have divided the state diagram into two diagrams. On the other hand, for the sequence diagram, we have created separate a sequence diagram for all the use cases when necessary.

#### **State Diagram**

from Level Select, Level Complete, and In Game Menus in Play Level

**Splash Screen**

Fig: top level state diagram

**Checking**

Do: isclicked

**Main Menu**

**Open Game**

**Idle**

**“Options”**

**“Quit”**

**“Play”**

To idle

**Close Game**

**Play Menu**

**Options Menu**

**“Select Level”**

To level select menu

**“Sound/Music”**

**“Controls”**

**“Return”**

Fig 20: top level state diagram



Fig 21: top level state diagram

# Implementation of the Proposed System

This chapter covers the project design phases, the system features and also the implementation of the features..

## Product Design Terms

For every enterprise product two key terms of design is very important. They are:

* UX (User Experience)
* Backend Programming

### User Experience (UX)

User experience design (UXD or UED) is any aspects of a user's experience with a given system, including the interface, graphics, industrial design, physical interaction, and the manual in most cases,

User Experience Design fully encompasses traditional Human-Computer Interaction (HCI) design, and extends it by addressing all aspects of a product or service as perceived by users. UX stands for mainly relevant access of usability, accessibility and HCI.

UX defines user experience as “a person’s perceptions and responses that result from the use or anticipated use of a product, system or service”.

### Backend Programming

#### Software and framework

I have used the following language and software to build this application

1. HTML5
2. CSS3
3. JavaScript
4. Sublime Text
5. Mozilla Firefox

**HTML5**

HTML5 is a markup language used for structuring and presenting content on the World Wide Web. It is the fifth and current version of the HTML standard. We took the advantage of HTML5 Canvas. Which is basically a drawing tool.

**CSS3**

CSS is the language for describing the presentation of Web pages, including colors, layout, and fonts. It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers. CSS is independent of HTML and can be used with any XML-based markup language.

**JavaScript**

JavaScript is a cross-platform, object-oriented scripting language. It is a small and lightweight language. Inside a host environment (for example, a web browser), JavaScript can be connected to the objects of its environment to provide programmatic control over them.

JavaScript contains a standard library of objects, such as Array, Date, and Math, and a core set of language elements such as operators, control structures, and statements. Core JavaScript can be extended for a variety of purposes by supplementing it with additional objects;

**Sublime Text**

Sublime Text is a proprietary cross-platform source code editor with a Python application programming interface (API). It natively supports many programming languages and markup languages, and functions can be added by users with plugins, typically community-built and maintained under free-software licenses.

**Mozilla Firefox**

Mozilla Firefox (or simply Firefox) is a free and open-source web browser developed by the Mozilla Foundation and its subsidiary the Mozilla Corporation. Firefox is available for Windows, macOS and Linux operating systems,

**Features**

I decided to prioritize the unique game mechanics over general game features like scenes, because I find that the most exciting about this project:

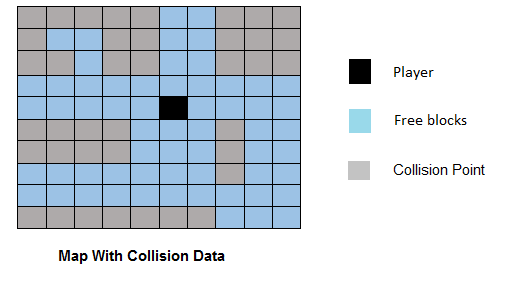
**Map creation**

As we have said before that we have used HTML5 canvas, where we can draw anything. We took our sample image and divide it into 32/32 matrix. Every block represents one blocks represent 32pixel data.

Similarly, we have added collision with another 32/32 matrix 0 for no collision and 1 for collision.

**User Movment**

As you can see in the picture the black block represent our player. And it can only move thorough the blue space. The gray pont’s are coillision point where map prevend out player to pass thorough.



**Change between scenes**

When any scenes change our JavaScript code made the adjustment to the map matrix replace the pixels and draw the image into the canvas

**Talking system**

When the game loaded to the client device all the assets are been downloaded. We create an audio object d then load the audio file into it and call the play() method depending on the predefined condition’s.

var audio = new Audio('audio\_file.mp3');

audio.play();

**Animation Creation**

When it’s comes to animation’s creation we have use the method frame repetitions. To make a perfect animation we loaded the image into and array and divide that into 5 sections and use this frame to represent each state of the battery. This technic is widely known as Sprite image.



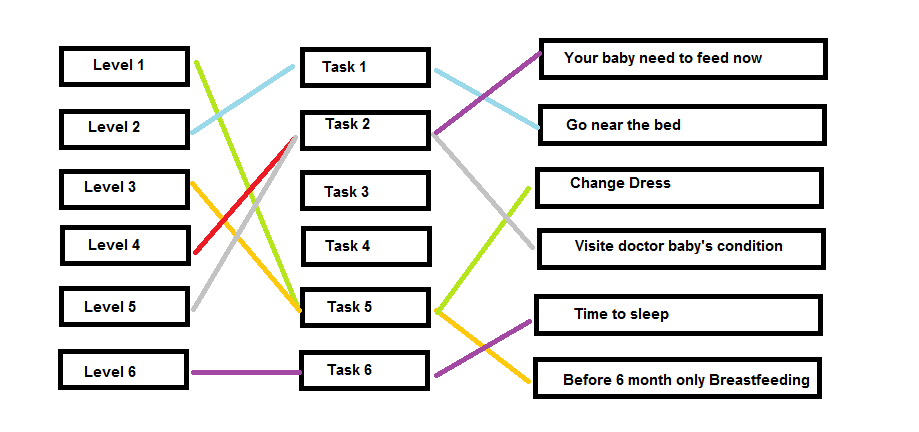
**Adding supporting character with main character**

We have different types of character, like mother with her baby and without her baby. Depending on the condition we undated our database that mother image has been charged. And when she inters to a new map we check our database to find if there is any change if so change then Sprite. First picture demonstrates our character with our child and 2nd picture we have removed the child from bed and charged the Mon current image.



**Navigation Bar**

We have created a dedicated navigation bar to pop all the key information and guideline that the player needs to follow. Depending on the current level and task massage has been ben loaded from the database into the navigation bar.



**Baby’s health Dependency**

We represent baby’s health with 3 different battery.

1. Battery, one depends on bath
2. Battery two depends on food
3. Battery three depends on cloth

After every N1, N2, N3 second one charge decrease from battery one, two and three.

If any one of its charge become less than 2 bar then we notify the player that his/her baby needs to take bath or food or cloth. When its come’s to priority

N2 > N3 > N4

The following graph shows the food Dependency happened

## Assumptions and Dependencies

The final destination of our game's operation will be the Android mobile device. However, Unity will be responsible for both the construction of the game and its integration within the Android framework

### Construction of the Game

Enchantjs includes many built-in components which will expedite the process of game development immensely. These include:

o Physics Engine

o Collision Detection and Handling

o Input Recognition

o Object Creation and Transform Manipulation (position and rotation of game objects)

o Scene Integration (transition of one level to the next)

o Model Attachment (representing game objects with 3D models from external programs)

### Integration with Android

Adobe Phonegap build settings simplify the process of transferring our game to the Android mobile device. After completing the project, or during any intermediary step for testing, we can select drop the zip file of the game into Phonegape to build the project, and upload it to one of our own devices.

## Key Resource Requirements of the Project.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Major Project activities** | **Skill/Expertise Required** | **Internal Resources** | **External Resources** | **Issues/Constraints** |
| Level Design | Ability to translate aspects of the story into playable levels | Me, myself made the decision about game levels | Ideas from existing games (Ex. Stealth) | Conflicting ideas per level |
| Physics Engine | Knowledge of functions available in Unity and the ability to change them as needed | Me and Soykoth worked on enchantjs game engine | enchantjs game engine | Ability to angle interactive portions of levels |
| Graphics Design | Knowledge of graphical modeling and implementation | Me worked for creating 2d Arts | 2d models design using Adobe illustrator and Photoshop | Visibility of the details on the 2d arts |
| Music Development/ Implementation | Ability to incorporate sound clips smoothly into the game | - | Sound clips from the Internet | Ability to play sound clips at precious times during game play |
| Level Implementation | Familiarity with scripting language of game engine | All members have some knowledge about scripting language | Open source arts from the internet | Level size dependent on hardware configuration |
| Documentation | Knowledge about SRS and Formal Report Writing | Me worked more on Reporting | Idea from Existing Reports | Game Reports are Different from Conventional ones |

## Implementation Tools Required

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Product of** | **Tool** | **Usage** | **Work exp.** |
| C:\Users\Alam\AppData\Local\Microsoft\Windows\INetCache\Content.Word\enchant.png | UEI Corporation / enchant.js Inc. | enchantJS | Game Engine | Backend activity |
| F:\Users\Nadia\Desktop\images.jpg | Adobe | Photoshop | Picture Edit | 2d art textures |
| C:\Users\Alam\AppData\Local\Microsoft\Windows\INetCache\Content.Word\index.png | Illustrator | Graphics Design and Animation | Create 2d art |
| C:\Users\Alam\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Sublime_Text_Logo.png | |  |  | | --- | --- | |  | Jon Skinner | | Sublime Text 3 | Text editor | Code |

## Implementation Code Example

We have a number of features that affect the environment of the game, to take care the baby we need to feed him, give a bath and change the dress, depending on the time the bar for each item decrease. To keep the complete bar user, have to keep doing those three works depending on time. We also created a proper fancy notification bar for showing the user which task he/she have to perform. So, we are presenting the code for it (using language javascript) as an example of implementation codes.

if ((player.x==168||player.x==152||player.x==184)&&(player.y==320||player.y==336||player.y==352) && outCurrentRoom=="img/bath\_room.png" && babyconfirmSceneCounter==0 && playerImage==2)

{

babyconfirmSceneCounter++;

var confirmScene = new ConfirmScene('Do you want bath your baby', 'Yes', 'No');

confirmScene.oncancel = function() {

babyconfirmSceneCounter=0;

game.rootScene.removeChild(confirmScene);

};

confirmScene.onaccept = function() {

window.location.href = 'bath.html';

};

game.rootScene.addChild(confirmScene);

var soap = new Sprite(81, 48);

soap.image = game.assets['img/sop.png'];

soap.x = 250;

soap.y = 325;

game.rootScene.addChild(soap);

soap.addEventListener('touchend', function() {

if(soopCounter==0){

soopCounter++;

bathBatteryCounter++;

bathBatteryFrameUndate(bathBatteryCounter);

if(++batteryOneCharge<7){

updatelocalStorage('batteryOne',batteryOneCharge);

}

}

var sopani = new Sprite(232, 348);

sopani.image = game.assets["img/soapf.png"];

sopani.x = 40;

sopani.y = 145;

sopani.scale(.8);

sopani.frame = 0;

game.rootScene.addChild(sopani);

sopani.addEventListener("enterframe", function() {

this.frame = [

0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,

3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,

4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,

5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,

6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,

7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7,

8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,

];

sopani.tl.setTimeBased();

sopani.tl.cue({

300: function() {

this.scene.removeChild(this);

}

});

});

sopani.tl.delay(1000).then(function() {

sopani2.image = game.assets["img/sp.png"];

sopani2.x = 40;

sopani2.y = 145;

sopani2.scale(.8);

sopani2.frame = 8;

game.rootScene.addChild(sopani2);

})

});

More code : Appnedix 1

# Testing of “The Adventure of new Mother”

This chapter includes some test cases for the game to check if the game works properly in various situations. We are giving four test examples for four different situations here.

## Test Case 1

|  |  |  |
| --- | --- | --- |
| **Test Case** | **:** | This test will check if the animation is working correctly. |
| **Test Procedure** | **:** | Import a character sprite sheet in enchantjs . Place character on the scene. Run the game. |
| **Expected Result** | **:** | Animation works perfectly in the environment. |
| **Actual Result** | **:** | Animation is not working. |
| **Comment** | **:** | Need to check character configuration on inspector window. The appropriate animation was not selected. Select it. |
| **Conditional Test** | **:** | Again, run scene. |
| **Expected Result** | **:** | Animation is working now. |
| **Actual Result** | **:** | Yes, it is working. |
| **Accuracy** | **:** | Accuracy depends on hardware configuration. |

## Test Case 2

|  |  |  |
| --- | --- | --- |
| **Test Case** | **:** | This test will check if the interaction between objects is working correctly. |
| **Test Procedure** | **:** | Add scripts of interaction in the objects that we want to interact with each other.Run scene. |
| **Expected Result** | **:** | Objects are interacting. |
| **Actual Result** | **:** | Run time exception |
| **Comment** | **:** | Need to add checking in the scripts for the objects that have a particular script. |
| **Conditional Test** | **:** | Run scene. |
| **Expected Result** | **:** | Interaction is ok now. |
| **Actual Result** | **:** | Interaction is ok now. |
| **Accuracy** | **:** | Perfectly accurate. |

## Test Case 3

|  |  |  |
| --- | --- | --- |
| **Test Case** | **:** | This test will check if the dialogue box is working. |
| **Test Procedure** | **:** | Add dialogue box in the scene. Run scene. |
| **Expected Result** | **:** | Dialogue box appears in the correct dimension. |
| **Actual Result** | **:** | Working perfectly |
| **Comment** | **:** | Tips and dialogues are working as expected. |

## Test Case 4

|  |  |  |
| --- | --- | --- |
| **Test Case** | **:** | This test will check if the automatic map changes when character is around |
| **Test Procedure** | **:** | Configure door. Run scene. |
| **Expected Result** | **:** | Map changes depending on character position. |
| **Actual Result** | **:** | Working perfectly |
| **Comment** | **:** | Automatic door can recognize character and opens. |

# Discussion and Result

My game “The Adventure of a new Mother”, aim to teach users about maternal issues and childcare. My game has great numbers of interactive capabilities, with the conversation system and navigation bar, this game can actually interact with the user and lead them what they should do and what they should not. This capability helps to provide the information to the user in an effective way. In the game, use must learn the information because base on each information the next mission is designed. The game design is so much responsive that user will never get bored. While the current awareness process is not so effective, this game can actually change that in a fun and easy way.

Charlton, Beryl; Williams, Randy Lee; McLaughlin, T. F. evaluated the effects of educational games on the performance of eight elementary school students with learning disabilities. The effects of educational games were evaluated in a multiple baseline design across students. The results indicated that each student improved their performance on reading when educational games were in effect. These differences were also educationally significant, *International Journal of Special Education*, v20 n2 p66-72 2005.

Marina Papastergiou’s study was another great example, her study on “The effects of modern mathematics computer games on mathematics achievement and class motivation” support our claim.

The aim of that study was to assess the learning effectiveness and motivational appeal of a computer game for learning computer memory concepts, which was designed according to the curricular objectives and the subject matter of the Greek high school Computer Science (CS) curriculum, as compared to a similar application, encompassing identical learning objectives and content but lacking the gaming aspect. The study also investigated potential gender differences in the game’s learning effectiveness and motivational appeal. The sample was 88 students, who were randomly assigned to two groups, one of which used the gaming application (Group A, N = 47) and the other one the non-gaming one (Group B, N = 41). A Computer Memory Knowledge Test (CMKT) was used as the pretest and posttest. Students were also observed during the interventions. Furthermore, after the interventions, students’ views on the application they had used were elicited through a feedback questionnaire. Data analyses showed that the gaming approach was both more effective in promoting students’ knowledge of computer memory concepts and more motivational than the non-gaming approach. Despite boys’ greater involvement with, liking of and experience in computer gaming, and their greater initial computer memory knowledge, the learning gains that boys and girls achieved through the use of the game did not differ significantly, and the game was found to be equally motivational for boys and girls.

Another study by Mansureh Kebritchi, Atsusi Hirumi, Haiyan Bai on “The effects of modern mathematics computer games on mathematics achievement and class motivation”, Computers & Education 55 (2010) 427e443.This study actually have the prove that, my kind of game can actually effective,

The purpose of that study was to examine the effects of a series of modern mathematics computer games, DimensionM, on mathematics achievement and motivation of high school students. In addition, the role of prior mathematics knowledge, computer skills, and English language skills of the participants on their mathematics achievement and motivation when they played the games were investigated. The total

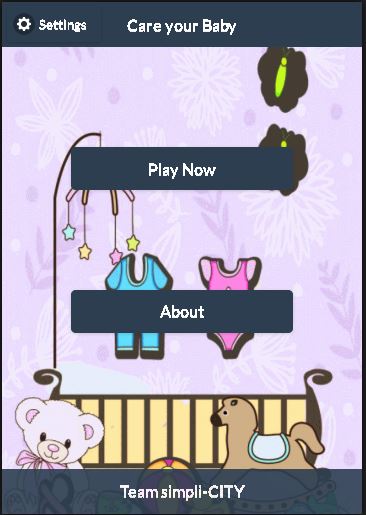
of 193 and 10 teachers from an urban high school in the southeast of the United States of America were participated in this study. The MANCOVA tests were conducted to analyze the data. In addition, interviews were conducted to cross validate the quantitative results. The DimensionM games had a significant positive effect on students' mathematics achievement in the public high school setting.

Students who played the mathematics computer games scored significantly higher on the district-wide math benchmark exam than students who did not play the games.

These three studies showing the effectiveness of my developed game. The design of the game is responsive and communicative, it’s easy to understand and have authenticate information for the user. The mechanism of this game can actually improve the awareness program.

# Some Snapshots

Here is some snapshot of the mobile view of the game ,



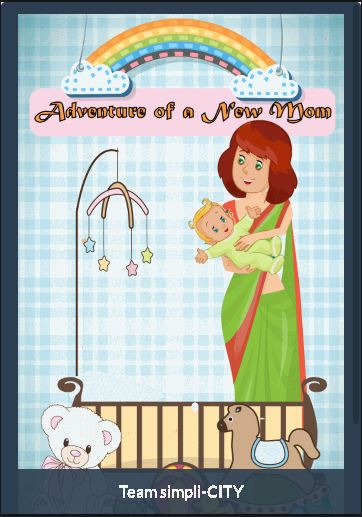


Fig-22: Opening Scene 1 Fig-23: Opening Scene 2



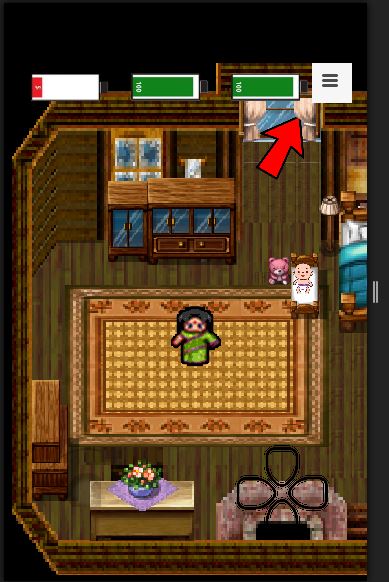


Fig-24: Bedroom Scene Fig-25: Dinning Room Scene





Fig-26: Bathroom Scene Fig-27:kitchen Scene





Fig-28: Outdoor Scene 1 Fig-29: Outdoor Scene 2





Fig-30: hospital Scene Fig-31: Bath Scene

# Conclusion

A software project means a lot of experience. In this section, we summarize the experience gained by project team during development of “The Adventure of a new mother”.

## Summary

During this study, we know much more about game engines. How it works? The properties, objects and others. We know how a model is constructed and how it is animated. The main thing is that as a software engineer, skill and expertise to create a SRS document and an overall software product report is now better than before-Operation between group members. Develop communication skills. Growing creative thinking and imagination capability.

## Evaluation

We learned a lot through this project. This project has sharpened our concept of Game engine, animation and the software-hardware interface. We learned a lot about different documentation. The piece of software we developed is intended to serve the gamers of the world. The success of this project may give pleasure to billions of game lovers among the universe. This project not only tested our technical skills but also our temperament. There were times that we almost lost hope but we recovered through constant concentration and hard work.

## Future Work

“The Adventure of a new Mother” is a game which provide information about maternal issues and child care in a fun and integrative way. However, it need lots of improvements like

* Level Extension
* Improve Graphical Representation
* Introduce new game features
* Introduce new environment and scenes
* Take user response through website and produce web rank list

Apart from these there was time limitation, this kind of project need more research,support and time.

References

[1] BayHinitz, April K., Peterson, Robert F., and Quilitch, H. Robert. “Cooperative games: a way to modify aggressive and cooperative behaviors in young children.” Journal of Applied Behavior Analysis27.3 (1994): 433446.

[2] Blackmon, Wayne D. “Dungeons and Dragons: the use of a fantasy game in the psychotherapeutic treatment of a young adult.” Journal of Psychotherapy 48.4 (1994): 624632.28 Nov. 2008

[3] McClain, Rachel. “Imagine That! Developing Fantasy RolePlay in the Language Arts Classroom.” The New York Times Learning Network. 8 Dec. 2000

[4] Phillips, Brian David. “Roleplaying Games in the English as a Foreign Language Classroom.” Proceedings of the Tenth National Conference on English Teaching and Learning, May 15, 1994. Taipei, China: Crane

Publishing Ltd., 19944

[5] Tietel, Jay. “Wanna Play?” Psychology Today Aug. 1998. 28 Nov. 2008

[6] Lafferty, Mur. “D&D Therapy.” The Escapist 29 Mar. 2006. 4 Dec. 2008

[7] Hughes, John. “Therapy is Fantasy: Roleplaying, Healing, and the Construction of Symbolic Order.” Unpublished Essay Australian NationalUniversity, 1988

[8] Haddad, Sam. “The Results of Playing The Game of Social Studies Knowledge in an Eighth Grade Classroom.” Diss. Wayne State University of Michigan, 2003.

[9] Cruz, Laura. “The Great War in the Classroom.” Academic Exchange Quarterly

[10] Yu-Hong Lin,"Pokémon: Game Play as Multi-Subject Learning Experience”, The First IEEE International Workshop on Digital Game and Intelligent Toy Enhanced Learning (DIGITEL'07)

[11] Barnes, T., Richter, H., Powell, E., Chaffin, A., and Godwin,A. 2007. Game2Learn: building CS1 learning games for retention. SIGCSE Bull. 39, 3 (Jun. 2007), 121-125.47

[12] Bayliss, J. D. and Strout, S. 2006. Games as a "flavor" of CS1. SIGCSE Bull. 38, 1 (Mar. 2006), 500-504.

[13] Beaubouef, T. and Mason, J. 2005. Why the high attrition rate for computer science students: some thoughts and

observations. SIGCSE Bull. 37, 2 (Jun. 2005), 103-106.

[14] Brandel, M. 2007. 12 IT skills employers cannot say no to.ComputerWorld, Jul. 2007.

[15] Burd, B., Goulden, J., Ladd, B., Rogers, M., and Stewart, K.2007. Computer games in the classroom, or, how to get perfect attendance, even at 8 AM. SIGCSE Bull. 39, 1 (Mar.2007), 496-496.

[16] Chamillard, A. T. 2006. Introductory game creation: no programming required. SIGCSE Bull. 38, 1 (Mar. 2006),515-519.

[17] Chao, C. 2006. An Investigation of Learning Style Differences and Attitudes toward Digital Game-based

Learning among Mobile Users. In Proceedings of the 4th IEEE International Workshop on Wireless, Mobile and Ubiquitous Technology in Education (Nov. 16-17, 2006).WMTE. IEEE CS, Washington, DC, 29-31.

[18] Claypool, K. and Claypool, M. 2005. Teaching software engineering through game design. SIGCSE Bull. 37, 3(Sep.2005), 123-127.

[19] Cliburn, D. C. and Miller, S. 2008. Games, stories, or something more traditional: the types of assignments college students prefer. SIGCSE Bull. 40, 1 (Feb. 2008), 138-142.

[20] Clua, E., Feijó, B., Rocca, J., Schwartz, J., das Graças, M.,Perlin, K., Tori, R., and Barnes, T. 2006. Game and

interactivity in computer science education. In ACM SIGGRAPH 2006 Educators Program (Boston, MA, Jul. 30 -

Aug. 3, 2006). SIGGRAPH '06. ACM, New York, NY.

[21] Charlton, Beryl; Williams, Randy Lee; McLaughlin, T. F. "Educational Games: A Technique to Accelerate the Acquisition of Reading Skills of Children with Learning Disabilities"

International Journal of Special Education, v20 n2 p66-72 2005

[22] Marina Papastergiou “The effects of modern mathematics computer games on mathematics achievement and class motivation”,

Computers & Education,Volume 52, Issue 1, January 2009, Pages 1-12

[23] Mansureh Kebritchi, Atsusi Hirumi, Haiyan Bai,“The effects of modern mathematics computer games on mathematics achievement and class motivation”,

Computers & Education 55 (2010) 427e443

Appendix 1

**Code**

**main.js**

enchant();

var player = new Sprite(64, 64);

outCurrentRoom='img/'+localStorage.getItem('1');

var arrowLevel1 = new Sprite(320,256);

/\*alert(outCurrentRoom);\*/

window.onload = function() {

var game = new Game(360, 512);

game.fps = 15;

game.scale = 1;

game.preload(

'img/babyFeedIcon.png',

'img/momwithchild.png',

'img/n.png',

'img/battery1.png',

'img/battery2.png',

'img/battery3.png',

'img/battery4.png',

'img/battery5.png',

'img/battery6.png',

'img/battery7.png',

'img/arrowright.png',

'img/chara0.png',

'img/taskbar.png',

outCurrentRoom,

'img/doctor.png',

'img/apad.png',

'img/baby.gif'

);

//Non player carecter(NPC)

var Npc = enchant.Class.create(enchant.Sprite, {

initialize: function(x, y, no, map) {

enchant.Sprite.call(this, 64, 64);

this.x = x;

this.y = y;

this.kind = no; // NPCの種類

// NPCの種類ごとにスプライト画像から使う分だけ切り取って、いったんサーフェイスとして書き出す

var image = new Surface(96, 128);

switch (this.kind) {

case 0:

image.draw(game.assets['img/chara0.png'], 0, 0, 96, 128, 0, 0, 96, 128);

break;

case 1:

image.draw(game.assets['img/chara0.png'], 96, 0, 96, 128, 0, 0, 96, 128);

break;

case 2:

image.draw(game.assets['img/chara0.png'], 192, 0, 96, 128, 0, 0, 96, 128);

break;

}

this.image = image;

this.isMoving = false;

this.noMoving = false;

this.direction = 0;

this.walk = 0;

this.frame = 0;

}

});

var NpcMove = enchant.Class.create(enchant.Sprite, {

initialize: function(x, y, no, map) {

enchant.Sprite.call(this, 64, 64);

this.x = x;

this.y = y;

this.kind = no; // NPCの種類

// NPCの種類ごとにスプライト画像から使う分だけ切り取って、いったんサーフェイスとして書き出す

var image = new Surface(96, 128);

switch (this.kind) {

case 0:

image.draw(game.assets['chara5.png'], 0, 0, 96, 128, 0, 0, 96, 128);

break;

case 1:

image.draw(game.assets['chara5.png'], 96, 0, 96, 128, 0, 0, 96, 128);

break;

case 2:

image.draw(game.assets['chara5.png'], 192, 0, 96, 128, 0, 0, 96, 128);

break;

}

this.image = image;

this.isMoving = false;

this.noMoving = false;

this.direction = 0;

this.walk = 0;

this.frame = 0;

this.addEventListener('enterframe', function() {

if (this.noMoving) return;

this.frame = this.direction \* 3 + this.walk;

// 移動中の処理

if (this.isMoving) {

this.moveBy(this.vx, this.vy);

this.walk = game.frame % 3;

if ((this.vx && (this.x - 8) % 16 === 0) || (this.vy && this.y % 16 === 0)) {

this.isMoving = false;

this.walk = 0;

}

} else {

// 移動中でないときは、ランダムに移動方向を設定する

this.vx = this.vy = 0;

this.mov = rand(4);

switch (this.mov) {

case 0:

this.direction = 0;

this.vy = 4;

break;

case 1:

this.direction = 1;

this.vx = -4;

break;

case 2:

this.direction = 2;

this.vx = 4;

break;

case 3:

this.direction = 3;

this.vy = -4;

break;

}

// 移動先が決まったら

if (this.vx || this.vy) {

// 移動先の座標を求める

var x = this.x + (this.vx ? this.vx / Math.abs(this.vx) \* 16 : 0) + 16;

var y = this.y + (this.vy ? this.vy / Math.abs(this.vy) \* 16 : 0) + 16;

// その座標が移動可能な場所なら

if (0 <= x && x < map.width && 0 <= y && y < map.height && !map.hitTest(x, y)) {

// 移動フラグを「true」にする

this.isMoving = true;

}

}

}

});

}

});

//end NPC

game.onload = function() {

var map = new Map(16, 16);

map.image = game.assets[outCurrentRoom];

map.loadData(

[

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31],

[32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63],

[64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95],

[96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127],

[128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159],

[160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191],

[192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223],

[224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255],

[256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287],

[288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319],

[320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351],

[352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383],

[384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415],

[416, 417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447],

[448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479],

[480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494, 495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511],

[512, 513, 514, 515, 516, 517, 518, 519, 520, 521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533, 534, 535, 536, 537, 538, 539, 540, 541, 542, 543],

[544, 545, 546, 547, 548, 549, 550, 551, 552, 553, 554, 555, 556, 557, 558, 559, 560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 570, 571, 572, 573, 574, 575],

[576, 577, 578, 579, 580, 581, 582, 583, 584, 585, 586, 587, 588, 589, 590, 591, 592, 593, 594, 595, 596, 597, 598, 599, 600, 601, 602, 603, 604, 605, 606, 607],

[608, 609, 610, 611, 612, 613, 614, 615, 616, 617, 618, 619, 620, 621, 622, 623, 624, 625, 626, 627, 628, 629, 630, 631, 632, 633, 634, 635, 636, 637, 638, 639],

[640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651, 652, 653, 654, 655, 656, 657, 658, 659, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669, 670, 671],

[672, 673, 674, 675, 676, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702, 703],

[704, 705, 706, 707, 708, 709, 710, 711, 712, 713, 714, 715, 716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 730, 731, 732, 733, 734, 735],

[736, 737, 738, 739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 762, 763, 764, 765, 766, 767],

[768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783, 784, 785, 786, 787, 788, 789, 790, 791, 792, 793, 794, 795, 796, 797, 798, 799],

[800, 801, 802, 803, 804, 805, 806, 807, 808, 809, 810, 811, 812, 813, 814, 815, 816, 817, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 830, 831],

[832, 833, 834, 835, 836, 837, 838, 839, 840, 841, 842, 843, 844, 845, 846, 847, 848, 849, 850, 851, 852, 853, 854, 855, 856, 857, 858, 859, 860, 861, 862, 863],

[864, 865, 866, 867, 868, 869, 870, 871, 872, 873, 874, 875, 876, 877, 878, 879, 880, 881, 882, 883, 884, 885, 886, 887, 888, 889, 890, 891, 892, 893, 894, 895],

[896, 897, 898, 899, 900, 901, 902, 903, 904, 905, 906, 907, 908, 909, 910, 911, 912, 913, 914, 915, 916, 917, 918, 919, 920, 921, 922, 923, 924, 925, 926, 927],

[928, 929, 930, 931, 932, 933, 934, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944, 945, 946, 947, 948, 949, 950, 951, 952, 953, 954, 955, 956, 957, 958, 959],

[960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 970, 971, 972, 973, 974, 975, 976, 977, 978, 979, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991],

[992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023],

],[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],

[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],

[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],

[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],

[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],

[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],

[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],

[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],

[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],

[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],

[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],

[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],

[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],

[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],

[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],

[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],

[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],

[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],

[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],

[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],

[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],

[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],

[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],

[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],

[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],

[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],

[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],

[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],

[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],

[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],

[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],

[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],

);

map.collisionData = [

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

];

if(outCurrentRoom=="img/bedRoom.png")map.collisionData = badRoomCollisionData;

else if(outCurrentRoom=="img/drawing\_room.png")map.collisionData = drawingRoomCollisionData;

else if(outCurrentRoom=="img/kitchen\_room.png")map.collisionData = kitchenRoomCollisionData;

else if(outCurrentRoom=="img/bath\_room.png")map.collisionData = bathRoomCollisionData;

player.x = +localStorage.getItem('playersPositionX');

player.y = +localStorage.getItem('playersPositionY');

/\* var image = new Surface(192, 256);

image.draw(game.assets['img/chara0.png'], 0, 0, 192, 256, 0, 0, 192, 256);\*/

if (playerImage==1) {

player.image = game.assets['img/chara0.png'];

} else {

player.image = game.assets['img/momwithchild.png'];

}

player.isMoving = false;

player.direction = 0;

player.walk = 1;

player.addEventListener('enterframe', function() {

this.frame = this.direction \* 3 + this.walk;

if (this.isMoving) {

this.moveBy(this.vx, this.vy);

// i am in badroom

if ((player.x==232 || player.x==248) && player.y==416 && outCurrentRoom=="img/bedRoom.png"){

updatelocalStorage(1,"drawing\_room.png");

updatelocalStorage("playersPositionX",120);

updatelocalStorage("playersPositionY",80);

location.reload();

}

if ( (player.x==232 ) && (player.y==192 || player.y==176) && outCurrentRoom=="img/bedRoom.png" && babyconfirmSceneCounter==0){

babyconfirmSceneCounter++;

if(+localStorage.getItem('playerImage')==1){

var babyconfirmScene = new ConfirmScene('Do you want to take your baby', 'Yes', 'No');

game.rootScene.addChild(babyconfirmScene);

babyconfirmScene.oncancel = function() {

babyconfirmSceneCounter=0;

game.rootScene.removeChild(babyconfirmScene);

};

babyconfirmScene.onaccept = function() {

playerImage=2;

player.image = game.assets['img/momwithchild.png'];

baby.visible=false;

babyconfirmSceneCounter=0;

updatelocalStorage("playerImage",2);

game.rootScene.removeChild(babyconfirmScene);

};

}

else{

var babyconfirmScene = new ConfirmScene('Do you want put your baby', 'Yes', 'No');

game.rootScene.addChild(babyconfirmScene);

babyconfirmScene.oncancel = function() {

babyconfirmSceneCounter=0;

game.rootScene.removeChild(babyconfirmScene);

};

babyconfirmScene.onaccept = function() {

playerImage=1;

player.image = game.assets['img/chara0.png'];

baby.visible=true;

babyconfirmSceneCounter=0;

updatelocalStorage("playerImage",1);

game.rootScene.removeChild(babyconfirmScene);

};

}

}

//start i am in kitchen\_room

//going to drawing\_room

if (player.x==24 && (player.y==256 || player.y==272) && outCurrentRoom=="img/kitchen\_room.png") {

updatelocalStorage(1,"drawing\_room.png");

updatelocalStorage("playersPositionX",344);

updatelocalStorage("playersPositionY",224);

location.reload();

}

//end

//start i am in drawing Room

//going to Bath Room

if (player.x==88 && (player.y==80||player.y==96||player.y==112) && outCurrentRoom=="img/drawing\_room.png") {

updatelocalStorage(1,"bath\_room.png");

updatelocalStorage("playersPositionX",344);

updatelocalStorage("playersPositionY",204);

location.reload();

}

if ( (player.x==134|| player.x==120) && player.y==352 && outCurrentRoom=="img/drawing\_room.png") {

updatelocalStorage("playersPositionX",200);

updatelocalStorage("playersPositionY",96);

window.location.href = 'mapoutdoor1.html';

}

//going to kitchen\_room

if (player.x==376 && (player.y==224||player.y==208) && outCurrentRoom=="img/drawing\_room.png") {

updatelocalStorage(1,"kitchen\_room.png");

updatelocalStorage("playersPositionX",40);

updatelocalStorage("playersPositionY",272);

location.reload();

}

//going to bedroom\_room

if ( (player.x==120 || player.x==104) && player.y==48 && outCurrentRoom=="img/drawing\_room.png") {

updatelocalStorage(1,"bedRoom.png");

updatelocalStorage("playersPositionX",232);

updatelocalStorage("playersPositionY",400);

location.reload();

}

//end

//start i am in bathroom

if (player.x==360 && (player.y==192||player.y==208||player.y==204) && outCurrentRoom=="img/bath\_room.png") {

updatelocalStorage(1,"drawing\_room.png");

updatelocalStorage("playersPositionX",104);

updatelocalStorage("playersPositionY",96);

location.reload();

}

//end

//Level 1

if (player.x>=248 && player.y>=64 && player.y<=288 && level==1 && currentTask==1 && outCurrentRoom=="img/bedRoom.png") {

updatelocalStorage('currentTask',2);

navigationBar.center=getTaskBarText(1, 2);

navigationBar.left="";

navigationBar.right="";

navigationBar.visible=true;

arrow.visible=true;

setTimeout(function() {

navigationBar.visible=false;

}, 1000);

}

if (level==1 && currentTask==2 && outCurrentRoom=="img/drawing\_room.png") {

updatelocalStorage('currentTask',3);

arrowUpdate(-8,16,180,arrow);

navigationBar.center=getTaskBarText(1, 3);

navigationBar.left="";

navigationBar.right="";

navigationBar.visible=true;

setTimeout(function() {

navigationBar.visible=false;

}, 1000);

}

if (level==1 && currentTask==3 && outCurrentRoom=="img/bath\_room.png") {

updatelocalStorage('currentTask',4);

arrowUpdate(160,132,0,arrow);

navigationBar.center=getTaskBarText(1, 4);

navigationBar.left="";

navigationBar.right="";

navigationBar.visible=true;

setTimeout(function() {

navigationBar.visible=false;

}, 1000);

}

if (level==1 && currentTask==4 && outCurrentRoom=="img/drawing\_room.png") {

arrowUpdate(160,152,0,arrow);

navigationBar.center=getTaskBarText(1, 4);

navigationBar.left="";

navigationBar.right="";

navigationBar.visible=true;

setTimeout(function() {

navigationBar.visible=false;

}, 1000);

}

if (level==1 && currentTask==4 && outCurrentRoom=="img/kitchen\_room.png") {

updatelocalStorage('currentTask',5);

arrowUpdate(-40,180,180,arrow);

navigationBar.center=getTaskBarText(1, 5);

navigationBar.left="";

navigationBar.right="";

navigationBar.visible=true;

setTimeout(function() {

navigationBar.visible=false;

}, 1000);

}

if (level==1 && currentTask==5 && outCurrentRoom=="img/drawing\_room.png") {

arrowUpdate(0,256,90,arrow);

updatelocalStorage('currentTask',6);

navigationBar.center=getTaskBarText(1, 5);

navigationBar.left="";

navigationBar.right="";

navigationBar.visible=true;

setTimeout(function() {

navigationBar.visible=false;

}, 1000);

}

//End of level 1

if (level>1 && batteryTwoCharge<3) {

babyFeedIcon.visible = true;

}

/// Level 2 star

if(level >1){

if (currentTask==1 && level== 2 && outCurrentRoom=="img/drawing\_room.png") {

updatelocalStorage('currentTask',2);

navigationBar.center=getTaskBarText(2, 2);

navigationBar.left="";

navigationBar.right="";

navigationBar.visible=true;

setTimeout(function() {

navigationBar.visible=false;

}, 1000);

}

if ((player.x==168||player.x==152||player.x==184)&&(player.y==320||player.y==336||player.y==352) && outCurrentRoom=="img/bath\_room.png" && babyconfirmSceneCounter==0 && playerImage==2)

{

babyconfirmSceneCounter++;

var confirmScene = new ConfirmScene('Do you want bath your baby', 'Yes', 'No');

confirmScene.oncancel = function() {

babyconfirmSceneCounter=0;

game.rootScene.removeChild(confirmScene);

};

confirmScene.onaccept = function() {

window.location.href = 'bath.html';

};

game.rootScene.addChild(confirmScene);

}

}

/// Level 2 end

////Level 3

if (player.x==72 && player.y==160 && level>1 && playerImage==2) {

var r = confirm("Do you want to charge your baby's dress?");

if (r == true) {

window.location.href = 'dresschange.html';

}

}

if (level==3 && currentTask==1) {

navigationBar.center=getTaskBarText(3, 1);

navigationBar.left="";

navigationBar.right="";

navigationBar.visible=true;

setTimeout(function() {

navigationBar.visible=false;

}, 1000);

}

////

if (!(game.frame % 3)) {

this.walk++;

this.walk %= 3;

}

if ((this.vx && (this.x-8) % 16 == 0) || (this.vy && this.y % 16 == 0)) {

this.isMoving = false;

this.walk = 1;

}

} else if(playerMovement){

this.vx = this.vy = 0;

if (game.input.left) {

this.direction = 1;

this.vx = -4;

} else if (game.input.right) {

this.direction = 2;

this.vx = 4;

} else if (game.input.up) {

this.direction = 3;

this.vy = -4;

} else if (game.input.down) {

this.direction = 0;

this.vy = 4;

}

if (this.vx || this.vy) {

var x = this.x + (this.vx ? this.vx / Math.abs(this.vx) \* 16 : 0) + 16;

var y = this.y + (this.vy ? this.vy / Math.abs(this.vy) \* 16 : 0) + 16;

if (0 <= x && x < map.width && 0 <= y && y < map.height && !map.hitTest(x, y)) {

this.isMoving = true;

arguments.callee.call(this);

}

}

}

});

var stage = new Group();

stage.addChild(map);

//stage.addChild(foregroundMap);

var pad = new Pad();

pad.x =220 ;

pad.y = 380;

var leftpad = new Sprite(64, 64);

leftpad.image = game.assets['img/apad.png'];

leftpad.x=25;

leftpad.y=380;

/// navigationBar And TaskBar

var navigationBar = new NavigationBar("","","");

navigationBar.center=getTaskBarText(level, currentTask);

navigationBar.left="";

navigationBar.right="";

navigationBar.visible=false;

var taskbar = new Sprite(40,40);

taskbar.image = game.assets['img/taskbar.png'];

taskbar.x=300;

taskbar.addEventListener('touchend', function() {

navigationBar.visible=true;

arrowLevel1.visible=false;

setTimeout(function() {

navigationBar.visible=false;

}, 1000);

});

/// navigationBar And TaskBar End

var arrow = new Sprite(320,256);

arrow.image = game.assets['img/arrowright.png'];

arrow.scaleX=.1;

arrow.scaleY=.1;

arrow.x = 110;

arrow.y = 320;

arrow.rotation=90;

arrow.visible=false;

/\* arrow.x = 0;

arrow.y = 256;

arrow.rotation=90;

arrow.visible=true;\*/

///level 1 arrow

arrowLevel1.image = game.assets['img/arrowright.png'];

arrowLevel1.scaleX=.2;

arrowLevel1.scaleY=.2;

arrowLevel1.x = 110;

arrowLevel1.y = -45;

arrowLevel1.rotation=300;

arrowLevel1.visible=false;

if (level==1 && currentTask==1 && outCurrentRoom=="img/bedRoom.png") {

arrowLevel1.visible=true;

}

///end level 1 arrow

///Battery

///Battery

var batteryTemp = enchant.Class.create(enchant.Sprite, {

initialize: function(x, y,charge) {

enchant.Sprite.call(this, 64, 128);

this.x = x;

this.y = y;

this.scaleX=.6;

this.scaleY=.7;

this.rotation=90;

var image = 'img/battery'+charge+'.png';

this.image = game.assets[image];

}

});

var batteryOne = new batteryTemp(25,-40,batteryOneCharge);

var batteryTwo = new batteryTemp(125,-40,batteryTwoCharge);

var batteryThree = new batteryTemp(225,-40,batteryThreeCharge);

window.setInterval(batteryLoop, 60000);

function batteryLoop() {

if(batteryOneCharge>1){

batteryOneCharge--;

updatelocalStorage('batteryOne',batteryOneCharge);

var abc = 'img/battery'+batteryOneCharge+'.png';

batteryOne.image = game.assets[abc];

}

if(batteryTwoCharge>1){

batteryTwoCharge--;

updatelocalStorage('batteryTwo',batteryTwoCharge);

var abc = 'img/battery'+batteryTwoCharge+'.png';

batteryOne.image = game.assets[abc];

}

if(batteryThreeCharge>1){

batteryThreeCharge--;

updatelocalStorage('batteryThree',batteryThreeCharge);

var abc = 'img/battery'+batteryThreeCharge+'.png';

batteryOne.image = game.assets[abc];

}

}

///ENd Battery

///Baby

var baby = new Sprite(232,328);

baby.scaleX=.1;

baby.scaleY=.1;

baby.x=175;

baby.y=50;

baby.frame = [1,1,1,1,1,1,1,1,2,2,2,2,2,2,2,3,3,3,3,3,3,4,4,4,4,4,5,5,5,5,5,6,6,6,6,6,7,8,9];

baby.image = game.assets['img/n.png'];

baby.visible=false;

stage.addChild(baby);

if (outCurrentRoom=='img/bedRoom.png' && playerImage == 1) {

baby.visible=true;

}

///Baby feed

var babyFeedIcon = new Sprite(50,45);

babyFeedIcon.frame=[0,0,0,0,0,0,1,1];

babyFeedIcon.y=100;

babyFeedIcon.x=10;

babyFeedIcon.image = game.assets['img/babyFeedIcon.png'];

babyFeedIcon.visible = false;

babyFeedIcon.addEventListener('touchend', function() {

if(playerImage==2){

window.location.href = 'babyfeed.html';

}

else alert("Take your baby in your hand to feed please.")

});

///End Baby

stage.addChild(player);

game.rootScene.addChild(stage);

game.rootScene.addChild(arrowLevel1);

stage.addChild(arrow);

game.rootScene.addChild(taskbar);

game.rootScene.addChild(batteryOne);

game.rootScene.addChild(batteryTwo);

game.rootScene.addChild(batteryThree);

game.rootScene.addChild(navigationBar);

game.rootScene.addChild(babyFeedIcon);

game.rootScene.addChild(leftpad);

game.rootScene.addChild(pad);

game.rootScene.addEventListener('enterframe', function(e) {

var x = Math.min((game.width - 16) / 2 - player.x, 0);

var y = Math.min((game.height - 16) / 2 - player.y, 0);

x = Math.max(game.width, x + map.width) - map.width;

y = Math.max(game.height, y + map.height) - map.height;

stage.x = x;

stage.y = y;

});

};

game.start();

};

**Bath.js**

enchant();

var sopani2 = new Sprite(232, 348);

var lsnani2 = new Sprite(320, 512);

var soopCounter = 0;

var waterTapCounter = 0;

var towelCounter = 0;

var shampooCounter = 0;

var bathBatteryCounter = 0;

window.onload = function() {

var game = new Core(320, 512);

game.fps = 24;

game.preload(

'img/bathBattery.png',

'img/battery1.png',

'img/battery2.png',

'img/battery3.png',

'img/battery4.png',

'img/battery5.png',

'img/battery6.png',

'img/battery7.png',

"img/tlf.png",

"img/ls.png",

"img/lsnf.png",

"img/dropf.png",

"img/n.png",

"img/1.png",

"img/sop.png",

"img/lsn.png",

"img/tawl.png",

"img/sr.png",

"img/2.png",

"img/soapf.png",

"img/sp.png"

);

game.onload = function() {

//background

var bk = new Sprite(game.width, game.height);

bk.image = game.assets['img/1.png'];

game.rootScene.addChild(bk);

//myani

var ani = new Sprite(232, 348);

ani.image = game.assets["img/n.png"];

ani.x = 40;

ani.y = 150;

ani.frame = 0;

ani.scale(.8);

game.rootScene.addChild(ani);

ani.addEventListener("enterframe", function() {

this.frame = [

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,

3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,

4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,

5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,

6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,

7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7,

8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,

9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9,

];

});

var bk2 = new Sprite(game.width, game.height);

bk2.image = game.assets['img/2.png'];

bk2.y = -10;

game.rootScene.addChild(bk2);

//water tap

var cl = new Sprite(214, 235);

cl.image = game.assets['img/tawl.png'];

cl.x = -70;

cl.y = 230;

cl.scale(.4);

game.rootScene.addChild(cl);

cl.addEventListener('touchend', function() {

if(towelCounter==0){

towelCounter++;

bathBatteryCounter++;

bathBatteryFrameUndate(bathBatteryCounter);

if(batteryOneCharge+1<=7){

batteryOneCharge++;

updatelocalStorage('batteryOne',batteryOneCharge);

}

}

var twl = new Sprite(320, 512);

twl.image = game.assets["img/tlf.png"];

twl.x = 0;

twl.y = 0;

twl.frame = 0;

game.rootScene.addChild(twl);

twl.addEventListener("enterframe", function() {

this.frame = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2];

});

twl.tl.setTimeBased();

twl.tl.cue({

3000: function() {

this.scene.removeChild(this);

}

});

});

//soap

var soap = new Sprite(81, 48);

soap.image = game.assets['img/sop.png'];

soap.x = 250;

soap.y = 325;

game.rootScene.addChild(soap);

soap.addEventListener('touchend', function() {

if(soopCounter==0){

soopCounter++;

bathBatteryCounter++;

bathBatteryFrameUndate(bathBatteryCounter);

if(batteryOneCharge+1<=7){

batteryOneCharge++;

updatelocalStorage('batteryOne',batteryOneCharge);

}

}

var sopani = new Sprite(232, 348);

sopani.image = game.assets["img/soapf.png"];

sopani.x = 40;

sopani.y = 145;

sopani.scale(.8);

sopani.frame = 0;

game.rootScene.addChild(sopani);

sopani.addEventListener("enterframe", function() {

this.frame = [

0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,

3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,

4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,

5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,

6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,

7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7,

8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,

];

sopani.tl.setTimeBased();

sopani.tl.cue({

300: function() {

this.scene.removeChild(this);

}

});

});

sopani.tl.delay(100).then(function() {

sopani2.image = game.assets["img/sp.png"];

sopani2.x = 40;

sopani2.y = 145;

sopani2.scale(.8);

sopani2.frame = 8;

game.rootScene.addChild(sopani2);

})

});

// jel

var lsn = new Sprite(133, 193);

lsn.image = game.assets['img/lsn.png'];

lsn.x = 190;

lsn.y = 190;

lsn.scale(.5);

game.rootScene.addChild(lsn);

lsn.addEventListener('touchend', function() {

if(shampooCounter==0){

shampooCounter++;

bathBatteryCounter++;

bathBatteryFrameUndate(bathBatteryCounter);

if(batteryOneCharge+1<=7){

batteryOneCharge++;

updatelocalStorage('batteryOne',batteryOneCharge);

}

}

var lsnani = new Sprite(320, 512);

lsnani.image = game.assets["img/lsnf.png"];

lsnani.x = 0;

lsnani.y = 0;

lsnani.scale(.8);

lsnani.frame = 0;

game.rootScene.addChild(lsnani);

lsnani.addEventListener("enterframe", function() {

this.frame = [

0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,

3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,

4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,

5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,

];

lsnani.tl.setTimeBased();

lsnani.tl.cue({

100: function() {

this.scene.removeChild(this);

}

});

});

lsnani.tl.delay(1000).then(function() {

lsnani2.image = game.assets["img/ls.png"];

lsnani2.x = 0;

lsnani2.y = 5;

lsnani2.scale(.8);

lsnani2.frame = 8;

game.rootScene.addChild(lsnani2);

})

});

//shower

var sr = new Sprite(133, 193);

sr.image = game.assets['img/sr.png'];

sr.x = 100;

sr.y = -40;

game.rootScene.addChild(sr);

//drp

sr.addEventListener('touchend', function() {

if(waterTapCounter==0){

waterTapCounter++;

bathBatteryCounter++;

bathBatteryFrameUndate(bathBatteryCounter);

if(batteryOneCharge+1<7){

if(batteryOneCharge+3<=7)batteryOneCharge+=3;

else batteryOneCharge = 7;

updatelocalStorage('batteryOne',batteryOneCharge);

}

}

var drop = new Sprite(320, 512);

drop.image = game.assets['img/dropf.png'];

drop.x = 15;

drop.y = -30;

drop.frame = 0;

game.rootScene.addChild(drop);

drop.addEventListener("enterframe", function() {

this.frame = [

0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,

3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,

];

});

drop.tl.setTimeBased();

drop.tl.cue({

4000: function() {

this.scene.removeChild(this);

if (sopani2.scene == null) {

return;

} else {

sopani2.scene.removeChild(sopani2);

}

if (lsnani2.scene == null) {

return;

} else {

lsnani2.scene.removeChild(lsnani2);

}

}

});

});

///Battery

var batteryTemp = enchant.Class.create(enchant.Sprite, {

initialize: function(x, y,charge) {

enchant.Sprite.call(this, 64, 128);

this.x = x;

this.y = y;

this.scaleX=.6;

this.scaleY=.7;

this.rotation=90;

var image = 'img/battery'+charge+'.png';

this.image = game.assets[image];

}

});

var batteryOne = new batteryTemp(25,-40,batteryOneCharge);

var batteryTwo = new batteryTemp(125,-40,batteryTwoCharge);

var batteryThree = new batteryTemp(225,-40,batteryThreeCharge);

/\* window.setInterval(batteryLoop, 20000);

window.setInterval(gobackpage, 5000);

function batteryLoop() {

if(batteryOneCharge>1){

batteryOneCharge--;

updatelocalStorage('batteryOne',batteryOneCharge);

var abc = 'img/battery'+batteryOneCharge+'.png';

batteryOne.image = game.assets[abc];

}

if(batteryTwoCharge>1){

batteryTwoCharge--;

updatelocalStorage('batteryTwo',batteryTwoCharge);

var abc = 'img/battery'+batteryTwoCharge+'.png';

batteryOne.image = game.assets[abc];

}

if(batteryThreeCharge>1){

batteryThreeCharge--;

updatelocalStorage('batteryThree',batteryThreeCharge);

var abc = 'img/battery'+batteryThreeCharge+'.png';

batteryOne.image = game.assets[abc];

}

}

function gobackpage(){

if (batteryOneCharge ==7 && babyconfirmSceneCounter==0) {

babyconfirmSceneCounter++;

var confirmScene = new ConfirmScene('Bath is done, You baby is ok, Want to go back now?', 'Yes', 'No');

confirmScene.visible=false;

confirmScene.oncancel = function() {

babyconfirmSceneCounter=0;

game.rootScene.removeChild(confirmScene);

};

confirmScene.onaccept = function() {

updatelocalStorage('playersPositionX',136);

updatelocalStorage('playersPositionY',384);

window.location.href = 'index.html';

};

game.rootScene.addChild(confirmScene);

}

}

\*/

var bathBattery = new Sprite(128,320);

bathBattery.image = game.assets['img/bathBattery.png'];

bathBattery.x=-30;

bathBattery.y=5;

bathBatteryFrameUndate(bathBatteryCounter);

bathBattery.scale(.4);

function bathBatteryFrameUndate(n){

if (n == 0) bathBattery.frame=0;

else if(n == 1) {

bathBattery.frame=1;

var abc = 'img/battery'+batteryOneCharge+'.png';

batteryOne.image = game.assets[abc];

}

else if(n == 2) {

bathBattery.frame=2;

var abc = 'img/battery'+batteryOneCharge+'.png';

batteryOne.image = game.assets[abc];

}

else if(n == 3) {

bathBattery.frame=3;

var abc = 'img/battery'+batteryOneCharge+'.png';

batteryOne.image = game.assets[abc];

}

else if(n == 4) {

var abc = 'img/battery'+batteryOneCharge+'.png';

batteryOne.image = game.assets[abc];

bathBattery.addEventListener('touchend', function() {

window.location.href = 'index.html';

});

bathBattery.frame=[4,4,4,4,4,5,5,5,5];

}

}

game.rootScene.addChild(bathBattery);

///ENd Battery

game.rootScene.addChild(batteryOne);

game.rootScene.addChild(batteryTwo);

game.rootScene.addChild(batteryThree);

//end

};

game.start();

};

**babyfeed.js**

enchant();

window.onload = function() {

var game = new Core(320, 512);

game.fps = 24;

game.preload(

'img/feedBabyBackground.png',

'img/1.png',

'img/battery1.png',

'img/battery2.png',

'img/battery3.png',

'img/battery4.png',

'img/battery5.png',

'img/battery6.png',

'img/battery7.png',

'img/bathBattery.png',

'img/zinc.png',

'img/backgroundcollor.png'

);

game.onload = function() {

var backgroundmain = new Sprite(320, 512);

backgroundmain.image = game.assets['img/backgroundcollor.png'];

game.rootScene.addChild(backgroundmain);

var background = new Sprite(320, 512);

background.image = game.assets['img/feedBabyBackground.png'];

game.rootScene.addChild(background);

var zinc = new Sprite(55, 100);

zinc.image = game.assets['img/zinc.png'];

zinc.scale(.5);

zinc.x=250;

zinc.y=50;

game.rootScene.addChild(zinc);

zinc.addEventListener('touchend', function() {

bathBattery.frame=5;

if(batteryTwoCharge<7){

batteryTwoCharge++;

updatelocalStorage('batteryTwo',batteryTwoCharge);

}

var abc = 'img/battery'+batteryTwoCharge+'.png';

batteryTwo.image = game.assets[abc];

});

var food2 = new Sprite(55, 100);

food2.image = game.assets['img/zinc.png'];

food2.scale(.5);

food2.x=250;

food2.y=150;

game.rootScene.addChild(food2);

food2.addEventListener('touchend', function() {

if(batteryTwoCharge<7){

batteryTwoCharge++;

updatelocalStorage('batteryTwo',batteryTwoCharge);

}

var abc = 'img/battery'+batteryTwoCharge+'.png';

batteryTwo.image = game.assets[abc];

});

if(level==3){

var food3 = new Sprite(55, 100);

food3.image = game.assets['img/zinc.png'];

food3.scale(.5);

food3.x=250;

food3.y=250;

game.rootScene.addChild(food3);

food3.addEventListener('touchend', function() {

if(batteryTwoCharge<7){

batteryTwoCharge++;

updatelocalStorage('batteryTwo',batteryTwoCharge);

}

var abc = 'img/battery'+batteryTwoCharge+'.png';

batteryTwo.image = game.assets[abc];

});

}

///Battery

var batteryTemp = enchant.Class.create(enchant.Sprite, {

initialize: function(x, y,charge) {

enchant.Sprite.call(this, 64, 128);

this.x = x;

this.y = y;

this.scaleX=.6;

this.scaleY=.7;

this.rotation=90;

var image = 'img/battery'+charge+'.png';

this.image = game.assets[image];

}

});

var batteryOne = new batteryTemp(25,-40,batteryOneCharge);

var batteryTwo = new batteryTemp(125,-40,batteryTwoCharge);

var batteryThree = new batteryTemp(225,-40,batteryThreeCharge);

window.setInterval(batteryLoop, 60000);

function batteryLoop() {

if(batteryOneCharge>1){

batteryOneCharge--;

updatelocalStorage('batteryOne',batteryOneCharge);

var abc = 'img/battery'+batteryOneCharge+'.png';

batteryOne.image = game.assets[abc];

}

if(batteryTwoCharge>1){

batteryTwoCharge--;

updatelocalStorage('batteryTwo',batteryTwoCharge);

var abc = 'img/battery'+batteryTwoCharge+'.png';

batteryTwo.image = game.assets[abc];

}

if(batteryThreeCharge>1){

batteryThreeCharge--;

updatelocalStorage('batteryThree',batteryThreeCharge);

var abc = 'img/battery'+batteryThreeCharge+'.png';

batteryThree.image = game.assets[abc];

}

}

var bathBattery = new Sprite(128,320);

bathBattery.image = game.assets['img/bathBattery.png'];

bathBattery.x=-30;

bathBattery.y=5;

bathBattery.frame=1;

bathBattery.scale(.4);

bathBattery.addEventListener('touchend', function() {

if(bathBattery.frame==5){

window.location.href = 'index.html';

}

});

game.rootScene.addChild(bathBattery);

game.rootScene.addChild(batteryOne);

game.rootScene.addChild(batteryTwo);

game.rootScene.addChild(batteryThree);

///ENd Battery

};

game.start();

};

**dresschange.js**

enchant();

window.onload = function() {

var game = new Core(320, 512);

game.fps = 24;

game.preload(

'img/n.png',

'img/feedBabyBackground.png',

'img/1.png',

'img/battery1.png',

'img/battery2.png',

'img/battery3.png',

'img/battery4.png',

'img/battery5.png',

'img/battery6.png',

'img/battery7.png',

'img/bathBattery.png',

'img/zinc.png',

'img/backgroundcollor.png',

'img/bgSlider.png'

);

game.onload = function() {

var backgroundmain = new Sprite(320, 512);

backgroundmain.image = game.assets['img/backgroundcollor.png'];

game.rootScene.addChild(backgroundmain);

var background = new Sprite(232, 348);

background.image = game.assets['img/n.png'];

background.frame=1;

background.x=40;

background.y=70;

game.rootScene.addChild(background);

var zinc = new Sprite(55, 100);

zinc.image = game.assets['img/zinc.png'];

zinc.scale(.5);

zinc.x=250;

zinc.y=50;

game.rootScene.addChild(zinc);

zinc.addEventListener('touchend', function() {

if(batteryThreeCharge<7){

batteryThreeCharge++;

updatelocalStorage('batteryThree',batteryThreeCharge);

}

var abc = 'img/battery'+batteryThreeCharge+'.png';

batteryThree.image = game.assets[abc];

});

var food2 = new Sprite(55, 100);

food2.image = game.assets['img/zinc.png'];

food2.scale(.5);

food2.x=250;

food2.y=150;

game.rootScene.addChild(food2);

food2.addEventListener('touchend', function() {

if(batteryThreeCharge<7){

batteryThreeCharge++;

updatelocalStorage('batteryThree',batteryThreeCharge);

}

var abc = 'img/battery'+batteryThreeCharge+'.png';

batteryThree.image = game.assets[abc];

});

var food3 = new Sprite(55, 100);

food3.image = game.assets['img/zinc.png'];

food3.scale(.5);

food3.x=250;

food3.y=250;

game.rootScene.addChild(food3);

food3.addEventListener('touchend', function() {

if(batteryThreeCharge<7){

batteryThreeCharge++;

updatelocalStorage('batteryThree',batteryThreeCharge);

}

var abc = 'img/battery'+batteryThreeCharge+'.png';

batteryThree.image = game.assets[abc];

});

var bgSlider = new Sprite(320, 59);

bgSlider.image = game.assets['img/bgSlider.png'];

bgSlider.y= game.height-70;

game.rootScene.addChild(bgSlider);

///Dress Type 1

var dressStage = new Group();

var dress=[];

dress[1] = new Sprite(55, 100);

dress[1].image = game.assets['img/zinc.png'];

dress[1].y= game.height-90;

dress[1].x= 50;

dress[1].scale(.3);

dress[1].addEventListener('touchend', function() {

background.frame=5;

dressBattery.frame=5;

updatelocalStorage('level',3);

updatelocalStorage('currentTask',1);

});

dressStage.addChild(dress[1]);

game.rootScene.addChild(dressStage);

/\* food2.addEventListener('touchend', function() {

alert("food two has been clinked");

if(batteryTwoCharge<7){

batteryTwoCharge++;

updatelocalStorage('batteryTwo',batteryTwoCharge);

}

var abc = 'img/battery'+batteryTwoCharge+'.png';

batteryTwo.image = game.assets[abc];

});

\*/

////Dress type 1

///Battery

var batteryTemp = enchant.Class.create(enchant.Sprite, {

initialize: function(x, y,charge) {

enchant.Sprite.call(this, 64, 128);

this.x = x;

this.y = y;

this.scaleX=.6;

this.scaleY=.7;

this.rotation=90;

var image = 'img/battery'+charge+'.png';

this.image = game.assets[image];

}

});

var batteryOne = new batteryTemp(25,-40,batteryOneCharge);

var batteryTwo = new batteryTemp(125,-40,batteryTwoCharge);

var batteryThree = new batteryTemp(225,-40,batteryThreeCharge);

window.setInterval(batteryLoop, 60000);

function batteryLoop() {

if(batteryOneCharge>1){

batteryOneCharge--;

updatelocalStorage('batteryOne',batteryOneCharge);

var abc = 'img/battery'+batteryOneCharge+'.png';

batteryOne.image = game.assets[abc];

}

if(batteryTwoCharge>1){

batteryTwoCharge--;

updatelocalStorage('batteryTwo',batteryTwoCharge);

var abc = 'img/battery'+batteryTwoCharge+'.png';

batteryTwo.image = game.assets[abc];

}

if(batteryThreeCharge>1){

batteryThreeCharge--;

updatelocalStorage('batteryThree',batteryThreeCharge);

var abc = 'img/battery'+batteryThreeCharge+'.png';

batteryThree.image = game.assets[abc];

}

}

var dressBattery = new Sprite(128,320);

dressBattery.image = game.assets['img/bathBattery.png'];

dressBattery.x=-30;

dressBattery.y=5;

dressBattery.scale(.4);

dressBattery.addEventListener('touchend', function() {

if(dressBattery.frame==5){

window.location.href = 'index.html';

}

});

game.rootScene.addChild(dressBattery);

game.rootScene.addChild(batteryOne);

game.rootScene.addChild(batteryTwo);

game.rootScene.addChild(batteryThree);

///ENd Battery

};

game.start();

};

**database.js**

**var outCurrentRoom="bedRoom.png";**

**var playerMovement = true;**

**//// No back**

**/\*(function (global) {**

**if(typeof (global) === "undefined")**

**{**

**throw new Error("window is undefined");**

**}**

**var \_hash = "!";**

**var noBackPlease = function () {**

**global.location.href += "#";**

**// making sure we have the fruit available for juice....**

**// 50 milliseconds for just once do not cost much (^\_\_^)**

**global.setTimeout(function () {**

**global.location.href += "!";**

**}, 50);**

**};**

**// Earlier we had setInerval here....**

**global.onhashchange = function () {**

**if (global.location.hash !== \_hash) {**

**global.location.hash = \_hash;**

**}**

**};**

**alert("Hello");**

**global.onload = function () {**

**noBackPlease();**

**// disables backspace on page except on input fields and textarea..**

**document.body.onkeydown = function (e) {**

**var elm = e.target.nodeName.toLowerCase();**

**if (e.which === 8 && (elm !== 'input' && elm !== 'textarea')) {**

**e.preventDefault();**

**}**

**// stopping event bubbling up the DOM tree..**

**e.stopPropagation();**

**};**

**};**

**})(window);\*/**

**//// No back End**

**if (typeof(Storage) !== "undefined") {**

**var state = history.state || {};**

**var reloadCount = state.reloadCount || 0;**

**if (performance.navigation.type === 1) { // Reload**

**state.reloadCount = ++reloadCount;**

**history.replaceState(state, null, document.URL);**

**} else if (reloadCount) {**

**delete state.reloadCount;**

**reloadCount = 0;**

**history.replaceState(state, null, document.URL);**

**}**

**if (reloadCount ==0) {**

**// Now, do whatever you want...**

**if(!localStorage.getItem('1')) localStorage.setItem('1',outCurrentRoom);**

**}**

**}**

**if(!(+localStorage.getItem('playersPositionX')) ) localStorage.setItem('playersPositionX',152);**

**if(!(+localStorage.getItem('playersPositionY')) )localStorage.setItem('playersPositionY',240);**

**if(!(localStorage.getItem('level')) )localStorage.setItem('level',1);**

**if(!(localStorage.getItem('currentTask')) )localStorage.setItem('currentTask',1);**

**if(!(localStorage.getItem('batteryOne')) )localStorage.setItem('batteryOne',7);**

**if(!(localStorage.getItem('batteryTwo')) )localStorage.setItem('batteryTwo',7);**

**if(!(localStorage.getItem('batteryThree')) )localStorage.setItem('batteryThree',7);**

**if(!(localStorage.getItem('playerImage')) )localStorage.setItem('playerImage',1);**

**function updatelocalStorage(key,value){**

**localStorage.setItem(key,value);**

**}**

**levelOneNavBarText = [**

**"Find the bed", //1**

**"Find drawing room follow the red arrow.", //2**

**"Find bath room", //3**

**"Find Kitchen room", //4**

**"Find out Door", //5**

**"Find the Hospital", //6**

**"Enter the Hospital", //7**

**"You tour is over now go back to your home",//8**

**];**

**function getTaskBarText(level, currentTask){**

**if(level==1){**

**return levelOneNavBarText[currentTask-1];**

**}**

**else if(level ==2){**

**if(currentTask==1){**

**return "You tour is over now go back to your home";**

**}**

**else if(currentTask==2){**

**return "Now go to bathroom cause your <br> baby's helth depend's on it";**

**}**

**}**

**else if(level ==3){**

**if(currentTask==1){**

**return "Visit doctor Baby's Health isn't good";**

**}**

**else if(currentTask==2){**

**return "";**

**}**

**}**

**else if(level ==4){**

**if(currentTask==1){**

**}**

**else if(currentTask==2){**

**}**

**}**

**else if(level ==5){**

**if(currentTask==1){**

**}**

**else if(currentTask==2){**

**}**

**}**

**}**

**function arrowUpdate(x,y,rotate,object){**

**object.x = x;**

**object.y = y;**

**object.rotation=rotate;**

**object.visible=true;**

**}**

**var level = +localStorage.getItem('level');**

**var currentTask = +localStorage.getItem('currentTask');**

**var taskbarText = "";**

**var batteryOneCharge = +localStorage.getItem('batteryOne');**

**var batteryTwoCharge = +localStorage.getItem('batteryTwo');**

**var batteryThreeCharge = +localStorage.getItem('batteryThree');**

**var babyconfirmSceneCounter = 0;**

**var playerImage = +localStorage.getItem('playerImage');**

**var carcounter=0;**

**///This a just a simple javascript array for level one doctor text**

**/// Copy and paste as many text as you want in oder**

**var level1DoctorChatText =[**

**//'qwertyuiopasdfghjklzb', One containt this much text**

**'ma: asalamualikum apa , <br>kemon achen?',**

**'dr: eito valo achi ,<br> apni kamon achen?',**

**'ma:valo.',**

**'dr:apner sisu kamon ache? mone rakben nobojatok er kheal ra oti guruttopurno.',**

**'ma: apa, ki ki dekhle <br>bujobo segulu amr<br> sisur jonoo bopodjonok?',**

**'dr: <br> মায়ের দুধ খেতে না পারা ,<br> িঁচুনি, অজ্ঞান হয়ে পড়া, <br> নেতিয়ে পড়া বা স্বাভাবিকের চেয়ে কম নড়াচড়া করা <br>',**

**'নবজাতকের শ্বাস নিতে কষ্ট হওয়া,<br> দ্রুত শ্বাস বা বুকের নিচের অংশ <br> মারাত্মকভাবে ভেতরে ঢুকে যাওয়া',**

**'শরীরের তাপ বেড়ে যাওয়া বা জ্বর <br>এবং শরীরের তাপমাত্রা কমে যাওয়া',**

**'নাভি পেকে যাওয়া <br> ma: egulo hole ki korbo apa ?',**

**'dr :নবজাতকের যেকোনো বিপদচিহ্ন বা জটিলতা <br> দেখা দিলে দ্রুত স্বাস্থ্যকর্মীর কাছে বা নিকটবর্তী স্বাস্থ্যকেন্দ্রে ',**

**'নিয়ে যেতে হবে। যেমন: কমিউনিটি ক্লিনিক,<br> ইউনিয়ন স্বাস্থ্য ও পরিবার কল্যাণ কেন্দ্র, উপজেলা স্বাস্থ্য ',**

**'কমপ্লেক্স, জেলা হাসপাতাল, মেডিকেল কলেজ <br> হাসপাতাল, এনজিও এবং বেসরকারি ক্লিনিক ইত্যাদি',**

**'প্রয়োজনীয় যানবাহনের ব্যবস্থা করতে হবে,<br>রোগীর সাথে কে যাবে তা ঠিক করতে হবে,',**

**'প্রয়োজনীয় মোবাইল নম্বর যোগাড় করে রাখতে হবে,<br>প্রয়োজনীয় টাকা-পয়সা সঙ্গে নিতে হবে,',**

**'রোগীর যাতে কোনো কষ্ট না হয়, <br> সেদিকে লক্ষ্য রাখতে হবে',**

**'ma: dhonnobad apa,aj tahole ashi :) <br> dr: thik asche.'**

**];**

**var level2DoctorChatText =[**

**];**

**var level3DoctorChatText =[**

**'you :Doctor Apa, amr sisu kemon jani br br oggan hoea jasse, ',**

**'chokh boshe geche abong , pani khachhe na',**

**'doctor apa: dekhi !!...delay',**

**'hmm , ei dohoroner lokkhon holo ডায়রিয়া er lokhhon',**

**'you : ekhn ami ki korbo ?',**

**'doctor apa: বারবার তরল খাবার খেতে দিন ঘন ঘন মায়ের দুধ দিন এবং প্রতিবার বেশি ',**

**'সময় ধরে দিন যদি শিশু শুধুই বুকের দুধ খায়, তাহলে সেই সাথে ওআরএস বা নিরাপদ পানি',**

**' দিন >যদি শিশু শুধুই বুকের দুধ না খায়, তাহলে ওআরএস, সুপারিশকৃত তরল, যেমন – ভাতের ',**

**'মাড়, চিড়ার পানি অথবা নিরাপদ পানি খেতে দিন > ২ বছর বয়স পর্যন্ত শিশুকে প্রতিবার পাতলা ',**

**'পায়খানার পর ৫০-১০০ মিলি এবং ২ বছর বা বেশি বয়সের শিশুকে প্রতিবার পাতলা পায়খানার ',**

**'পর ১০০-২০০ মিলি অতিরিক্ত তরল খাবার খেতে দিন।জিংক বড়ি খেতে দিন ছোট শিশু যারা জিংক ',**

**'বড়ি চিবিয়ে খেতে পারে না, তাদেরকে বুকের দুধ, ওআরএস অথবা নিরাপদ পরিস্কার পানির সাথে',**

**'মিশিয়ে ছোট কাপ বা চামচ দিয়ে জিংক বড়ি খাওয়ান।',**

**'You : dhonnobad apa '**

**];**

**function doctorTalk(chatBoxText , chatBox,counter){**

**//Level 1 doctor Chat**

**if (level == 1) {**

**if(counter<level1DoctorChatText.length)**

**{**

**chatBoxText.text= level1DoctorChatText[counter];**

**return ++counter;**

**}**

**else{**

**playerMovement=true;**

**chatBoxText.visible=false;**

**chatBox.visible=false;**

**counter = 0;**

**return counter;**

**}**

**}**

**/// level 2 doctor chat**

**else if (level==2) {**

**if(counter<level2DoctorChatText.length)**

**{**

**chatBoxText.text= level2DoctorChatText[counter];**

**return ++counter;**

**}**

**else{**

**playerMovement=true;**

**chatBoxText.visible=false;**

**chatBox.visible=false;**

**counter = 0;**

**return counter;**

**}**

**}**

**/// level 2 doctor chat**

**else if (level==3) {**

**if(counter<level3DoctorChatText.length)**

**{**

**chatBoxText.text= level3DoctorChatText[counter];**

**return ++counter;**

**}**

**else{**

**playerMovement=true;**

**chatBoxText.visible=false;**

**chatBox.visible=false;**

**counter = 0;**

**updatelocalStorage('level',3);**

**updatelocalStorage('currentTask',2);**

**return counter;**

**}**

**}**

**else if (level==4) {**

**}**

**else if (level==5) {**

**}**

**else if (level==6) {**

**}**

**}**

**var badRoomCollisionData =**

**[**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1],**

**[1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1],**

**[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1],**

**[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1],**

**[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1],**

**[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],**

**[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],**

**[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],**

**[1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],**

**[1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],**

**[1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],**

**[1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],**

**[1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],**

**[1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],**

**[1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],**

**[1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],**

**[1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1],**

**[1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]**

**],**

**drawingRoomCollisionData =**

**[**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]**

**],**

**bathRoomCollisionData =**

**[**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]**

**],**

**kitchenRoomCollisionData =**

**[**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],**

**[1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],**

**[1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],**

**[1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],**

**[1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],**

**[1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],**

**[1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],**

**[1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],**

**[1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]**

**];**

Appendix 2 – User guide

User have to play as a role of “New Mother”, he/she have to follow the instruction which will appear in the navigation bar, as well as on screen. The game is task driven, so user have to finish each task to move to the next.

Appendix 3 – Installation guide

User need to Download the Apk file of the game and then just install it in phone.