1. **Introduction**
   1. **Introduction:**

Get your hands on the ultimate color matching game with a unique & challenging gameplay! Try to match the color of the cube with the coming same colored cube! If you do so that will increase one life and as far as you go that will increase your score.

**1.2 Scope of Our Game**

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 “Color Catch”. “Color Catch” is a single-player game on the Cross-platform. The player(cube) will progress through levels which require precise manipulation of the environment, though the game Encourages creativity and daring via branching pathways. The episodic structure of the game facilitates the pace of the story. The player(cube) will increase the score by getting the same colored cube and the cube color also be changing.

**1.3 System Environment**

Gamer

Input

(Keypad/game pad)

Renders

(Display)

Script

(Compile)

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 occur (if the value is changed) this object send to renders to display the things (a character can change its place).

**1.4 User Story of Our Game**

“Color Catch” is a cool colored game. It is a multi-platform game which is supported by PC, web player, android phone, IOS and other platforms also. So the gamer can use any of these platforms to run the game.

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” Gamer can push the game using Escape key and can use Resume to resume the game. User can change Game’s controls like key binding and can quit the game.

1. **Technology and Literature Review**
   1. **About Tools And Technology**

**2.1.1 What is c# ?**

**C# Language**

C# syntax is highly expressive, yet it is also simple and easy to learn. The curly-brace syntax of C# will be instantly recognizable to anyone familiar with C, C++ or Java. Developers who know any of these languages are typically able to begin to work productively in C# within a very short time. C# syntax simplifies many of the complexities of C++ and provides powerful features such as nullable value types, enumerations, delegates, lambda expressions and direct memory access, which are not found in Java. C# supports generic methods and types, which provide increased type safety and performance, and iterators, which enable implementers of collection classes to define custom iteration behaviours that are simple to use by client code. Language-Integrated Query (LINQ) expressions make the strongly-typed query a first-class language construct.

In addition to these basic object-oriented principles, C# makes it easy to develop software components through several innovative language constructs, including the following:+

* Encapsulated method signatures called *delegates*, which enable type-safe event notifications.
* Properties, which serve as accessors for private member variables.
* Attributes, which provide declarative metadata about types at run time.
* Inline XML documentation comments.
* Language-Integrated Query (LINQ) which provides built-in query capabilities across a variety of data sources.

**2.1.2 What is Unity3d ?**

Unity3D is a powerful cross-platform 3D engine and a user friendly development environment. Easy enough for the beginner and powerful enough for the expert; Unity should interest anybody who wants to easily create 3D games and applications for mobile, desktop, the web, and consoles.

A workspace for creative people

The Unity Editor is a creative hub for artists, designers, developers and other team members. Available on Windows and Mac, it includes 2D and 3D scene design tools, instant play mode for rapid editing and iteration, and a powerful animation system.

* **Storytelling:** The Timeline tool gives artists the power to create stunning cinematic content & gameplay sequences.
* **Cinematic content:** With Cinemachine’s suite of smart cameras, you can control shots just like a movie director from within the Editor.
* **AI pathfinding tools:** your NPC will navigate the world in no time. Advanced profiling tools offer developers insights they need to optimize performance.
* **Custom tools:** You can extend the Editor with whatever tools you need to match your team’s workflow.

**3. Specific Requirements**

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

**3.1 User Interfaces**

Every game must has 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.

**3.2 Software and Hardware Requirements**

**3.2.1 Hardware Requirements**

“Color Catch” 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. “Color Catch” has been developed for Android developed Version and all subsequent releases. In the future we released in the android platform. 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.

**3.2.2 Software Requirements**

“Color Catch” has been developed using a series of game development tools.

Working tools and platform

• Unity3D

• Autodesk Maya Or 3ds Max

• Blender

• Android Software Development Kit (Android SDK): Software development kit for applications on the Android platform. We want to release this game in the Android platform.

**4. 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 the 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 (developers).

**4.1 Analysis Model of Our Game Project**

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

**4.2 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.

**5.Figures**

**5.1 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 swimlane diagram connects UX with background programming which are the two important views of a game SRS

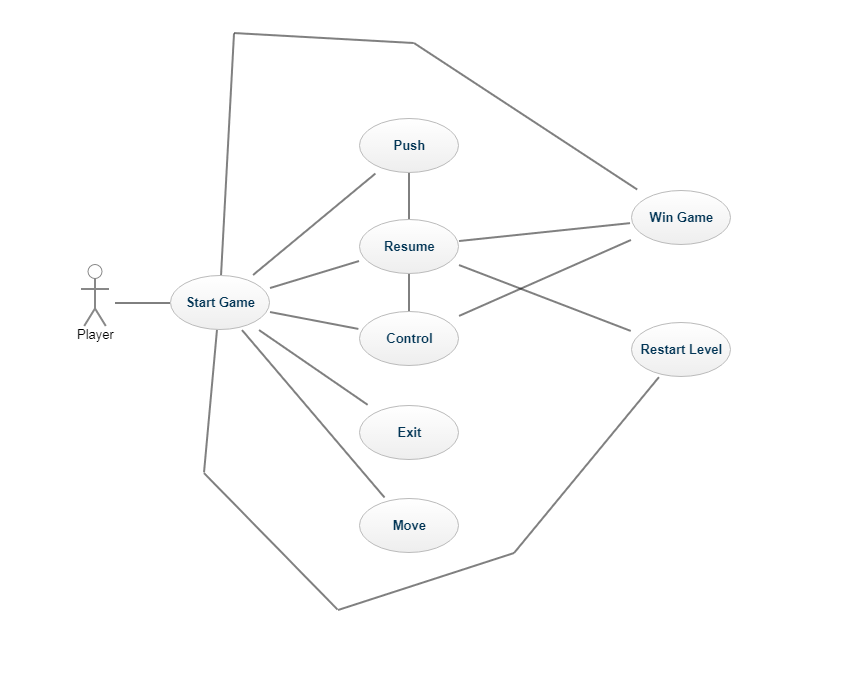
|  |  |  |
| --- | --- | --- |
| **Level – 0** | **Level – 1** | **Level – 2** |
| Game ( Color Catch) | Play | New Game |
| Resume Game |
| Select Level |
| Exit Game |
| Options | Show Control |
| Change Configuration (Graphics) |
| Change Sound/ Music Volume |
| Score Board | View Scores |
| Reset Score Board |
| Life | View Life |
| Quit | - |

**5.1.1 Use Case Diagrams**

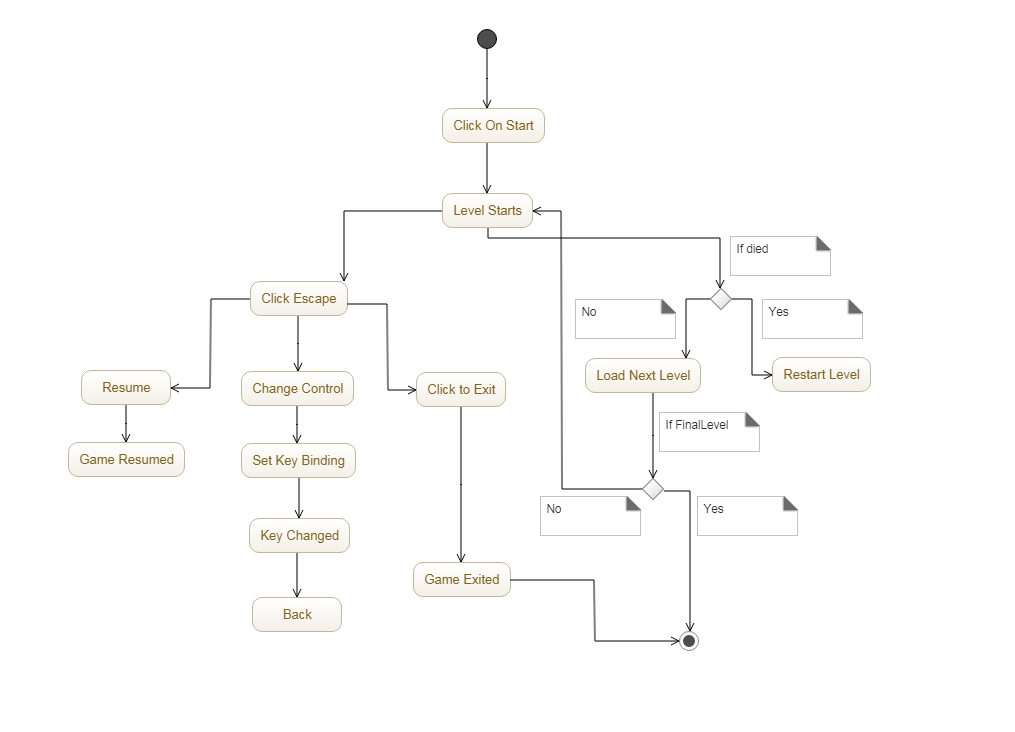
Player

System

Fig 1: Level 0 for Game UX



**5.2 Activity Diagram**



**5.3 Data Model**

If software requirements include the need to create, extend or interface with database or if complex data structures must be constructed and manipulated, the software team may choose to create a data model as part of overall requirements modeling. Although our game has many data objects, it does not have any data storage. All the objects and their related data are handled by the game engine. So the developers need not think about data storage. For this reason, data model is redundant for this game project.

**5.4 Behavioral Model**

The Behavioral 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.

**5.5 State Diagram**

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

**Checking**

Do: isclicked

**Push Menu**

**Start Game**

**Idle**

**Welcome Screen**

**“Quit”**

**“Control”**

**“Resume”**

To idle

**Close Game**

**Control Menu**

**“Return”**

**“Move Right”**

**“Move Left”**

**“Back”**

**5.6 Sequence Diagram**

**Active Object**

**Level Loaded**

Click resume game

**Backend**

Open game

**UX**

**Checking**

lookup

level returned

Fig 31: Sequence diagram (Resume Game)

**Backend**

**UX**

Press Game Pause

**Taking input**

Click back to main menu

Fig 33: Sequence diagram (Exit Game)

**Pause Menu Appears**

**Main Menu Appears**

**UX**

**Backend**

Open game

Click options

**Option Menu Appears**

Click back to show controls

**Controls Showed**

Fig 34: Sequence diagram (Show Control)

**6. Future Enhancement**

**6.1 Same colored cube Counter**

**6.2 Descriptions and Priority**

There will be same colored cube here and there in the scene for the Color Catch which he needs to collect. Same colored cube collection is used for the measurement of game score. The more the player collects same colored cube, the more points he gets.

**6.3 Stimulus/Response Sequences**

Step 1: Player notices same colored cube in a certain place

Step 2: He moves Color Catch to the place

Step 3: Color Catch automatically collects when gets in touch of it.

**6.4 Functional Requirements**

REQ 1: Store Color Catch collections during the game

REQ 2: Calculate points depending on total same colored cube collected

**6.5 Timer**

**6.6 Descriptions and Priority**

There will be a timer in the scene for keeping track of the time player takes to finish the level. This timer is used for the measurement of game score. The less time the player takes, the more points he gets.

**6.7 Stimulus/Response Sequences**

Step 1: Timer starts automatically at game begin.

Step 2: Player finishes the game.

Step 3: Timer stops.

**6.8 Functional Requirements**

REQ 1: Keep track of taken time of the game.

REQ 2: Calculate points depending on timer.

**7. 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.

**7.1 Construction of the Game**

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

Physics Engine

Collision Detection and Handling

Input Recognition

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

Scene Integration (transition of one level to the next)

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

**7.2 Integration with Android**

Unity3D's 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 Android from the list of options, build the project, and upload it to one of our own devices. A separate license is required for this functionality, which has already been obtained by one of the members of our group.

**7.3 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 | All three members made the decision about game levels together | 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 | Nadia and Tahmid worked on Unity game engine | Unity game engine | Ability to angle interactive portions of levels |
| Graphics Design | Knowledge of graphical modeling and implementation | Arif and Nadia worked for creating 3d models | 3d model design using Maya and 3dsMax | Visibility of the details on the 3d models |
| 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 | Background images from the internet | Level size dependent on hardware configuration |
| Documentation | Knowledge about SRS and Formal Report Writing | Arif and Nadia worked more on Reporting | Idea from Existing Reports (Ex. IITDU Online Judge System) | Game Reports are Different from Conventional ones |

**7.4 Implementation Tools Required**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Product of** | **Tool** | **Usage** | **Work exp.** |
| F:\Users\Nadia\Desktop\unity-35-released-eat-3d-740x314.jpg | Unity Technologies | Unity3d | Game Engine | Backend activity |
| F:\Users\Nadia\Desktop\3ds-max-logo.jpg | Autodesk | 3ds Max | Graphics Design and Animation | Create 3d Model and Animate |
| F:\Users\Nadia\Desktop\Autodesk_Maya_ruukasu.png | Maya | Graphics Design and Animation | Create 3d Model |
| F:\Users\Nadia\Desktop\images.jpg | Adobe | Photoshop | Picture Edit | 3d Model textures |
| F:\Users\Nadia\Desktop\reel.jpg | Microsoft Windows | Movie Maker | Create Videos | Game story creation |

**8. Conclusion**

A software project means a lot of experience. In this section we summarize the experience gained by project team during development of “Color Catch”.

**8.1 The Obstacles**

1. Working with game engine completely a new experience for us. Normally we are working with different OO languages, DBMS, mark up languages etc.

2. We adopt these things by video tutorials, text tutorials, internet and learning materials given by the tools themselves. It's a matter of time, patience and hard work.

3. It is very sensible work and it demands much time because the game engines try to connect game environment with the real world.

4. Creating a 3d model is very difficult because you need to work with each and every point of the model.

5. The Exists game engines demands vast knowledge about its properties, sections and sub-sections.

After all the thing is that a game project is not a project of 6 or 8 months for three people!

**8.2 The 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.

8.3 Future Plan

Level Extension

Improve Graphical Representation

Introduce new game features

Introduce new environment and scenes

Take user response through website and produce web rank list

**8.4 Last Few Words**

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.

If any kind of suggestion, improvements, more efficient development idea please feel free to communicate with us.