1. **INTRODUCATION**

Puzzle Game 3D is the application which is for Windows platform as well as for Android.at primary stage the game divided into 5 Levels of difficulty. The game saved coins/tokens in it preference of internal memory.

**1.1 Purpose**

The purpose of this document is to give a detailed description of the requirements for the “Puzzle Game 3D” software. It will illustrate the purpose and complete declaration for the development of system. It will also explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to a customer for its approval and a reference for developing the first version of the system for the development team.

**1.2 Scope**

The “Puzzle Game 3D” is a Desktop Application & mobile application which entertain people and also helpful for improve their concentration. The application should be free to download from either a mobile phone application store or similar services.

* 1. **Objective**

To make useful & interesting 3D Game using Windows Desktop Application & android smart phone technology. Puzzle Game 3D is windows desktop game / an android game application. Which is supported in any android smartphone & windows desktop.

1. **Project Management**

Effective management of a software project depends on thoroughly planning the progress of the project. A well-planned strategy leads to the best and optimal use of the resources available and ensures completion of project on time. Project plan sets out the resources available to the project, the work breakdown and a schedule for carrying out the work. The project needs a lot of research and thus scheduling was a difficult task as there was a need for carrying out a lot of study about various algorithms and techniques and testing them at various stages, thus maintaining the schedule was also difficult.

**2.1 Project Development Approach and Justification**

The life cycle model that has been followed for developing the system is the “Iterative Model”. This model is shown in the figure.

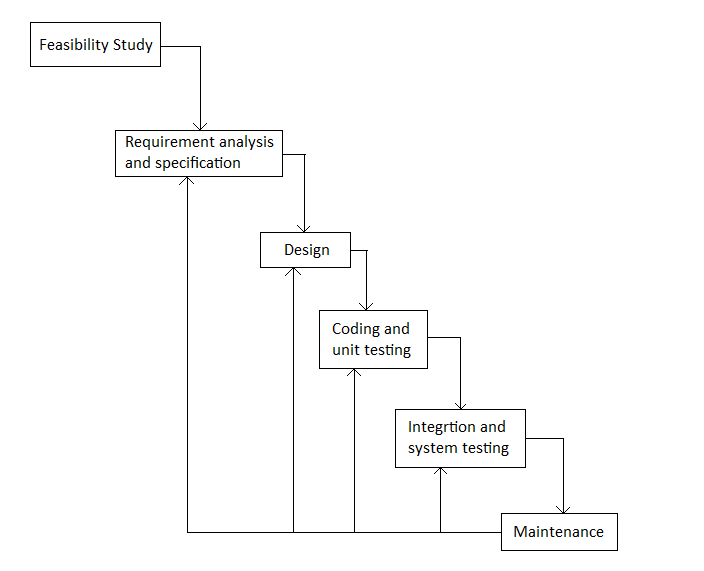


Figure 2.1 Process Model

In this model, Deaf and dumb communicator starts with an initial feasibility study phase. Once it is clear that the system can be implemented with necessary technical and economic support the next phase of requirement gathering and analysis is under taken.

When the system developers are clear with all requirement specifications they go for design. After preparing the design and implementation plan the coding is done and unit testing is performed. All units are integrated together for entire system testing in next phase. At the end the maintenance of the system is to be taken care of.

The main advantage of taking up Scrum Model is of Phase Containment of Errors. That is it provides the system developers the concept of feedback path, so it becomes easy for them to correct the errors committed during a phase, as and when these are detected in a later phase. This model is also advantageous because of its simplicity. The only thing here to be taken into consideration is that later detection of errors in previous phases may make the development costlier.

**JUSTIFICATION**

This model is best suited for this system as the customer mostly has clear idea about the requirements. As well as there are very less possibilities for major changes in requirements. It makes the implementation easier in terms of simplicity. There are no

Risks present in developing the system and step by step development is desirable.

* 1. **Project Effort, Time and Cost Estimation**

**Function Point (FP) Analysis (Theory)**

For a software project, we are interested primarily in estimating the cost and duration of the project. To arrive at these, we need to have a fix on the ‘size’ of the software that is to be developed. Since a major part of the cost of any software project is the effort expended by skilled software developers, accurate estimation of cost and schedule. Figure depicts a simple estimation model relating size, effort, schedule and cost.

* **Size estimate**

The size of the software to be developed is one of the parameters to be estimated. Some size measures of software are lines of code, number of function point, number of objects, number of reports, number of screens etc.

* **Effort estimate**

The estimate for the manpower that is required for the software product. Effort is normally measured in term of person-hours or person-days or person-month or person-years, with conversion factors to convert from one measuring unit to another.

* **Schedule estimate**

Schedule is the duration between the start of the project and the end of the project. Schedule estimate may include high-level intermediate milestones, like end of various phases.

* **Cost estimate**

A major component of costs is the manpower cost in any software project. Apart from that, other costs, such as travel, communication facilities, project specific training, hardware and software for the project team need to be estimated.

Considering the above, a simplistic definition of software project estimate could be “prediction, within a given tolerance, of the size of the software to be developed and the effort, cost and calendar time required to develop the software.”

**2.2.1 Project Estimates:**

Historical Data Used:

For developing this #3D Windows Game/ Android application data information of puzzle game and Snake game / application, Subway Surfer was used. From that application only idea of different function was gather and implemented. Also different methods were executed.

Estimation Technique Used:

For this project Heuristic Technique i.e. COCOMO was used. Using that project size, project duration, and efforts was calculated.

Estimation:

Line of code = no. of files \* avg. no. of line

=8 \* 130

=1040= 1.04(KLOC)

Since project is organic so a1=2.4 and a2=1.05

Efforts= a1 \* (KLOC) ^a2

=2.4 \* (1.04) ^1.05

= 2.4 \* 1.0420 = 2.5

=3 person month.

**Functional Point Calculation**

|  |  |  |
| --- | --- | --- |
| **Sr.no** | **Factors** | **value** |
| 1 | Data Communication | 4 |
| 2 | Distributed Data Processing | 4 |
| 3 | Performance | 4 |
| 4 | Heavily used Configuration | 2 |
| 5 | Transaction Rate | 3 |
| 6 | End User Efficiency | 3 |
| 7 | Online Update | 4 |
| 8 | Complex logic | 4 |
| 9 | Reusability | 5 |
| 10 | Conversion ease | 5 |
| 11 | Operational ease | 3 |
| 12 | multiple sites | 2 |
| 13 | facilitate change | 4 |
|  | **Total** | **47** |

UFP= (no of ip)\*4+ (no of op)\*5+ (no of inquiries)\*4+ (no of files)\*10+ (no of interface)\*10

= (2)\*4+ (2)\*5+ (4)\*4+ (8)\*10+ (5)\*10

=8+10+16+80+50

UFP =164

TCF=0.65+0.01\*47

=1.12

FP=UFP\*TCF

=164\*1.12

=183.68 **FP =183.68**

**COCOMO Model (Theory)**

Boehm postulated that any software development project can be classified into one of the

Following three categories based on the development complexity:

• Organic

• Semidetached

• Embedded

|  |  |
| --- | --- |
| **Phase** | **Time Duration** |
| Feasibility Study | 1 Week |
| Requirement Gathering and Analysis | 1 Week |
| Design | 6 Week |
| Coding | 5 Week |
| Testing and Maintenance | 8 Week |
| Documentation | 1 Week |

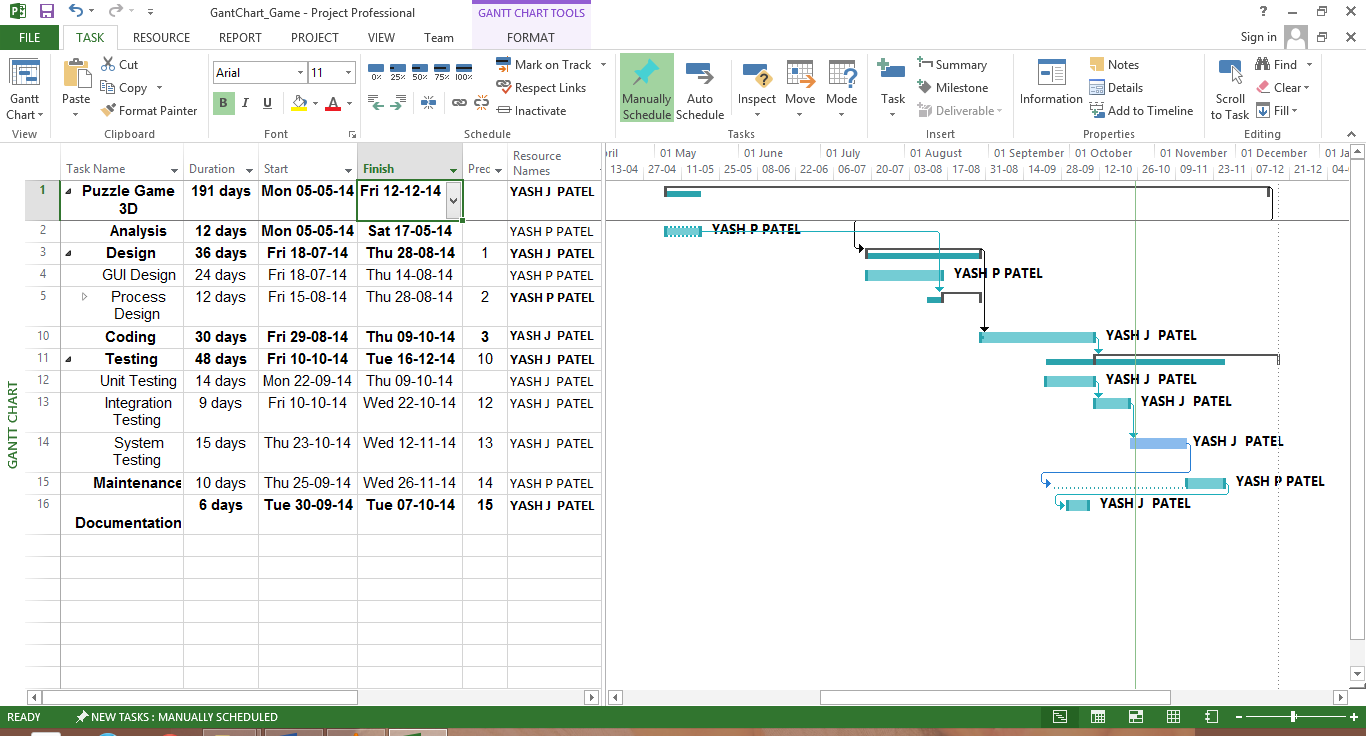
* 1. **Project Scheduling**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task Name | Duration | Start | Finish | Predecessors | Resource Names |
| **Puzzle Game 3D** | **191 days** | **Mon 05-05-14** | **Fri 12-12-14** |  | **YASH J PATEL** |
| **Analysis** | **12 days** | **Mon 05-05-14** | **Sat 17-05-14** |  | YASH P PATEL |
| **Design** | **36 days** | **Fri 18-07-14** | **Thu 28-08-14** | 1 | **YASH J PATEL** |
| GUI Design | 24 days | Fri 18-07-14 | Thu 14-08-14 |  | YASH P PATEL |
| Process Design | 10 days | Fri 15-08-14 | Thu 28-08-14 | 2 | **YASH P PATEL** |
| Class Diagram | 5 days | Sat 09-08-14 | Thu 14-08-14 |  | YASH P PATEL |
| Use Case | 3 days | Mon 11-08-14 | Wed 13-08-14 | 6 | YASH J PATEL |
| Statechart Diagram | 2 days | Tue 12-08-14 | Wed 13-08-14 | 7 | YASH P PATEL |
| Sequence Diagram | 2 days | Wed 13-08-14 | Thu 14-08-14 | 8 | YASH J PATEL |
| **Coding** | **30 days** | **Fri 29-08-14** | **Thu 09-10-14** | **3** | **YASH J PATEL** |
| **Testing** | **48 days** | **Fri 10-10-14** | **Tue 16-12-14** | 10 | **YASH J PATEL** |
| Unit Testing | 14 days | Mon 22-09-14 | Thu 09-10-14 |  | YASH J PATEL |
| Integration Testing | 9 days | Fri 10-10-14 | Wed 22-10-14 | 12 | YASH J PATEL |
| System Testing | 15 days | Thu 23-10-14 | Wed 12-11-14 | 13 | YASH J PATEL |
| **Maintenance** | 10 days | Thu 25-09-14 | Wed 26-11-14 | 14 | YASH P PATEL |
| **Documentation** | **6 days** | **Tue 30-09-14** | **Tue 07-10-14** | **15** | **YASH J PATEL** |

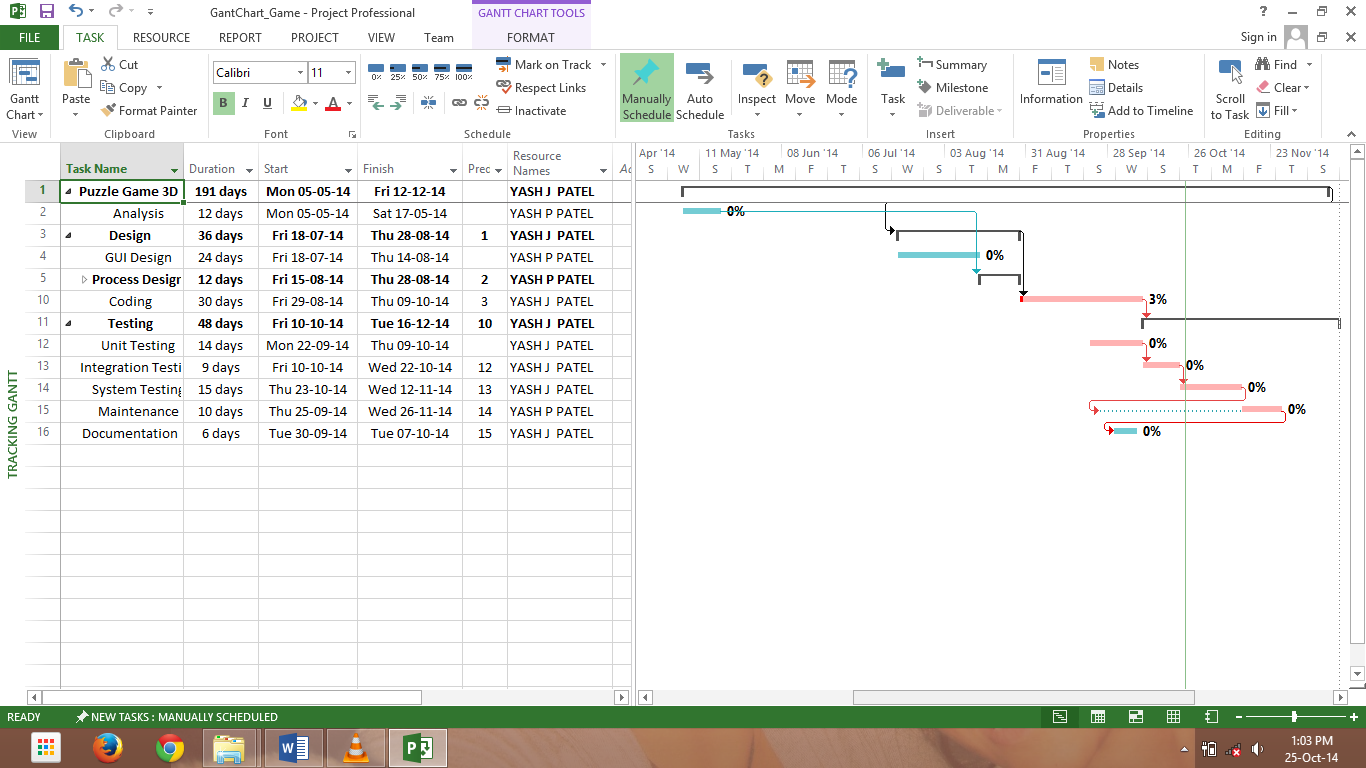
**Resource Sheet:**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Resource Name | Type | Material Label | Initials | Group | Max. Units | Std. Rate | Ovt. Rate | Cost/Use | Accrue At | Base Calendar |
| **YASH P PATEL** | **Work** |  | **Y** |  | **40%** | **$0.00/hr** | **$0.00/hr** | **$0.00** | **Prorated** | **Standard** |
| **YASH J PATEL** | **Work** |  | **Y** |  | **60%** | **$0.00/hr** | **$0.00/hr** | **$0.00** | **Prorated** | **Standard** |

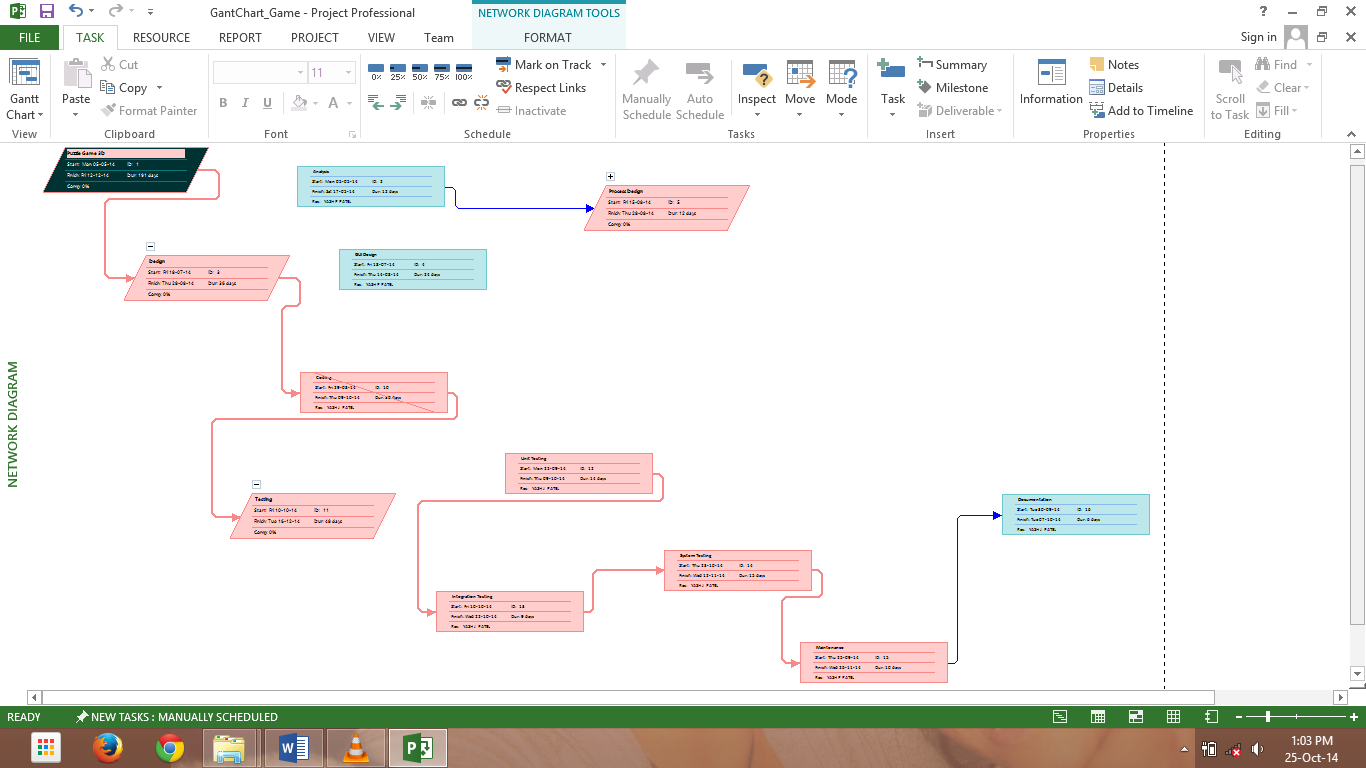
**(Gantt chart/ Tracking Gantt / Network Chart)**



**Gantt chart**



**Tracking Gant Chart**



**Network Chart**

**2.4 Project Resource:**

2.4.1 People:

People resources are: YASH J PATEL

2.4.2 Hardware and Software:

For implementing this project Windows Xp / Android Jelly bean or above version is required and at least 512 MB RAM. Good Processing Speed For better Performance and execution is recommended. Also it will work on all Android platform above Jelly Bean. Graphics must be required for this Application.

2.4.3 Special Resource:

Unity 3D, Maya Design, Blender, Eclipse Are required as a special resource for development of project.

**Staff Organization:**

Team Structure:

Our team is made up of one member. So whole project is develop and manage by me only. So analysis, design, implementation, and testing etc. were done by me.

Management Reporting:

Every week I need to submit our report of completion to our faculty. Also appropriate feedback was given by them using which we can make system more efficient.

**Miscellaneous Plans:**

System testing plan:

During project development only several testing were done by me. Also core modules are under development and are tested several times.

Validation and Verification:

All module are validated and verified using validation code. In each module both input and output were validated and verified.

System testing Plan:

System is to be tested after development of the core modules has been developed fully.

Delivery, Installation and Maintenance Plan:

Since it is Windows / Android application so it need to be install. But it can be accessed or used by any user. User just needs to install it and use this Application.

1. **System Analysis**

**3.1 Users of System**

They can be any one in the world having Windows Desktop / an android phone.

User should be fully aware of the functionality of a smart android based phone. He should know how to install and use android applications.

User should be fully aware of the functionality of Windows operating system and also comfortable with desktop application that how to install and how to use it.

**3.2 Requirements of System**

**3.2.1 Functional Requirement**

When we will install the application after that it will ask two things

1. New Game
2. Quit

**After playing first time this we will have three options**

1. Continue
2. New Game
3. Quit

**Timer**

To increase difficulty level of this game we had include 2 more features. In that one of them is Timer. Timer should be of 60 sec in each level.

**Token**

To increase difficulty level of this game we had include 2 more features. In that one of them is Token. Token should be have created randomly in each level. Player should collect possible tokens of that level of game.

* + 1. **Nonfunctional Requirement**
* Device will have enough space to install this game.
* Device must have enough RAM to support this application quickly and mobile battery.
* Device must should be good graphics to support good quality of every level

**3.3 Feasibility Study**

**3.3.1 Technical Feasibility**

* All Graphics Design work done in Blender / Maya Designing Tool.
* Imported all graphics in Unity Project using Default library of Unity which is called Accents.
* C# scripting language used in backend to control everything after designing.
* Rest of the job is possible in Unity 3D.

**3.3.2 Economical Feasibility**

As it is a software its cost would be less

* + - The tanning and development in android 5000 per person.
    - Google play store charges $25.

**3.3.3 Operational Feasibility**

* + - We have tested our product on 23 different Windows desktops & android phones and its response is good
    - There is quiet a possibility that it may support all android phones worldwide. But we cannot be sure about it

**Game development and production**

**Game production**

Game development is a process where a group of various assets is crafted to a playable game. These assets can be code, graphics, audio, script, etc. Because of the amount of assets and the work the game usually needs, games are not developed alone but in groups. This group can be a small group of indie developers, a massive game company or anything between.

In addition to the traditional game development, there is a phenomenon called indie game movement. Indie is a term to describe a game development process executed with a small team without any notable funding. Usually indie developers get their funding via crowd funding.

**Game production phases**

Game production process itself can be divided into four phases: pre-production, production, testing and post-production. Some games have multiple production cycles instead of one. (Chandler 2014. 4)

The figure below represents the cycles and the phases of game production.

Representation of game production cycles (Chandler 2014. 4.)

Pre-production is a critical phase which defines what the game is, how long it will take to make it, how many employees will be needed and the costs. As a rule of thumb, pre-production should require 10-25% of the total development time. Pre-production can also be broken down to a concept, game requirements and a game plan, which is a roadmap for finishing the game. (Chandler 2014. 5-6)

In the production phase the team can begin work on making the assets and code for the game. Although the production should go smoothly based on the planning in pre-production phase, there are usually some new features and assets that need to be added which were not planned. The main focus here is on content, code, tracking progress and completing tasks. (Chandler 2014. 9-10)

Testing phase is a phase where a game will be fully tested so that it works completely and without bugs. The testing phase can already be started in the 14 production phase, after a feature or asset is published in the build. In a beta phase, developers will be fixing the bugs found while testing. (Chandler 2014. 12-13)

The post-production is also called a wrap-up of the game. This is where the team can make points and notes for the projects in future and review the pros and cons of the project. The phase usually consists of learning from experience and archiving the plan. (Chandler 2014. 15)

**Roles of game development**

In typical game development a group or team is divided into various roles (Figure 4). There are numerous different possible roles depending on the project itself, however, the most usual roles can be placed under managers, designers, programmers and artists.

Representation of four main roles of a game development team.

Lead designer comes up with the game idea and develops it with other designers towards a full game by designing the mechanics, content, visual style etc. After the prototype they work closely with artists and programmers. There can be many types of designers, e.g. a level designer, game mechanic designer and visual art style designer.

Managers manage teams and the project itself. Project management in game development can be similar to actual software development processes. Their 15 job is to listen to the team, give tasks for them and report what has been accomplished. Producers can be listed in this group.

Programmers program the game by forming game mechanics and interaction with all the assets. The first task for them to do is to take the game design document and do the prototype of the game. After the prototype they listen to the designers and develop the game further.

Artists make game assets like graphics, music, scripts, sound effects, 3D models etc. Artists work closely with designers so they can produce assets which go along with the designed style.

**Unity3D**

**Introduction**

“Unity is a game development ecosystem: a powerful rendering engine fully integrated with a complete set of intuitive tools and rapid workflows to create interactive 3D and 2D content; easy multiplatform publishing; thousands of quality, ready-made assets in the Asset Store and a knowledge-sharing community.” (Unity3d.com website 2014)

At this point Unity3D has over 2.5 million users worldwide. Unity has an office here in Finland and it is located in Helsinki. (Public relations)

Unity has two main versions: Unity Free and Unity Pro. Unity Free is free to use as long as a company does not exceed 100,000$ annual gross revenues or budget. Unity Pro costs 1500$ and has more features than Free. Pro license is for one user only. There are also some add-ons for different 16 platforms which may cost extra. If more users are wanted, a team license can be purchased, which is cheaper than buying many single-user licenses. (Unity3D License Agreement 2013)

Unity has some competitors, for example Unreal Engine and Cry Engine.

**Unity3D as a game engine**

As a game engine Unity3D is somewhat different from other game engines. It has a really simple user interface which is easy to get to know even for a complete beginner. It has many integrated tools so developers might not need that many different programs.

Here is a screenshot of a Unity project.

Unity has an asset store, a place where the community can publish their assets, plugins or whatever game developers might need. This is a great way to outsource workload depending on your team. Scripting in Unity is uses JavaScript, C# or Boo for scripting and it comes with an integrated script editor called MonoDevelop; also user favorite editor can be used. It is advised to use C# because of its object-oriented nature. There are also some visual scripting tools in the asset store for those who really do not want to write any code.

Unity pro also has an extensive support for native code plugins, thus parts of the project can be written using C++, Objective C, etc. This can be used to extend the game mechanics or to extend Unity’s features.

For 3D graphics a 3D modeling software is needed. Unity can import some formats, however, other formats might need the modeling software to be installed to get the models imported. Unity can usually import more information from the files than just the mesh, e.g. height maps, animations, and textures can be imported.

Unity has a tool called Mecanim for making and adjusting animations for 3D models also in Unity. In addition Unity has a tool called Shuriken for creating diverse particle effects using colors, shapes, meshes etc. For terrains Unity has its own terrain editor.

Unity has great documentation on their website and the community forums are a useful place to get help and suggestions from other developers. Unity also provides a set of really useful video tutorials on their website with script references for every script language listed above.

**Example usage**

Unity can be used for many different styles of games, or even for things which are not game related. With Unity both 2D and 3D games can be developed, although it originally was meant for 3D games only. One issue where Unity3D performs really well is multiplatform publishing. At this point Unity can publish games to:

Windows

Mac

Linux

Android

IOS

Windows Phone 8

Blackberry

PS3

Xbox360

Wii

Windows Store Apps

Unity web player

Unity is as good in over-network multiplayer games as it is in single player games. Unity script library has an integrated server class for simple network server creation.

Unity can also be used to make website plugins to visualize 3D-models. If there is for example a web shop and a 3D model of a product, Unity web player plugin can be published for the customer to view the product in 3D space. Unity is also used in industrial visualizing of architecture and mechanical objects. Visually Unity is not as good as some of its competitors. When developing a game with as good graphics and visuals as possible, some of Unity’s competing engines might perform better.

**Unity3D influences in game development**

“Unity is a great choice for small studios and indie developers and with its large user base and community it allows everyone from newbies to seasoned developers to get help and share information quickly”, says Sue Blackman in her book. (Blackman, 2013. xxvi)

Unity with other easy low-budget game developing solutions has had a great effect on game development. These entry-level engines have made it possible to develop games in smaller teams with much smaller budgets. In the big game companies there can be enough employees so they can make their own game engine and produce almost everything they need for their games, but in smaller groups this is not possible. This is why Unity has gained such great popularity among the Indie developers.

With Unity time, workload and expenses can be saved compared to making the engine alone.

Among the Indie developers, it is very common to use as much free and open source assets as is possible and to outsource as much work as can be outsourced. Unity community helps here. There is much free or cheap content on the asset store. If there is a problem the solution can probably be found from Unity community forums or documentation.

PlayRaven is a good example of one of these groups. PlayRaven is a freshly founded game company with five founding members. They pursue to outsource as much as they can. If the team has, for example a graphic designer and he/she has not any work to do in the early stages of development and still has to paid that is wasted money. If graphics are outsourced, the developers can concentrate on core mechanics for example and buy the graphics just when they are needed. (Seppänen, 2014)

Unity is one of these free or cheap to use game development platforms which provide users with a good set of tools, great community and loads of assets to use allowing these small Indie teams to exist in the first place. The easy user interface and great community is why Unity is very popular among the game development beginners.



**3.4 UMLDiagrams**









1. **System Design**

**4.1 Data Dictionary**

**New Game**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr.No. | Collect | Data Type | Size | Constraints |
| Level | Token | Integer (Counter) | As per Difficulty level | Not Null |

**4.2 Table Relationship**

* There is one table and there is no connection between them
  1. **User Interface Design**
* We have used Windows / Android as an up design part
* We have kept different button for different purpose

1. **Implementation & Testing**

* Why Unity 3D?

Unity3D is a game engine which is greatly used now, especially in ‘Indie’ game development since it can be used cheaply or even free. Despite being made for game development, it can be used for many other purposes as well, e.g. for example visualizing architecture in 3D environment, using Oculus Rift (virtual reality glasses) to experience an artificial 3D reality.

It is also an ideal engine to teach game development process since it speeds it up greatly and does not require a team with so many programmers. This is a way for a team can get familiar with different roles and processes of a typical game project.

**Game industry**

**Globally**

Game Industry has been the fastest growing branch of entertainment industry through 21st century. Game industry is already bigger than music industry, and it might be catching movie industry soon. (Industry Info 2014)

This year the global market revenues are estimated to grow to 70.4 billion dollars which represent an increase of 6%. This year the number of gamers is estimated to go over 1.2 billion. Smart phone and tablet games grow 35% this year easily outpacing any other studied segment. (Global Games Market Report Infographics 2013)

The Entertainment Software Association published a study they carried out on people of the United States about their relationship with computer and video games. Some findings can be seen below (Figure 1).

Figure 1. Gamer statistics from ESA study

(Essential facts about the computer and video game industry 2013)

IGDA, International Game Developers Association is the largest non-profit membership organization for game developers worldwide. It is a network for collaborative projects and communities the main goals of which are to develop game industry, advance the careers and to enhance the lives of game developers. (International Game Developers Association website) 11

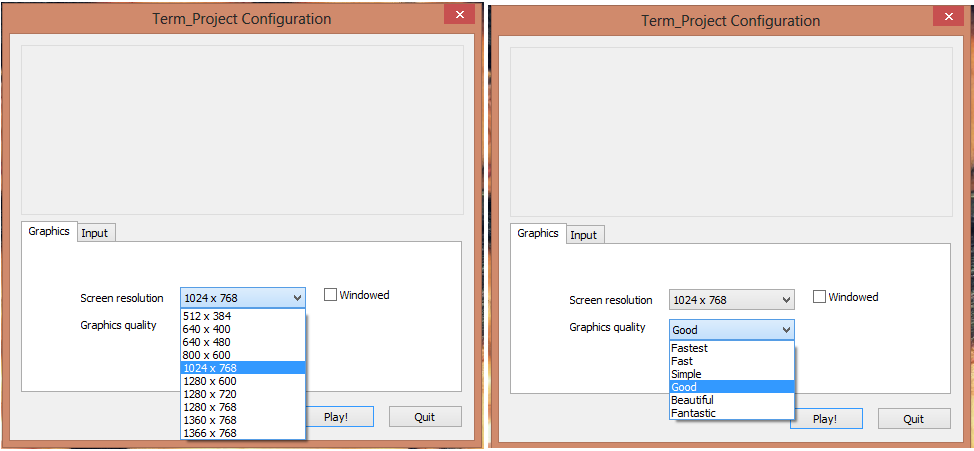
**In India**

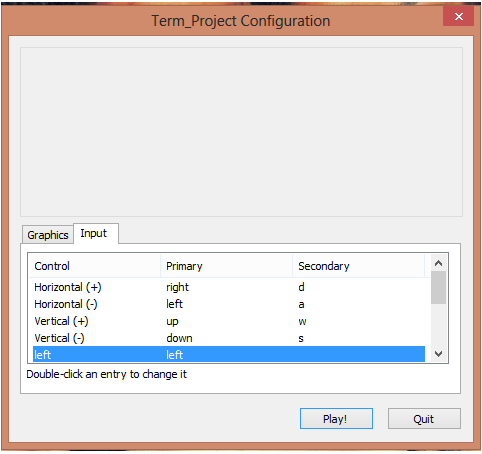
The game industry in India has been growing a great deal from year 2000 to this day. Because of the small market here in India, games have become a great import for India. (Industry Info 2014)

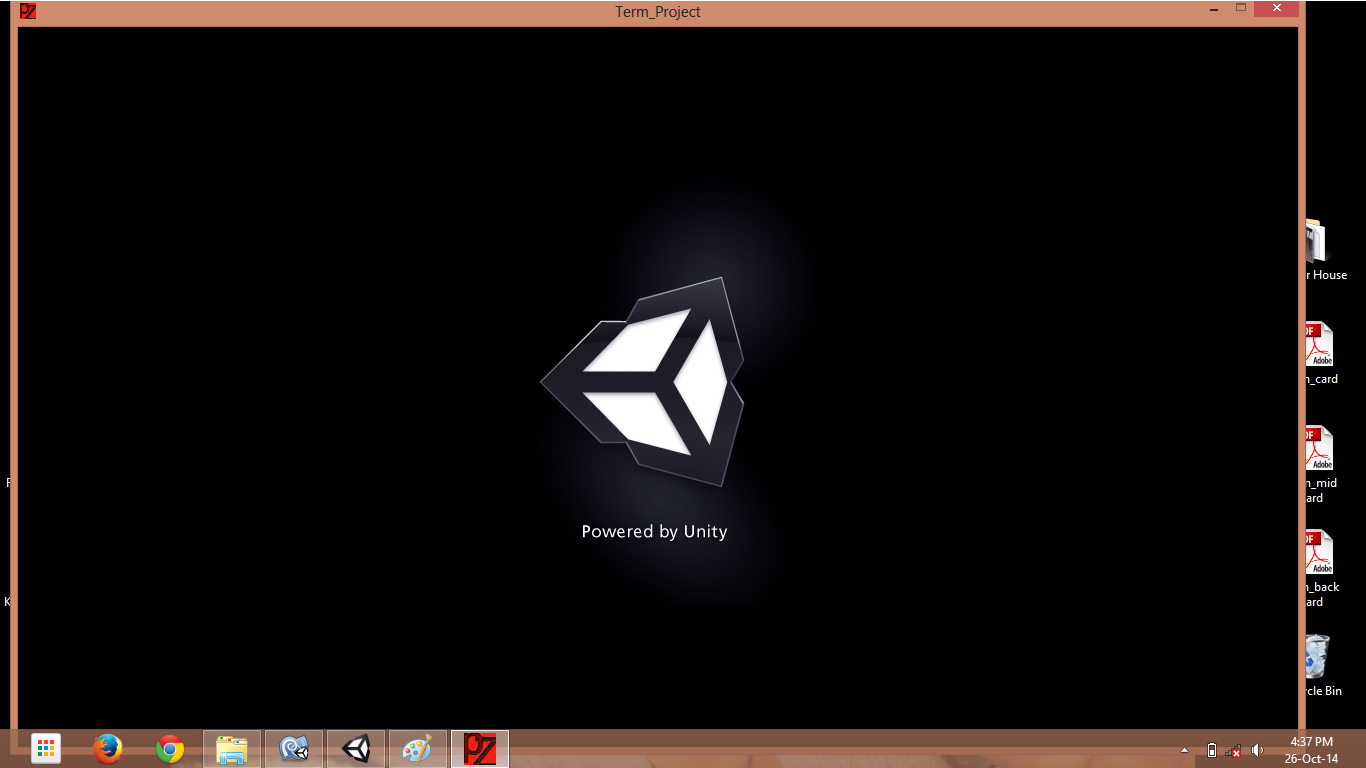
In 2012 there were more than 1,500 employees in the game industry in Finland from India and the average salary in 2012 was 3590 euros. (The Game Industry in Finland 2012)

**5.1.1 GUI of Form and Reports**

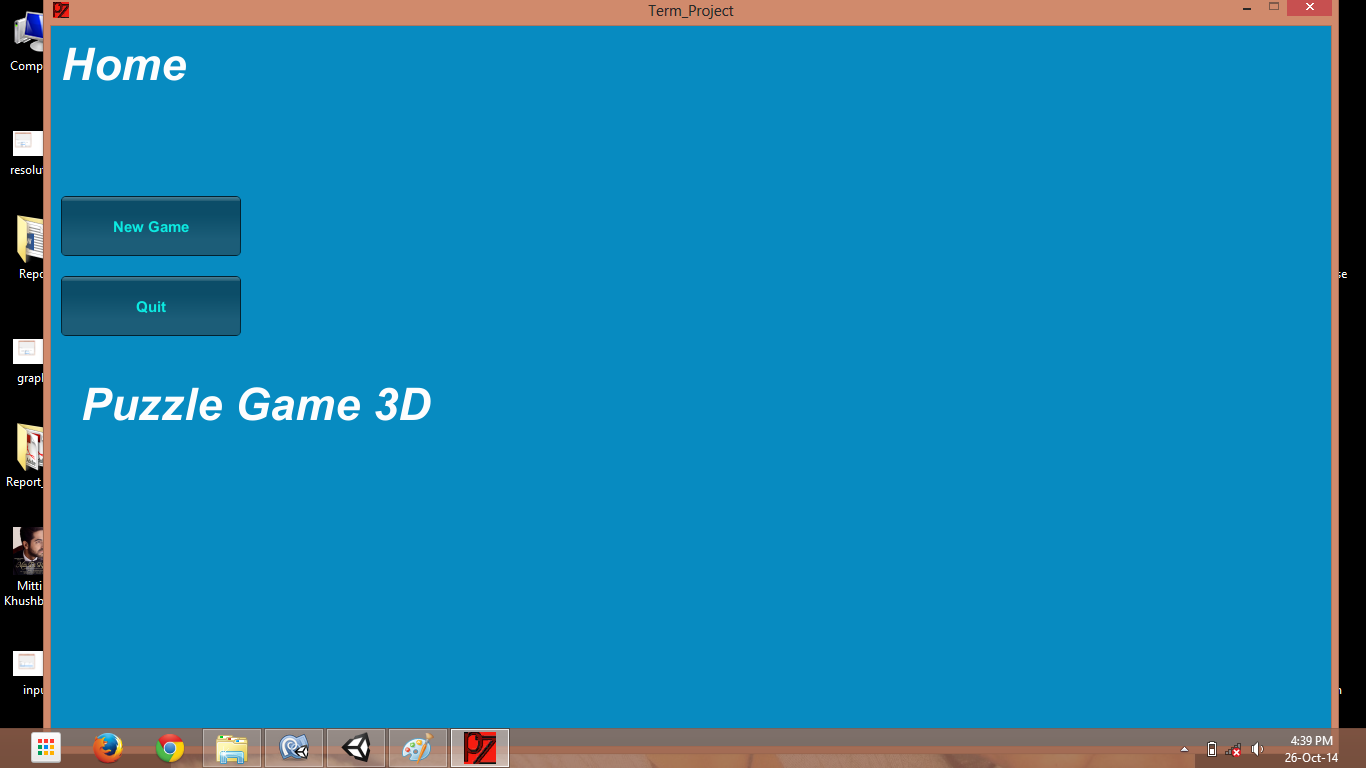
**Settings Window (Screen Resolution / Graphic Quality)**

****

**Input Setting Control Window**

**Splash page (Developer’s page)**

**Main Menu (HOME)**

****

**New Game (Level 1)**

****

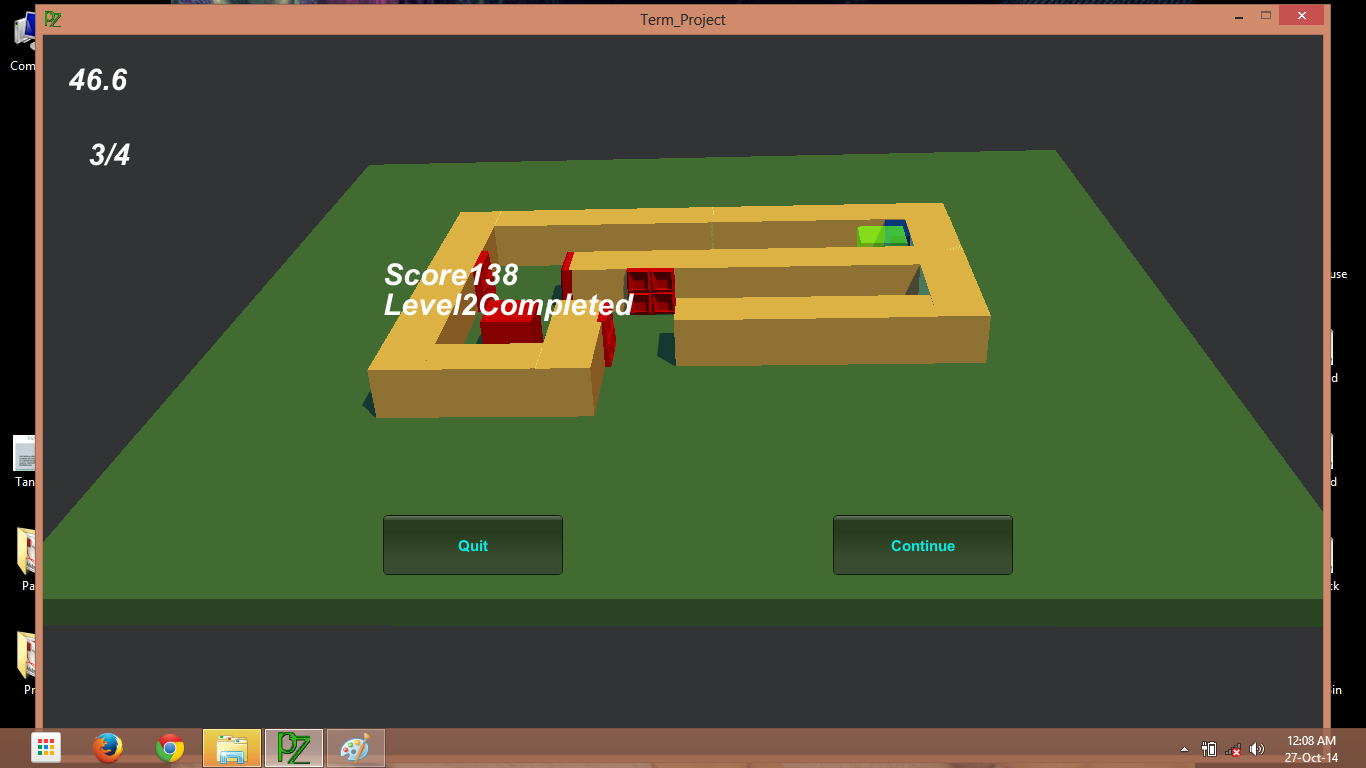
**Destroy Player Because of Spikes / Enemy (Object)**

**Distortion of object because enemy object using Animation**

**Load Next Level (Display Current Level / Score) Collected Tokens**



**Load Next Level (Level 2)**

****

**Load Next Level(Level 3)**

**5.1.2 Test Cases of System**

**Test Suites No: 1**

Test Suite Detail:

**Collect incomplete Tokens**

Valid score board saved in data on game (internal storage).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Function Name** | **Input** | **Expected Output** | **Actual Output** | **Pass/Fail** |
| **1** | Timer | Start Time = 40sec  Current Time = 26sec | Depends on player how much he / she expected Time for completing level  Max time expected = 39sec | Actual time for completing the level = 26sec | pass |
| **2** | Token | Total Token at level = 2  Collected Tokens = 1 | Depends on player how many he can collect in the defined time for level | Collected tokens in given time = 1 | pass |
| **3** | Current Level | Previous Level = 0  Current Level = 1  Next Level = 2 | Current level number should be the same as what Level number player will playing at. | 1 level completed  Go to next level | Pass |
| **4** | Current Level Score | Completed Level = 1  Current Level = 2  Level Score = 26 | Collected tokens \* current time = score  So that as per level 1 the maximum score should be 80 | Score = 26  1 level completed | pass |

**6.0 Conclusion, Limitation & Future Enhancement**

**Conclusion:**

* This Windows / Android Game (application) will serve much entertaining game in your hand. It will be free of cost. It will be usefully to different people in different ways. So more users will download it and use it all over the world.

**Limitation:**

* We should have an android smart phone and the knowledge how to install and use application.
* We should have appropriate RAM, Graphics and internal storage to play this game.

**Future Enhancement**

* Platform independent using Unity Project 3D.
* Freely available to all users.