1. **Title Page**
   1. **Game Name**

The name has originated from the word that is being used to describe eccentric or non-mainstreamed people. The term ‘Geeks’ has been constantly used to label people who are technically knowledgeable and is often seen to be operating with a computer.

Bugs, on the other hand is a jargon term used by software professionals to point an unnecessary issues or errors within a software. Bugs are parts of a software program that is not properly written according to the programmer’s purpose. But in this game, bugs are portrayed in a typical form—insect.

As ‘Geeks’ is a label used for IT literate people—mostly programmers, and ‘Bugs’ as their all-time antagonists, this game will revolve around the conflict between the two entities. It involves puzzle to locate bugs and remove it from the hardware devices within the computer. The player, as the geek, will be entitled to pick from two Nanobots—a robot with a size of an insect. The player will be able to control the Nanobot through encoding a single thread of instruction, which will be executed once.

1. **Game Overview**
   1. **Game Concept**

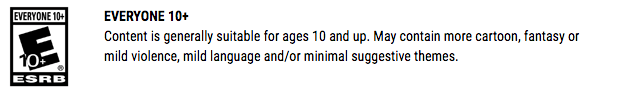
Geeks Vs Bugs is an educational puzzle game that aims to capture the attention of its audience but with underlying purpose of introducing basic computer programming concepts to the players. The story of the game revolves around the story of a computer error and bug that will be posing threats in the future—year 3000AD.

* 1. **Genre**

Geeks VS Bugs is a multiple genre game. It is an educational video game that uses the method of puzzle to stimulate logic and challenge to the player.

* 1. **Target Audience**

The game is recommended and designed for the needs of ages 12–22 years old but holds a rating of “Everyone 10+”. This rating follows the standard of “Entertainment Software Rating Board” (ESRB).



* 1. **Game Flow Summary**

To play the game, the player—as the geek who controls the robot—has to encode instructions to a thread. After compiling, the game will first check if all encoded values are logically possible. If, in any case, the game detected a logical error within the game the Nanobot will refuse to execute the given set of instructions and require the gamer to check and recode the thread of commands. The goal of the Nanobot is to locate and remove the bugs that are in the computer hardware. In most instances, the Nanobot will be required to take a necessary tool that in the map to accomplish it’s mission.

* 1. **Look and feel**

The game is design in a futuristic theme as for the story has a connection with the future. It applies 3D design for the characters and game world while 2D was used for buttons and other graphics.

1. **Gameplay and Mechanics**
   1. **Gameplay**
      1. **Game progression**

The game consists of 4 stages with 6 levels each. Each level requires the player to be able to accomplish tasks though providing instructions to the character—the Nanobot. After accomplishing the tasks, the player will be allowed to proceed to the next level. Each stage requires a minimum number of medals that are being awarded by each level depending on the player’s acquired score.

* + 1. **Mission/Challenge Structure**

The game has various missions and challenges.

In early stages and levels, the player’s mission is to program the Nanobot to reach certain spots from the map. As the player’s level raise, the mission changes to finding tools or locating and killing bugs. The missions seem simple but the game world’s maze and the method of programming the Nanobot makes complex and challenging.

* + 1. **Puzzle Structure**

There are two kinds of puzzle in the game. First is the maze of the game world, second is the combination of commands the player would use in order to accomplish its mission.

* + 1. **Objectives**

The main objective of the game is to remove the bug on the computer components by moving the character using the set of commands provided by the player.

* + 1. **Play flow**

The game flows with the succession of levels and stages. Each level may start with the explanation of new element for the player. Introduction of level’s mission will then be presented. After the mission has been laid out, the player will be allowed to input combinations of commands to program the Nanobot’s travel within the map’s maze. Then when the player is already satisfied with his provided instructions, they will compile the instructions. The game will then check for possible logical errors that the compiled commands may result to. If there are no errors with the commands, the robot will then execute the instruction to accomplish its mission. If the Nanobot was able to accomplish its mission, the score will be computed and the equivalent number of medals will be awarded to the player and next level will be unlocked.

* 1. **Mechanics**

The game will begin with only one level unlocked.

The player will encode instructions to the Nanobot and compile it. After compilation, the Nanobot will automatically execute the commands and travel within thin the game map.

Each level requires the previous level’s mission to be completed. After completing the mission, next level will be unlocked.

Stages on the other hand will be unlocked only with meeting the minimum number of medals. The player could play previous accomplished levels in order to gain their missing medals.

Scores will vary on steps, commands and errors number.

Medals will be awarded each level according to its equivalent value in the score.

* 1. **Replaying and Saving**

Replay of the current level will be accessible only right after the Nanobot completed the execution of commands. Previous performance aside from the current will not be accessible.

Levels and stages that are completed will still be accessible for the player anytime within the game application.

The game could be saved within the mobile’s local memory. Saving game will only save the current Nanobot’s status—level, stage, score and number of medals, character attribute and Nanobot name.

1. **Story, Setting, and Character**
   1. **Story and Narrative**

The story starts in year 3000 when computers are most advanced ever. But users suddenly experienced an error with their computer system and no one in their time is able to fix it. The person who can only fix this bug is a geek from the past. Fortunately, in their time, the time machine was already invented. Their most knowledgeable IT engineer used a time machine in order to get that geek person from the past. He brought along with him the most advanced robot they have—Nanobot.

The Nanobot has the ability to learn new instructions that could be combined with his previous ones. Its size is about an inch that the IT engineer was able to shove it inside his pocket.

After his travel through time, the IT engineer arrived to his destination—year 2016. He was inside a classic computer laboratory with computers that are flat and their computer units are so spacious. He then found a middle-aged man. The man is wearing a wide, thick glasses and his hair was like a bird nest. He remembered that man from the pictures in there times most successful company. He then realizes that the man is the one who invented the computer technology from year 3000.

He knew this was the man he was looking for. He hurriedly approached the man and introduced himself. He explained the tragedy of their time. Understandingly, the man did not refuse to go with him to the future without wasting anytime.

After arriving back in year 3000, he introduced the device that they suspect to be from where the error first emerged. He also introduced the man the Nanobot that would help him solve the problem. Living the man and the doctor alone in the laboratory he trusted them the future of the future world.

* 1. **Game World**
     1. **General look and feel of world**
     2. **(Areas)**

**Background:** the background has been design resembling from the internal look of a circuit board

**Game worlds:** All four stages have different game worlds

**Sage 1:**

**Sage 2:**

**Sage 3:**

**Sage 4:**

* + 1. **(Characters)**

**Male:**

**Female:Levels**

* 1. **Stages**
  2. **Training Level**

1. **Interface**
   1. **Visual System**
   2. **Control System**
   3. **Audio, Music, sound effects**
   4. **Help System**
2. **Artificial Intelligence**
   1. **Opponent and Enemy AI**
3. **Technical**
   1. **Target Hardware**

**Android Device with 4.3 Operating System Versions**

* 1. **Developmental hardware and software**
* **Hardware**
* Laptop
* Mouse
* External Hard drive
* **Software**
* Unity 5
* Blender
* Cinema4d
* Adobe Photoshop CS3
  1. **Network requirements**

No network requirements used nor needed.

1. **Game Art**