Requirements Document

Sam Prettyman | Chad Huckvale | Nathaniel Mason | Jaymes Bunce

**Title:**

Get Out

**Description:**

This adventure takes place in an abandoned warehouse in the middle of Central America. User wakes up in a dark room, dazed, confused, and unaware how they got there. The objective is to search the area and find a way to GET OUT. The user will need to locate objects and other victims in order to escape. Each victim has strengths and weaknesses that will be needed to overcome specific challenges.

Along the way there will be a variety of obstacles including (but not limited to) locked doors, puzzles. In order to defeat your enemies and overcome the obstacles, the user will need to invite others to join their party and these people will be crucial to their survival.

**End User Stories**

* Start the program
* Main menu
* Getting help
* Map
* Game Menu
* Move to a new location
* Explore a location
* View Inventory
* Tools/Weapons
* Doors
* Door #1
* Room #1
* Door #2
* Room #2
* Door #3
* Room #3
* Door #4
* Room #4
* Door #5
* Use Item
* Save the game
* Restart existing game

**Description of end user stories**

**Start program**

The end user enters the command to start the program. The computer displays a banner screen with a short description of the game. The player is prompted for and enters their name. The user enters their name and the computer displays a personalize welcome message and the Main Menu. An error message is displayed if an invalid name is entered and the player is prompted to re-enter a valid name or quit.

**Main Menu**

Display the following menu

N - Start new game

R - Load saved game

H - Get help on how to play the game

E - Exit

The end user (player) enters the selected item. The computer then displays the selected end user story. If an invalid menu item is entered, the computer displays an error message and redisplays the main menu. The program terminates when Exit is selected.

**Getting Help**

The following menu is displayed.

G - What is the goal of the game?

M - How to move

D - Doors open

T - Tools

Q - Quit

The user selects one of the options and the appropriate help message is displayed. If Quit is selected, either the Main or Game Menu is displayed depending on which one was previous displayed before this Hemp Menu was displayed. An error message is displayed and the menu redisplayed when and invalid menu item is selected.

**Game Menu**

**After starting the game it will pull up the map and at the bottom will be the game menu.**

V - View Map

M - Move Player

I - View Inventory

T - View list of tools

E - Explore Location

S - Save Game

H - Help

X - Exit game menus

Q - Quit

**Map**

Open the game menu and press M to view Map. The Map is 12x12. The map will display and show areas visited. It will show walls and when you enter rooms display objects in the room that can be searched.

**View inventory**

All of the tools and keys collected by the user on their journey will be available to view on this page. The keys will be separated from the other tools. Once a tool is used for everything it is needed for, it will disappear from this page. The user will be prompted with the message “Which item would you like to inspect?” The user then enters the number or letter associated with the item they would like to inspect. If the user enters an invalid number or letter, the message “this is an invalid value” will appear. When the user enters a valid value, they are able to view the object. A brief description will appear. For example, when the user chooses to examine the 2-LITER BOTTLE, the description “A 2-Liter bottle. The label has been removed but from its green-tinted color, you can only assume that it was Sprite or Mountain Dew.”

**Tools**

Open the game menu press T to view tools. To select the tools press the correct letter.

H - Hammer

P - Drift pin

L - Lock pick

T - Two Liter Bottle

W - Sand

E - Exit

**Doors**

There are seven doors that you must unlock using a series of equations, items or keys that are found throughout the building in order to escape the building. Each door you visit will prompt you to ask would you like to open the door. It will either say it is unlocked or tell you the door is locked. If it is locked it will display what you need to do to unlock. Example please find a key, or enter the pin, etc.

**Room #1**

**The user starts in a room. He sees a door at the end of the room. It prompts the user would you like to look around or try the door. If you select to try the door you will approach the door. If the user selects to look around he will notice an end table with an empty bottle and a bag of sand by it. It will prompt the user would you like to take the bottle. If yes then it is placed in inventory and if no it will say you do not pick up the bottle. It will then prompt the user would you like to take the bag of sand. If yes it is placed in inventory and if no it will say you leave the bag of sand. Then it will prompt the user to approach the door. If the user enter an invalid response like the letter ‘A’ then in says it is an invalid response. Please enter a valid value.**

**Door # 1**

The user will approach the door and be prompted on if he wants to open it. If the answer is yes it door will be unlocked and prompt the user to enter the area and if the answer is no it will say the door is locked and the user will stay in the room. **If the user enter an invalid response like the letter ‘A’ then in says it invalid response. Please enter a valid value.**

The user needs to find the 2-Liter bottle in this room in order to open Door # 2.

**Door # 2**

There is writing on the wall that reads a random value generated by the system followed by “lbs”. In this example, the number will be “20 lbs”. The user will see the writing with a scale underneath. The dial that measures the weight on the scale is rusted and broken. Next to the door are bags of sand. The user will be prompted with the message “Place the correct amount of sand on the scale”. Using the 2-liter bottle already acquired, the user must determine how much sand to put on the scale, converting liters to pounds. The user is unable to place sand on the scale without the bottle. They will use the formula 1 liter = 2.20 lbs. The correct value needed to be entered is 9.091 liters. The user will be prompted to enter how many times they will need to fill the bottle. The correct number of liters is 9.091 liters, however since the bottle is 2 liters, the user must fill up the bottle and place the sand on the scale 4.545 times. When the user inputs the correct value, the message “This is the correct amount of sand. The door is unlocked” will appear. When the user enters an incorrect value, the message “This is not the correct amount of sand” appears.

**Room #2**

The user sees a bookshelf and a couch. The user is prompted with the message “What item would you like to inspect?” At the couch, the user is prompted with the message “You have searched under the couch and under and inside the cushions. Inside one of the cushions you find what looks like a blue key, possibly crucial to your escape. You continue to search the couch but find nothing of note. BLUE KEY is added to your INVENTORY.” If the user chooses to inspect the bookshelf the system responds with the message “There are several books on the bookshelf and you begin to flip through them. After a few books, you notice that a slip of paper falls out. Written on the paper is the equation “(10 / 2 + 41)^2” You continue to search the other books around the shelves themselves but find nothing of note. PAPER SLIP has been added to your INVENTORY.”

**Door #3**

**The player comes to a door with a keypad to the right if the door. The user will be prompted to open the door or leave. If they choose to open the door it will displayed locked on the keypad. The keypad has digits 0-9 and is waiting for the user’s input. If the player has explored and found the clue with the equation on it will ask, “Would you like to view your clues?” If the user types yes it will display the clue from the piece of paper, “x= y/2 + 32 x^2≅????” The program will create a random, even value for “y” that will be between 0 to 134. This will be created at the beginning of the game. Then the user will be prompted if they would like to exit from viewing the note. Then the player will be asked if they would like to enter into the keypad. If the end user does not enter correct answer of the equation or enters letters, the keypad outputs an error and requests for the code again. If the end user succeeds then they door unlocks and they can enter Room #3.**

**\_ \_ \_ \_**

**| 1 | 2 | 3 |**

**| 4 | 5 | 6 |**

**| 7 | 8 | 9 |**

**| 0 |**

**Room #3**

**The user spots a pile of broken wood, an old refrigerator, and some cabinets. The player is prompted.**

**What item would you like to inspect:**

**B - The pile of broken wood**

**R - The old refrigerator**

**C - Cabinets**

**If the player makes an invalid choice it asks the prompt again. If the player inspects the pile they are prompted, “You found a rusty hammer.” If the player inspects the cabinets the player is prompted, “You found a drift pen.” If the player inspects the refrigerator the player will be prompted, “You found a mysterious red key.”**

**Door # 4**

**The screen will display a message informing them that they are at the the door # 4’s entrance. The user will be prompted to either open the door or continue without opening. If the user chooses to open the door a message will display saying the door is locked and needs to be opened by force. They will need to break the look then kick the door open. If the user has the hammer and the drift pin in their inventory they may attempt to open the door they will be asked to use them to open the door. A message will display stating that they need tools to open and they will be continue their journey. To use the tools to open the door the display message will say the need to use the hammer to hit the drift pin and break the lock. They will then be prompted to enter in the speed they want to swing the hammer. Force = Mass x Acceleration. The user will enter a rate of acceleration and the mass will be the weight of the hammer that is generated by the computer. The hammer mass will be randomly generated each time a user attempts to open the door. If their outcome is below 450 newtons, a message will display saying that they need more force. If the user has an output value higher to 650 newtons, a message will display saying that they need less force. A message will be displayed that reads the door is now unlocked but the door is jammed, kick the door to open when the players output value is within 450 and 650 newtons. Their weight and acceleration of their kick will be used to factor the force needed to open the door. They will then be prompted to enter their weight and the acceleration of their kick. If their force is below 2000 newtons a message will display saying they need more force. If their force is above or equal to 2000 newtons a message will be displayed saying the door is now open and they enter the room.**

**Room #4**

**The room will have a desk, a chair and a cabinet. The player will need to locate the green key which will be be inside a desk. The player will be able to explore the room. As they move around the room they will be notified when they encounter an obstacle. A message will appear that indicates the obstacle they have encountered. They will have a message display asking if they wish to explore the obstacle or continue walking around without exploring. If the choose to explore the cabinet they a message will display saying all the drawers are empt. When they approach the desk they will be prompted to open the drawer. If the do they will find the green key and a message will display stating that they have found the green key and it is vital for their escape. They may leave the room whenever and the new location will appear on the map.**

**Door #5**

The user will approach the door. It will prompt the user would you like to open the door. If the answer is yes it will then prompt the user to insert the three keys, red, blue and green. Then it will prompt the user to insert the red key press R, to insert blue key press B, to insert the green key press G. If you try to enter the key and you don’t have the key it will prompt the user go and find the key. If you have the keys and insert them it will prompt the user to open the door and it will congratulate them on escaping. If the user approach the door. It will prompt the user would you like to open the door. If the answer is no then it will prompt the user to continue his search for the keyes. **If the user enter an invalid response like the letter ‘A’ then in says it invalid response. Please enter a valid value.**

**Use Item**

The following is an example of what could be displayed:

E- To use the blue key

The user is prompted in different circumstances to use a key. This will be when using keys on doors, solving puzzles and interacting with puzzle solving items. If an invalid value is entered the program asks to re-enter a value.

**Move to a new location**

End user enters in coordinates based on where they want to move their character. User will enter coordinates in this manner: (A, 1). The user can only go to locations that have been opened on the map. If they try to enter coordinates for a room or location that is unlocked or outside of the map, they will be unable to go there. For example, if the user enters coordinates that are inside of a locked room, the message “You move towards your destination, but at (X,X) there is a door that is locked preventing you from reaching the desired coordinates” will appear.

**Explore a location**

Explore location The computer displays a message describing the contents of the location and then the location is marked as visited. The map and game menus is then displayed with the contents of the location showing on the map.

**Save the game**

The user is prompted to enter the save location where the game will be saved. If a valid file location is specified, save the player, game, map, inventory items, and actor data to a file. Display a message saying that the game was saved successfully and then return back to the Main Menu. An error message will be displayed if the user enters and invalid file location and the user prompted again to enter a valid file location.

**Restart existing game**

The user is asked to enter the file location where the last is. If a valid save location is entered, read the player, game, map, inventory items, and actor data from the file. Set the current game to the game read in. Display a message saying that the save was retrieved and then display the Game Menu. An error message will be displayed if the user enters and invalid save location and the user prompted again to enter a save location.