**No One Left Behind**

**By; Brady Nelson and Daniel Shuss**

**Description:**

No One Left Behind is a text based RPG set to the following:

“You have found yourself at the edge of a fast wilderness on a dark and stormy night. The rain is not heavy yet, but the wind is picking up and as the thunder booms so near it almost knocks you over. The brilliant flash of lightning briefly reveals you are standing on the last step of a very large old mansion made from large jagged cut stone. Strange and eerie sounds emanate from the unhallowed mansion with a dreadful smell that envelops you in fear, but your dear old friend Abinadi has been missing, and a strange but most powerful feeling over powers your fear and compels you to enter the abyss to rescue Abinadi from whatever evil is holding him captive.”

The player is prompted to start a new game or continue an existing saved game. The player selects a choice. If continue game option is selected then the old game is read from a saved file. If no game has been saved then a message indicating that there is no saved game will be displayed. The player is then prompted to start a new game or return to the main menu. The Main Menu is redisplayed if that option was selected; otherwise, a new game is started and specified below.

If a new game was selected, the computer will:

* Prompt the player to enter their name and the name of the person most dear to them. If no both prompts are not completed the player is prompted to reenter valid names.
* Create and initialize all of the objects needed for the game.
* Set the player at the beginning location.

**List of end user stories**

* *Description*
* *Program Start*
* *Main Menu*
* *Help Menu*
* *Game Menu*
* *Start game*
* *Movement*
* *Battle System*
* *View Map* 
  + *Map Locations*
* *Inventory Menu*
* *Save Game*
* *Equations Puzzles*
* *GateKeeper*
* *Healing Pools*
* *View Weapons*
* *View Potions*
* *View Items*
* After Death/Game Over

**Description of end user stories:**

***Program Start***

The user will enter the command to begin the game. The game scenario is displayed explaining what lies ahead and then the main menu runs. There is no user input here.

***Main Menu***

The Menu View is displayed:

N – Start new game

L – Load existing game

H – View help files

R - Replay opening dialog

B - View battle system description

The player is prompted to start a new game or continue an existing saved game. The player selects a choice. If continue game option is selected then the old game is read from a saved file. If no game has been saved then a message indicating that there is no saved game will be displayed. The player is then prompted to start a new game or return to the main menu. The Main Menu is redisplayed if that option was selected; otherwise, a new game is started and specified below.

If a new game was selected, the computer will:

* Prompt the player to enter their name and the name of the person most dear to them. If no both prompts are not completed the player is prompted to reenter valid names.
* Create and initialize all of the objects needed for the game.

***Help Menu***

The Help Menu is displayed:

N – Restart and show game story again

D – View game control

I – View a description of in game items

B - View battle system description

If valid input is inputted, the program will go to the selected option. If invalid input is received, the screen will display a message telling the user to input a valid option.

***Game Menu***

The Game Menu is displayed:

M – View Map

I – View Inventory

E – View Equipment

W – Walk Forward

S – Save Game

H – View Help menu

Q – Quit

The user will make a selection and then the program will run that choice.

**Start game**

In order to begin playing, the player will have selected either ‘New Game’ or ‘Load Game’. If the player chooses ‘New Game’, the game will create the necessary game objects and officially begin the game by displaying a message to the user explaining the story, game objectives, and a list of some of the basic game controls. The message will read something like this:

“You find yourself inside the eerie mansion. There’s an ominous feeling about this place that makes you want to leave, but something else tells you that Abinadi is here and you’re determined not to leave without him. You don’t know where he is in this labyrinth of darkness, so be careful as you seek him out and prepare yourself for encounters with the unexpected and unholy. You get the feeling that any hostile creatures you may encounter will be susceptible to different weapons or attacks. Try to find their weaknesses by varying your weapon inventory and techniques in battle. Most importantly, find Abinadi and get out of there!

(Press ‘M’ for the menu and a list of game commands)”

If the player selects “M”, the game control menu from within the help menu will be displayed.

After which, the player will have full control to begin playing the game.

If the player selects ‘Load Game’, then the program will check to see if there is already a saved game on file. If there is not, then game will will inform the player “There is no save data available” and will return them to the main menu. If there is a save file, the game will load the most recently saved game data and allow the player to continue from where they left off. There will only be one save file to access so it will be done automatically without asking the player to select save data

**Movement**

Moving to new Locations:

The computer displays a message prompting the player by name to enter the desired direction and distance to move. The player enters the direction and distance If either of the desired options entered are invalid, a message is displayed to the user by name and they are prompted to re-enter the desired direction and distance.

When good directions are entered by player, the computer will move the player to the desired location, if any things blocks their path they would be moved directly in front of the object and a message is displayed revealing what the object is. If they are in imminent danger by the object they will receive a warning of the danger.

A map and game menu are then displayed.

When player enters a new room a new end story (with room contents) is displayed along with a new map so the player can explore the new location.

**Battle System**

The following options are displayed on the screen.

In the initial phase of the Startup, and again in the help menu, the user has the option to view the battle system description, which will display the below table.  It will give the player insight as to which kind of adversary he will encounter and which weapons or tactics may be used.

|  |  |  |
| --- | --- | --- |
| Enemy | Battle Scenario | Spell or Weapon |
| Vampire: | - Wants to be in the lower levels or innermost part of the  Mansion — away from sunlight.  - Limit to 3 encounters.  - Encounter results in an immediate battle or need to  escape until proper weapons are located. | Sun Light Spell; Wooden Stake; Silver Knuckles; Air Changer Spell (slows the Vampire down to human speed. Prevents it from changing shape); The Scythe |
| Zombie | - Can be anywhere: Limit to 3 encounters  - Initial Encounter will be hand to hand combat; you don’t  want to kill a Zombie if you don’t have to; they can  become your ally; during hand to hand combat you will  try to persuade or bring the Zombie back to the light. | Hand to Hand; Club; Shackle to Bind;  Knife; Shotgun |
| Ghost | - Can be anywhere – but are rare — limit to 2 encounters.  - Represents Fear – when you encounter a ghost it will  since your fear, i.e. heights, snakes, confinement etc. As  you overcome your fear or fears the Ghost is diminished  and will disappear. | Courage;  Fear scenarios will make available tools to  help overcome fear (snakes – stilts),  (confinement – escape lever) etc.  - Ghost encounters are an inception — a game  within a game.  Invisibility spell |
| Skeleton | - Can be anywhere – most common battle scenario.  - Limit to 10 encounters. Skeletons are easily defeated  with a blow from any object to the mid-section causing it  to break apart  - Are slow but do not get tired. | Metal Ax; Acid spell;  Any object that you can swing, including a side kick. |
| Empty Room or Hallway | No Battle Scenario, but once in a room, player must solve a puzzle or a question to find game helps or battle helps and to be able to progress to the next room. | Wits |

Any victory increases life force; Victory allows the player to explore the room unmolested; It enables the player to find clues and weapons and spells etc.

A - Attack the opponent (While in attack mode the below commands are enabled)

L - Low Strike

M - Middle Strike

H - High Strike

K - Kick (always middle strike but with greater force)

I - Invisibility Spell

T- Shot Gun Blast

J- Club Strike

P - Cast Spell

E - Escape (takes you back to your previous room) (reduces life force by 1)

R - Run for your life (Moves you randomly – computer decides (Math.random))

(reduces life force by 1)

S - Check for available spells

While in the “S” menu, you can select spells by choosing the spells name or number.

W - Check for available weapons

While in the “W” menu, you can select weapons by choosing the weapon name.

H - Review player’s health

F - First Aid – allows player to rest and recuperate (regain 1 life force)

Our hero needs to keep in mind that not all weapons can damage all monsters and that not all spells will cause the desire effect.

Weapons can run out of bullets so will have limited use until you can find more. Hand to Hand weapons do not run out. Spells are only useful 3 times per spell bottle.

If you select “R” to run for your life, you lose all weapons except the current weapon that you are using and 1 spell cast.

If you “E” escape then you lose 1 spell cast.

Only the right weapons or correct spell can get rid of the enemy!

The undead enemy can also attack the adventurous user and affect his health so it is important to always check the health and maybe use a recovery spell to gain strength again.

**View map**

This will show a map of the hallways and Rooms that the user has explored. Cleared rooms will be marked cleared. If there are rooms that the user found, but did not clear, they will show that they still need to be explored. This view will show only one option for the user to return back to the game menu. Rooms that have not been found will not be displayed until they have been found. The map will have over 25 locations once fully explored.

R – Return to game menu.

**Map Locations**

Most of the locations will be inside the haunted mansion but there will be some that might be on the mansion grounds instead of the mansion itself.   
1. Crypt

1. Crypt
2. Cemetery
3. Library
4. Storage room
5. Bedroom 1
6. Bedroom 2
7. Bathroom
8. Laboratory
9. Dungeon cell 1
10. Dungeon cell 2
11. Dungeon cell 3
12. Crematorium
13. Healing pool room 1
14. Healing pool room 2
15. Healing pool room 3
16. Hidden room (the one behind the scale puzzle door)
17. Kitchen
18. Great hall/entrance to mansion
19. Servants’ quarters
20. Art gallery
21. Attic
22. Garden
23. Morgue
24. Guest house
25. Observatory

**Inventory Menu**

Displays the items that the user has collected along his journey, as well as what he currently has equipped. Items that are single use, like potions can be used. Items that the user chooses to equip will swap, moving to the user, while the users old equipment will be moved into the users inventory. The condition of used items will also be shown.

W - View Weapons

P - View Potions

V - View Items

U – Use item

E – Equip item

R – Return to game menu

**View Weapons**

The user will enter a menu that will show a list of his weapons on the left side of the screen. He can select an item to view its description which will give hints to which monsters the weapon is likely to be strong or weak against.

D - See Description of a weapon

E - Back to Inventory menu

**View Potions**

The user will enter a menu that will show a list of his potions. He can select a potion to view its description.

D - See Description of a potion

E - Back to Inventory menu

**View Items**

The user will enter a menu that will show a list of all his random items collected throughout the game. He can select an item to view its description which will give hints to the possible use of each item throughout the game.

D - See Description of an item

E - Back to Inventory menu

**Save Game**

Save all the object conditions in the game and let the user know that the game was successfully saved. Then prompt the user to either return back to the game menu, or the main menu.

**After Death**

Players can die when encountering enemies in a battle. If the player’s HP is reduced to zero, the game will display a message saying, “You have died, but Abinadi *(or whatever the name of your lost friend ends up being)* still needs your help! What would you like to do?”

Below this message, the game will also present a menu containing different options, prompting the user to:

C - Continue from the last save point

Q - Quit the game and return to the main menu screen

X - Quit the game and exit the program entirely

**Locked Door**

One of the rooms will be locked behind a door that has no key. The room is filled with piles of rocks and a scale. The only way to unlock the door is to trigger the unlocking mechanism that is tied to the scale, which unlocks the door when the weight of right side of the scale is 75% of the left side of the scale (each brick will have an assigned weight of .25 pounds as an attribute of the object). The room can contain clues for the player to let them know what ratio is required. Clues can be something like “Harry Potter had to board the train at platform 9 and….?”  
  
When the user interacts with the scale, they will be met with messages that prompt them to add an amount of weight to each side of the scale. The messages will read:  
  
“How many bricks do you want to add to the left side of the scale?”  
  
“How many bricks do you want to add the to the right side of the scale?”

Example code:

double leftSide = input.nextDouble;

double rightSide = input.nextDouble;

if (rightSide / leftSide = .75) {

// unlock door

};

else {

// keep door unlocked and prompt player to try again or exit the scale puzzle

};

**GateKeeper**

Along the journey there will be certain doors that will be guarded by a mystical GateKeeper character that will reappear at random doors. He will require the user to gather certain metal items found throughout the game, which he will then magically change into new upgraded items for the hero. The GateKeeper will tell the hero that each metal weighs “x” amount, and that he needs a certain amount of weight total. The hero will have to decide how much of each type of metal to give the GateKeeper. If the wrong amount is entered, the GateKeeper disappears, but the door unlocks and the hero receives no bonus weapon. Equation will use the weight and amount of 2 different objects to determine which percentage of each is required to get a certain alloy.

**Healing Pools**

Certain rooms, probably every 10 rooms or so, will have a healing pool. Each pool will have an empty cup of a different size being held by a golden hand magically floating next to the pool. The hero will hear a voice from the pool that tells him how much water to fill the cup with to be healed. The cup will be labeled with a height and radius, and the user will have to figure out the volume. The hero can try as many times as needed, but can only heal at the pool once. Equation will be the volume of a cylinder.

|  |  |
| --- | --- |
| Requirement | Status |
| 1. The program must start by displaying a banner welcome screen that displays the name of your game and gives a brief description of the the game. The end user is then prompt inputs their name. The name is then saved. (One end user story required) | Completed |
| 2. The program must contain a main menu, a help menu, a game play menu and at least one one other menu. There must be end user stories for each menu item selected. (Four end user stories required: Main menu, Help Menu, Game menu and one other menu of your choice) | Completed  We have the Main Menu, Help menu, Game menu and Inventory Menu |
| 3. There be at least three different end user stories that require the player to enter input on the console where the computer calculates a complex equation (e.g., calculate the volume of a barrel). Be sure to list the complex equations for each of these end user stories. A complex equation is one that has at least three values and two different operators. (Three end user stories required - one for each scene or view that requires the calculation of a complex equation) | These are completed in Locked door, GateKeeper, and Healing Pools. |
| 4. The game must contain a two or three dimensional map or game board with different locations that the characters in the game may visit. The map or game board must contain at least 25 different locations. Each location will contain a different scene or setting in the game. The same scene may be used in the one or more locations. (One end user story required to display the map) | Do we need add the 25 location right now to the “View Map” user story???  \*\*update\*\* Completed under View Map → Map Locations |
| 5. The game must contain one end user stories to move the different actors or characters in the game from one location to another in the map or game board. (One end user story required to move a character) | Completed |
| 6. The game must contain three or more a list of items (e.g., resources, items, weapons, players, etc.) and separate end user stories to sort and display each of the list of items.(Three end user stories required to display the three list). | These are completed in the inventory item lists |
| 7. The game must contain end user stories to save a game and restart a saved game. (One end user story to save the game and one end user story to restart a saved game) | Completed |