Internal Design Document

https://docs.google.com/document/d/1fjH6mw\_cqWIS8YQyzE7Z4y0gyIuAGyT7CQPBr4iE-3s/edit

Team Javalava

David Laevsky, Cheng Liang Lin and ChristelleMarie Chua

Professor File, TA Josh Roschette

**Section 1 – Purpose**

This program functions as a text based video game that is coded in Java. The program is meant to fulfill the requirements for the ITI 202 final group assignment. Our game tells a story to the player by presenting them with a menu to its user and asking them to input different commands that correspond with different functions in the program. Each specific function in the program corresponds with a different object in the program and the specific objects are structured in a way that allows them to interact with each other to allow the user to solve puzzles in the game.

The structure of the game is linear and consists of 10 rooms, each with a distinct puzzle that the user must solve before progressing to the next area. The secondary purpose of this game is to entertain the user by making them think in an innovative manner. The main goal of the game is for the player, who is taking on the role of a child in a sweatshop, to escape their abhorrent working conditions and their evil employer. During the course of the game the player accidentally uncovers mysteries surrounding the factory and the conditions that other workers were facing through solving various puzzles, unravelling the sadistic nature of their employer.

**Section 2 - High Level Entities in the Design**

\*Note - the descriptions I am making for this and the following parts will be based off of the Api documentation in:

https://drive.google.com/open?id=0B8TZEjRBGZUaY2N1N21xOGxZNWM

or CURRENT PROJECT

and in The Finalized Demo that was uploaded to Github by Chris

**TextAdventure Class:**

This method contains the source code used to both initialize and close the game.

**AdventureModel Class:**

This class will contain a tutorial for the user that introduces them to the text bsed syntax commands used in the game as well as all of the source code for room and area descriptions in each of the rooms. The code that allows the user to “use” items on objects within the game map will also be in this class.

**Adventurer Class:**

The Adventurer class will contain the user's position within the game map, in addition to an inventory of the items that the player has collected throughout the duration of the game.

**Map Class:**

This class contains the source code for all 10 of the rooms used in the game. Included in this class are also the puzzles that the player will need to solve in order to progress. Both the system output that the player will see and the internal operational code will be housed in this class

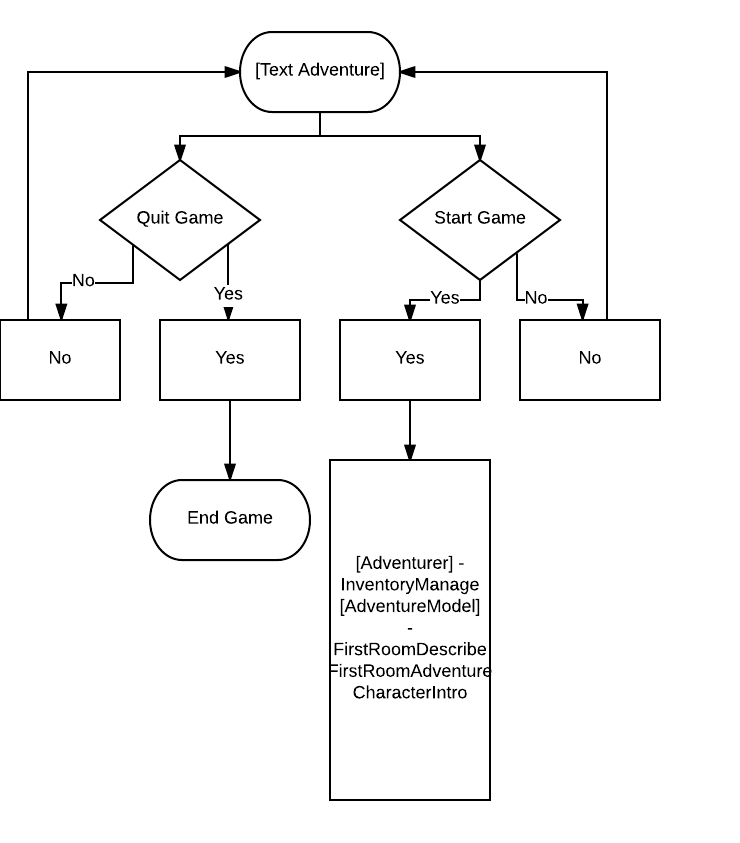
**Section 3 - Low Level Design**

**TextAdventure Class:**

**Usage -**

This method will allow the user to initialize the game as well as close it. The QuitGame class can be called on by other methods.

**Model -**



**Interaction -**

**This class interacts with the Adventurer method through the Start Game function. The user is also linked to this method through the menu function in the Adventurer method when they want to close the game.**

**AdventureModel Class:**

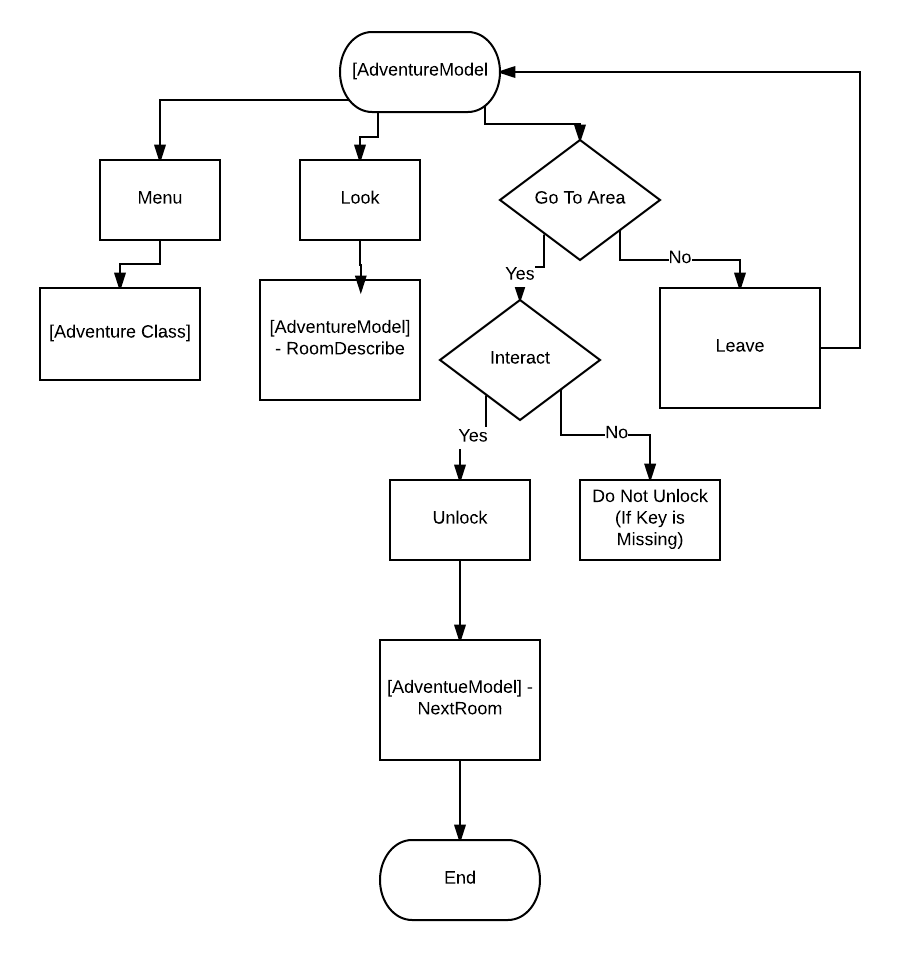
**Usage -**

**To house all of the map, and item objects that will be used in the game. These objects will be called upon by the player as they are navigating through the game from the menu in the Adventurer class**

**Configuration -**

**This Class does not need any sort of special configuration since it is made up of a bunch of smaller objects that are called upon throughout the course of the game.**

**Model -**



**Interaction -**

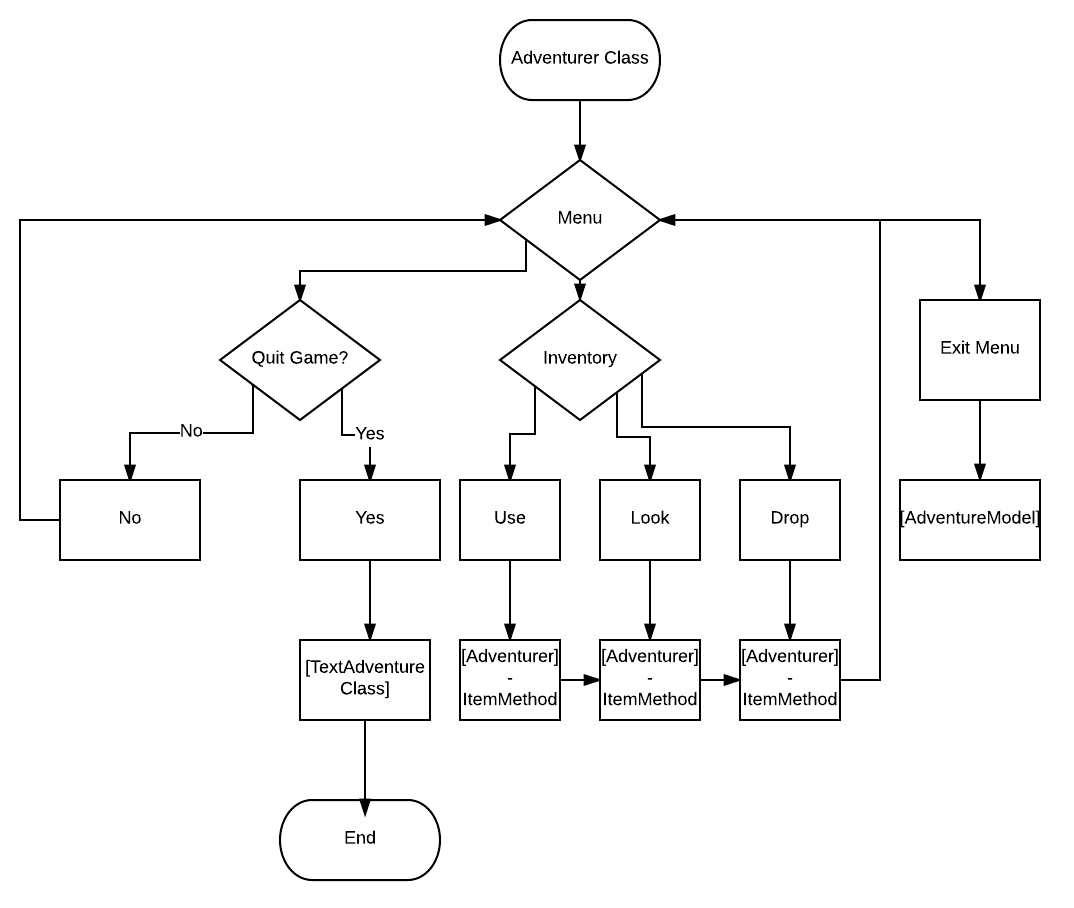
**This class interacts with the Adventurer and the Adventure model Classes**

**Adventurer Class:**

**Usage -**

**The Adventurer class contains the user's position in the game as well as the items that they have collected. They can view their inventory, quit the game and exit the menu from this object**

**Model -**



**Interaction - The Adventurer class interacts with the TextAdventure method when closing the game. It also interacts with the AdventureModel class when the user is ready to close the menu**

**Section 4 – Methods**

**A: TextAdventure Class**

Handles opening and closing the game and other miscellaneous functions.

* 1. startGame()
     1. introduces game title
     2. asks user to press A to start the game
     3. (TBA) asks user if they wish to close the program without playing

/\*\*

\*@ constructor Public Method

\*@param Non

\*@notes method to start game

\*/

* 1. quitGame()
     1. asks user if they wish to quit game (YES/NO)
     2. if YES, close program.
     3. if NO, return user to MENU. C1.

/\*\*

\*@ constructor Public Method

\*@param Non

\*@notes method to end game

\*/

**B: AdventureModel Class**

**/\*\* @constructor Public Class**

**\*@param characterintro, firstroomdescribe, firstroomadventure**

**\*@notes**

Contains information and descriptions of the room. User can implement actions / move here.

* + 1. CharacterIntro()
       1. Prints an introduction to the game to give context to the user. Only prints once.

/\*\*

\*@ constructor Public Method

\*@param Non

\*@notes method for background info for user

\*/

* + 1. FirstRoomDescribe()
       1. Prints description of the first room.
       2. Directs the user to B3.

/\*\*

\*@ constructor Public Method

\*@param Non

\*@notes method to describe the setting of room 1

\*/

* + 1. FirstRoomAdventure()
       1. Asks the user what they want to do in the main area of the first room
       2. Can open menu C1.
       3. Can look B1.
       4. Can go to desk B4.
       5. Can go to door B7.

/\*\*

\*@ constructor Public Method

\*@param Non

\*@notes method to prompt user for actions within the room 1

\*/

* + 1. DeskDescribe()
       1. Prints description of desk
       2. Directs the user to

/\*\*

\*@ constructor Public Method

\*@param Non

\*@notes method to print the desk information, keys, objects etc

\*/

* + 1. FirstRoomDesk()
       1. Ask user what they want to do by the desk
       2. Can open menu C1.
       3. Can look B4.
       4. Can open drawer B6.
       5. Can go back B3.

/\*\*

\*@ constructor Public Method

\*@param Non

\*@notes method to describe the setting of room 1

\*/

* + 1. DeskDrawer()
       1. IF user has the key, print “There’s nothing here”
          1. return to B5.
       2. Else print description of the desk with the key
          1. User can look B6.
          2. User open menu C1.
          3. Take the key (add key to itemList)
          4. Go back B5

/\*\*

\*@ constructor Public Method

\*@param Non

\*@notes method try catch incorrect uses for the object

\*/

* + 1. DoorDescribe()
       1. Describe the area by the door
       2. Directs the user to B8
    2. FirstRoomDoor()
       1. Asks the user what they want to do by the door
          1. User can look B7
          2. User can go back B3
          3. User can open door

IF the user has the key, the door can be opened

ELSE the door cannot be opened.

* + - * 1. User can open menu C1.

**C: Adventurer Class**

Contains ArrayList itemList, aka the inventory

* 1. MENU
     1. Takes the user’s origin as a parameter
     2. User can look at inventory - C3
     3. User can quit game - A2
     4. User can close the menu
        1. returns user to its origin
  2. inventoryManage()
     1. adds starting items to the inventory
  3. INVENTORY
     1. Takes the user’s origin as a parameter
     2. Prints the current items in user’s inventory
        1. if nothing in inventory, closes inventory
           1. returns user to origin.
     3. Allows user to choose an item to look at
        1. item X (all items will have similar behaviors)
  4. itemXMethod
     1. Takes the user’s origin as a parameter
     2. User can look at item
        1. Prints a description of the item.
        2. Returns user to C1
        3. User can use item
           1. If user’s origin corresponds to the item’s use, then item can be used and return to B - current position.
           2. else “You can’t use that here” and return to C4.
     3. User can drop the item
        1. if item will still be used, print, “You decide not to drop it”
           1. return to C4.
        2. else drop it
           1. return to C4.

**Command List**

Look - Allows the player to examine either a place or an item

1. Door - The player examines a door before either using a key or the Go function
2. Key - Examines the key. Gives a hint on where it might be used in the game
3. Gunpowder - Player examines the gunpowder and is presented with a hint on using the lighter on it.
4. Right - checks surroundings to the right
5. Left - Checks surroundings to the left
6. Cobwebs - Uncovers secret passageway in Room 9 (Shed)
7. Window - Player examines window and is presented with a hint on using the rope
8. Conveyor Belt - lets the player examine the conveyor belt in Room 3
9. Yarn - Examines the yarn in the Room 4 (Textile) room

Use - Allows the player to interact with an item

1. Key - Opens doors and chests throughout the game
2. Ladder - Allows the user to exit a multistory room
3. Wall - Lets the user uncover a passage that leads to a different room
4. Door - Opens a door
5. Rope - Used to climb out of windows
6. Lighter - Sets things on fire

Drop - Removes item from inventory. Item will respawn at the original location that it was found

1. Key - Drops the key
2. Gunpowder - Drops gunpowder
3. Rope - Drops rope
4. Lighter - Drops lighter

Take - Adds an item into inventory

1. Key - Player picks up a key
2. Gunpowder - Player takes gunpowder found in the armory room

Go - Lets the player navigate through the game map

1. Hallway - The player moves down that hallway
2. Door - The player moves to the door in a room, labelled with corresponding letters if there are multiple in a room
3. Shed - Used to enter the shed
4. Table - Moves the player to the table in a room
5. Desk - Moves the player to the desk in the room
6. Chest - Moves the player to a chest inside the room
7. Tunnel - Enters the tunnel in Room 9

**Room List:**

**Room 1 - Storage Room**

The user awakens inside a dark room

Go -

1. Desk - Contains the R1 Key
2. Chest - Contains nothing useful, but can be opened with the R1 key anyway
3. Room 2 Door (Hallway-Guard) - Locked. Opened with R1 Key

Puzzle -

1. Go to desk
2. Look Desk
3. Take R1 key
4. Go to Door
5. Use R1 key on door
6. Go to Room 2 (Hallway)

**Room 2 - Hallway-Guard Room- Main Hall way**

Go -

1. Room 3 Door (Conveyer Belt Room) - Locked. Opened with R2 Key
2. Room 4 Door ( Textile Room) - locked, Opened with R3 Key
3. Room 5 Door (Battalion Armory) - Locked. Opened with R4 Key

Puzzle -

1. Use Rope - Gain R2 key
2. Go to Room 3 Door
3. Use R2 key
4. Go to Room 3 (Conveyer Belt Room)

Exceptions and Try/Catch Usage:

* Using any item that is not the rope will yield an error and will be caught in a Try/Catch statement.

**Room 3 - Conveyor Belt Room**

Look -

1. Conveyor Belt - Lets the user examine the conveyor belt and find a key

Puzzle -

1. Look - Conveyor Belt
2. Take R3 key
3. Return to Room 2 Hallway)
4. Use R3 Key on Room 4 Door (Textile Room)

**Room 4 - Textile Room**

Look -

1. Yarn - The player is haunted with memories of his best friend Pablo
2. R4 Key - This key is found in the ashes of the Yarn in the the Textile room

Puzzle -

1. Use Lighter
2. Look at key
3. Take key
4. Go to R5 Door
5. Use R4 key on R5 Door

**Room 5 -Battalion Armory**

Look -

1. Gunpowder - The player is given a hint of using the lighter with the gunpowder

Take -

1. Gunpowder - Adds gunpowder to the inventory

Puzzle -

1. Look at gunpowder
2. Take Gunpowder
3. Use lighter
4. Go to Room 6

**Room 6 - Prison Room**

Look -

1. Door - Blocked. Prison Guards are closing in on the player
2. Window - Possible escape route. Looking at this gives the player a hint

Puzzle -

1. Look at window
2. Use Rope - Escape to Room 7 (Courtyard)

**Room 7 - Courtyard**

Look -

1. Left - A description of a massive, hungry, growling pitbull causes the player to turn the other way. There is a door behind the
2. Right - There is a shed with a door facing the player on it

Puzzle -

1. Look Shed - The player looks at a description of the shed
2. Go Shed - The player enters the shed. No key is required. Enters Room 8 (Shed)

**Room 8 - Shed**

Go -

1. Table - The player finds a key on the table

Look -

1. Cobwebs - The player finds a door here where they can use the key they found on the table

Puzzle -

1. Go Table - The player finds the key
2. Take R9 Key
3. Look Cobwebs - Uncovers trap door
4. Use R9 key on Trap Door (Room 9 - Shed Basement)

**Room 9 - Shed Basement**

Look -

1. Right - Reveals a secret tunnel
2. Cage - Looking at the cage opens it because the player accidently hits a switch on the cage. The fleeing children give the player the hint of looking right to a secret tunnel

Go -

1. Tunnel - Goes to Room 10

Puzzle -

1. Look Cage - Children escape and drop the hint of looking right for the secret corridor
2. Look Right - Find secret door
3. Go tunnel

**Room 10 - Hidden Corridor Escape**

Look -

1. Door - The player can look at the door. Which will allow them to escape and end the game.

Exceptions -

1. Attempting to use any items will be caught by a try and catch statement and yield a - “cannot do that here” message