CS_162_400_S2019
Final Project
Susan Hibbert
ID# 933913975

<u>Final Project</u>

<u>Design Description</u>

Theme: I have based the theme on one of my favourite TV shows 'RuPaul's Drag Race'

Description: The following text will appear on screen to the user:

'You were in the middle of getting ready to hit the stage for the finale of RuPaul's Drag Race when suddenly the dressing room was hit by a tornado! All of your drag has been scattered all over the studio! And right before the big finale to crown America's next Drag Superstar! If you still want a chance to snatch the crown, you will need to collect all the items and get into drag before your feet give out from walking in heels!'

The player starts the game in the studio. They will have to collect a number of items before they are allowed to see RuPaul in his dressing room. Once they collect and use these items, RuPaul's dressing room will unlock and the player can go to visit him there. RuPaul will give the player an additional item and tells the player to check their appearance in the mirror. The player must use this additional item before checking their appearance in the mirror. Once they check their appearance in the mirror after using this item, a message will appear telling them that they are ready to hit the stage and the stage door will subsequently unlock. The player must make their way to the stage door to complete the game.

The player's health declines with each move of the game, due to the fact that walking in high heels hurts their feet. They can restore their health by collecting one of several Ointment objects, named 'RuPaul's Magic Foot Ointment', scattered around the studio.

A game map will be displayed at every move to allow the user to see the placement of the various items and locations of the different Space objects and allow them to track their movement across the game map.

Classes:

PARENT CLASS – SPACE (abstract class)		
Derived classes	*Description*	*Conditions for use*
Studio	The player starts in the studio where they collect the following items: Dress, Eyelashes, Ointment	None – the player can move freely between the Studio objects
Wall	Wall objects surround the edge of the studio	n/a – the player cannot move into Wall objects

Dressing Room	Where RuPaul can be found. RuPaul gives you a Wig item and tells you to go check yourself in the Mirror	They can only enter the dressing room once they have collected and used the Dress and Eyelashes objects
Mirror	There are 2 Mirror objects in the game. The player must go to the Mirror once they have obtained and used the Wig from RuPaul. A message will appear telling them they look ready and to head to the Stage Door to complete the game	The player will be told that they do not look ready if they interact with the Mirror before obtaining and using the Wig. Once they have obtained and used the Wig, they will be told they look ready and to go to the Stage Door. At this point the Door object is unlocked and the player can interact with it
Stage Door	Represents the door to the stage where the finale of RuPaul's Drag Race takes place. Once unlocked, when the player interacts with the Door object the game will end	The player will be told that they are not ready to go on stage if they interact with the Door object before viewing themselves in the Mirror following use of the Wig object.

Container to store items: The container to store items is represented as a Purse. I will create a Purse class which stores the various items collected in the game, implemented as an array of 4 pointers to Item objects. The limit will be 4 items

Items: I will create an Item abstract parent class, and each particular item is a derived class of the Item parent class. The derived classes are Dress, Eyelashes, Wig, and Ointment

Development Plan:

I initially plan to write the Space class and its derived classes, then write a function to build the game map with the different Space objects in particular locations on the map in order to create a visual representation of the TV studio. Then, I will write a function to print the game map on screen to verify the placement of the various Space objects. Once I am happy with the game map I will link all the different Space objects together via their Space pointers. Following this, I will implement the Player class and get the Player object to display on the game map, so it can be visually tracked. I will ensure that the Player can move and that their health declines with each movement and verify that they can interact with the various Space objects. Following this I will create the Item class and derived classes and place the various items around the studio. After I am able to do this, I will implement the Purse container, ensuring that when the user picks up items they go into the purse and are removed from the game map. After all this has been completed and I have implemented a menu function I plan to implement the test cases listed in the Test Cases table to detect any runtime issues with my code.

CS_162_400_S2019
Final Project
Susan Hibbert
ID# 933913975

<u>Test Cases</u>

Test Case	**Input Values**	**Driver Functions**	**Expected Outcomes**	**Observed Outcomes**
START OF GAME				
Player does not enter an integer (1 or 2) when prompted to play game or exit program	Input is a not an integer. May be a character, float, or a mix of characters and integers including spaces. Player may just have pressed the enter key	Int_input_val function of display_menu function of menu functions *Do-while loop prompts player to enter an integer*	Player is prompted to enter an integer until an integer is entered by the player	Player is prompted to re-enter an integer until integer is entered, as expected. Chars, floats and input containing spaces are rejected. If player presses enter, input is rejected
Player enters an integer other than 1 or 2 when asked whether they would like to play game or exit	Input is an integer other than 1 or 2	Display_menu function of menu functions *While loop prompts player to enter 1 or 2*	Loop back to the question prompting player to enter 1 or 2	Program loops back to the question prompting player to enter 1 or 2, as expected.
GAME PLAY				
Check game map displays correctly with correct placement of the different Space objects	The game map is set up correctly and the positions of the various Space objects remain constant throughout the running of the game	Initialize function of game class Display_map function of game class	The game map is set up correctly and the positions of the various Space objects remain constant throughout the running of the game	The game map is set up correctly and the positions of the various Space objects remain constant throughout the running of the game as expected
Player does not enter an integer when prompted to select a direction to move from the 4 integer options displayed on screen	Input is a not an integer. May be a character, float, or a mix of characters and integers including spaces. Player may just have pressed the enter key	Int_input_val function in direction function of menu functions *Do-while loop prompts player to enter an integer*	Player is prompted to enter an integer until an integer is entered by the player	Player is prompted to re-enter an integer until integer is entered, as expected. Chars, floats and input containing spaces are rejected. If player presses

CS_162_400_S2019
Final Project
Susan Hibbert
ID# 933913975

				enter, input is rejected
Player enters an integer other than 1 – 4 when prompted to select a direction to move from the 4 integer options displayed on screen	Input is an integer other than 1-4	Direction function of menu functions *While loop prompts player to enter an integer between 1 and 4*	Loop back to the question prompting player to enter a number between 1 and 4	Program loops back to the question prompting player to enter an integer between 1 and 4, as expected
Check player starts in correct position on game map and are able to track their movement across game map. Check health declines at each move	Player starts at row position 1 and column position 1 on game map. Player moves in direction specified by player and their health declines by 10 points at each move	Set_player function of Game class Move function of Game class Lose_health function of Player class Display_map function of Game class	Player starts at row position 1 and column position 1 on game map. Player moves in direction specified by player and their health declines by 10 points at each move as displayed after each move	Player starts at row position 1 and column position 1 on game map. Player moves in direction specified by player. Their health starts at 100 and declines by 10 points after each move as expected
Check that picking up or receiving an item adds the item to the Purse. Using the item causes the item in the Purse to change or be removed, depending on the item. If the item was picked up in the Studio, it disappears from game map	When the player picks up an item from the studio, it is added to their purse and no longer appears on the studio map. When they use an item in their Purse, the item's name changes (if the item used was a Dress, Eyelashes or Wig – this is to indicate that the player used the item) or it is removed from the purse (if the item was Ointment)	Get_char function of Studio class Item_action function of Item class Use_item function of purse class *If/else statements call on set_item_type function of Item class for Dress, Eyelash and Wig objects. They are not deleted from the purse, just have their string data member item_type changed. Ointment*	When player puts on Dress, a message appears on screen informing them that they are now wearing the Dress and the Old Sweater they were wearing is now in their purse. Old Sweater appears in purse. When player puts on False Eyelashes, a message appears on screen informing them that they are now wearing the False Eyelashes and the	When player puts on Dress, a message appears on screen informing them that they are now wearing the Dress and the Old Sweater they were wearing is now in their purse. Old Sweater appears in purse. When player puts on False Eyelashes, a message appears on screen informing them that they are now wearing the False Eyelashes and the Contact Lens they were wearing is now in their purse. Contact Lens

		objects are deleted after use	Contact Lens they were wearing is now in their purse. Contact Lens appears in purse. When player puts on Wig, a message appears on screen informing them that they are now wearing the Wig and the hat they were wearing is now in their purse. Hat appears in purse. When the player uses ointment, after restoring their health it is removed from the purse. If the item was picked up in the Studio, it is removed from the game map and cannot be picked up again.	appears in purse. When player puts on Wig, a message appears on screen informing them that they are now wearing the Wig and the hat they were wearing is now in their purse. Hat appears in purse. When the player uses ointment, after restoring their health it is removed from the purse. If the item was picked up in the Studio, it is removed from the game map, as expected
Player does not enter an integer (1 or 2) when asked if they would like to use any items in their Purse	Input is a not an integer. May be a character, float, or a mix of characters and integers including spaces. Player may just have pressed the enter key	Int_input_val function of display_menu function of menu functions *Do-while loop prompts player to enter an integer*	Player is prompted to enter an integer until an integer is entered by the Player	Player is prompted to re-enter an integer until integer is entered, as expected. Chars, floats and input containing spaces are rejected. If Player presses enter, input is rejected
Player enters an integer other than 1 or 2 when asked if they would like	Input is an integer other than 1 or 2	Display_menu function of menu functions	Loop back to the question prompting player to enter 1 or 2	Program loops back to the question prompting player to enter 1 or 2 as expected

to use any items in their Purse		*While loop prompts player to enter 1 or 2*		
Player does not enter an integer when asked which item from the Purse they would like to use	Input is a not an integer. May be a character, float, or a mix of characters and integers including spaces. Player may just have pressed the enter key	Int_input_val function in display_menu function of menu functions *Do-while loop prompts player to enter an integer*	Player is prompted to enter an integer until an integer is entered by the player	Player is prompted to re-enter an integer until integer is entered, as expected. Chars, floats and input containing spaces are rejected. If player presses enter, input is rejected
Player enters an integer other than 1 – 4 when asked which item from the Purse they would like to use	Input is an integer other than 1-4	Display_menu function of menu functions *While loop prompts player to enter an integer between 1 and 4*	Loop back to the question prompting player to enter a number between 1 and 4	Program loops back to the question prompting player to enter an integer between 1 and 4, as expected
Ensure player cannot use empty slots in purse. Ensure player cannot use Old Sweater, Contact Lens or Hat in purse, which are put into the purse after player puts on Dress, Eyelashes and Wig respectively	Player will not be able to 'use' any empty slots in the purse. Player will also not be able to use Old Sweater, Contact Lens or Hat removed from player during the course of the game	Use_item function of purse class *If/else statements informs player they cannot use Old Sweater, Contact Lens or Hat if they selected to use one of those items* Display_menu function of menu functions *While loop prompts user to select another item if they selected an empty slot*	Purse is not displayed to the player if it contains no items. If player selects to use an empty item slot they are asked to pick another item. If the player selects to use Old Sweater, Contact Lens or Hat they will be told they cannot use that item and not be re-prompted	Purse is not displayed to the player when the player has not picked up any items. When player selects to use an empty item slot they are asked to pick another item. If the player selects to use Old Sweater, Contact Lens or Hat they are told they cannot use that item and are not re-prompted, as expected

		While loop prevents purse being displayed when the purse contains no items		
Ensure that using RuPaul's Magic Foot Ointment restores health back to 100	When player picks up an Ointment object from the game map, it goes into their purse. When they use the ointment, their health is reset back to 100	Item_action function of Ointment class *Calls upon set_health function of Player class*	When the player uses the ointment, their health is reset back to 100	When the player uses the ointment, their health is reset back to 100, regardless of their current health points, as expected
Ensure that when the purse is full the player cannot add any additional items to the purse	If purse is full no more items can be added to the purse and player will be prompted to use items in the purse to make room	Add_item function of purse class *If/else statements. When a fourth item is added to the purse, the bool full is set to true*	When the purse contains 4 items and the player picks up another item it will not be added to the purse and the player will be prompted to use current items in the purse to make room	When the purse contains 4 items and the player picks up another item it is not added to the purse and the player is prompted to use items in the purse to make room as expected
Check player cannot move into a space occupied by a Wall object . If player moves into a Wall they are prompted to pick another direction	When player hits a Wall in the game map, they are asked to pick another direction to move	Is_wall function of Space class Move function of Game class Direction function of menu functions	When player moves into a wall, a message will display on screen telling them that they hit a wall and they are prompted to pick another direction to move until they select a direction that does not cause them to hit a wall again	When player moves into a wall, a message is displayed on screen telling them that they hit a wall and they are prompted to pick another direction to move until they select a direction that does not cause them to hit a wall again, as expected
Player cannot access the Dressing Room to see RuPaul until found and put on Dress	When player tries to access Dressing Room object, representing RuPaul's Dressing	Is_ready function of DressingRoom class	When player tries to access Dressing Room object before they have collected and put	When player tries to access Dressing Room object before they have collected and put on both the

and Eyelashes objects	Room, before they have found and put on the Dress and Eyelashes objects they will be informed that RuPaul is not ready to see them yet. When the player has found and put on the Dress and Eyelashes objects and then visit the Dressing Room, dialogue with RuPaul appears on screen and they receive the Wig item	Move function of Game class	on the Dress and Eyelashes objects they will be informed that RuPaul is not ready to see them yet. If they have found and put on the Dress and Eyelashes objects, they will be able to access the Dressing Room object. They are not able to re-enter the Dressing Room once they receive the wig from RuPaul	Dress and Eyelashes objects they are informed that RuPaul is not ready to see them yet. When they have found and put on the Dress and Eyelashes objects and then visit the Dressing Room, dialogue with RuPaul appears on screen and they receive the Wig item. Player is not able to re-enter the Dressing Room once they receive the wig from RuPaul, as expected
Check that Mirror only tells player they look ready and unlocks the Stage Door after they have received and put on the Wig object from RuPaul	When the player tries to access a Mirror object before they have received and put on the Wig they will be informed that they do not look ready yet. When the player has received and put on the Wig object (and completed all preceding steps), when they visit the Mirror it will tell them that they look amazing and to go to the Stage Door, which will subsequently be unlocked	Is_ready function of Mirror class Move function of Game class	When the player tries to access a Mirror object before they have received and put on the Wig they will be informed that they do not look ready yet. When the player has received and put on the Wig object (and completed all preceding steps), when they visit the Mirror it will tell them that they look amazing and to go to the Stage Door, which will subsequently be unlocked	When the player visits the Mirror object before finding and putting on any items, a message appears on screen telling them that they are not ready. When the player visits the Mirror object after finding and putting on the Dress and Eyelashes objects a message appears on screen telling them that they are not ready and asks them to choose another direction to move. When the player visits the Mirror object after receiving the Wig

CS_162_400_S2019
Final Project
Susan Hibbert
ID# 933913975

				object a message appears on screen telling them that they are not ready and asks them to choose another direction to move. After the player puts on the Wig object, only then does a message appear telling the player that they look ready and to head to the stage, as expected
Check that player cannot access Stage Door and complete the game unless a number of conditions have been met	When player tries to access Door object, representing the Stage Door, before they have been to the Mirror object wearing the Wig they will be informed that they are not ready to go on stage. If player has been to the Mirror object wearing the Wig (and completed all preceding steps), the Door object will unlock, and player can access Door object and complete game	Is_ready function of Door class Move function of Game class Game_over function of Game class *Displays end of game dialogue. Bool game_over set to true and while loop in display_menu function stops*	When player tries to access Door object before completing any of the preceding steps a message is displayed on screen informing them that they are not ready to go on stage and asks them to choose another direction to move. If player has been to the Mirror object wearing the Wig (and completed all preceding steps), the Door object will unlock, and player can access Door object and complete game	When player tries to access Door object before completing any of the preceding steps a message is displayed on screen informing them that they are not ready to go on stage and asks them to choose another direction to move. When player has been to the Mirror object wearing the Wig (and completed all preceding steps), the Door object unlocks, the end of game dialogue appears and game ends, as expected
Game continues unless player dies or they complete game	The player will continue playing the game and selecting directions	Display_menu function of menu functions	Player continues to play the game while their health is above zero and	Player continues to play the game while their health is above zero and they have

	to move while they still have health points and they have not yet unlocked the stage door. Player cannot have negative health points. When they have 0 health points the game ends	*While loop continues while player's health is greater than zero and game_over bool is false* Lose_health function of player class *If/else statements set health to 0 if player has negative health points*	they have not yet unlocked and interacted with the Door object, representing the stage door. Their health does not drop below 0 and when they have 0 health points the game ends	not yet unlocked and interacted with the Door object representing the stage door. When their health reaches 0 the game ends as expected

CS_162_400_S2019
Final Project
Susan Hibbert
ID# 933913975

Reflection

There were some important details I missed from my original design description, such as defining the roles of each different class destructor and how to access the different types of Space objects once a set of conditions had been met. I think I greatly underestimated the complexity of my design and as such ran into a number of issues. Had I spent more time on the planning stage I maybe would have realized this and simplified some aspects of my final project.

The largest issue I had was with memory leaks. However, one unexpected benefit from this was that it improved my understanding and appreciation of valgrind and gdb which helped me get to the root of the problem. The issue was caused by having multiple class destructors all trying to delete the same pieces of memory. In my initial design I did not consider this issue as I was planning to have each class clean up its own memory with its own destructor. After running valgrind on my program, I could see that I was freeing more memory than I was initially allocating. After a lot of debugging with no resolution, I decided to rewrite all my destructors and have only the Game class destructor responsible for freeing all the dynamically allocated memory in the program and this resolved the problem. This issue had made me appreciate the importance of defining the role of the destructor in the initial program design, thinking about how and what it should delete and how its actions can have a domino effect on other class destructors which is something that never occurred to me before.

Another issue I encountered was regarding how to access certain Space objects once a set of conditions were met. For instance, in my initial design I did not consider how to actually check if the Dressing Room object was accessible to the player, just that the player could interact with it once they had picked up and used the Dress and Eyelashes objects. This was the same for the Mirror and Door objects. In hindsight, I could have used a simple bool variable set to true once they had picked up and used both objects, but instead I implemented a more complex method involving a number of if/else statements and bool variables. Again, more thorough planning at the initial design stage could have saved me some time and simplified my code.