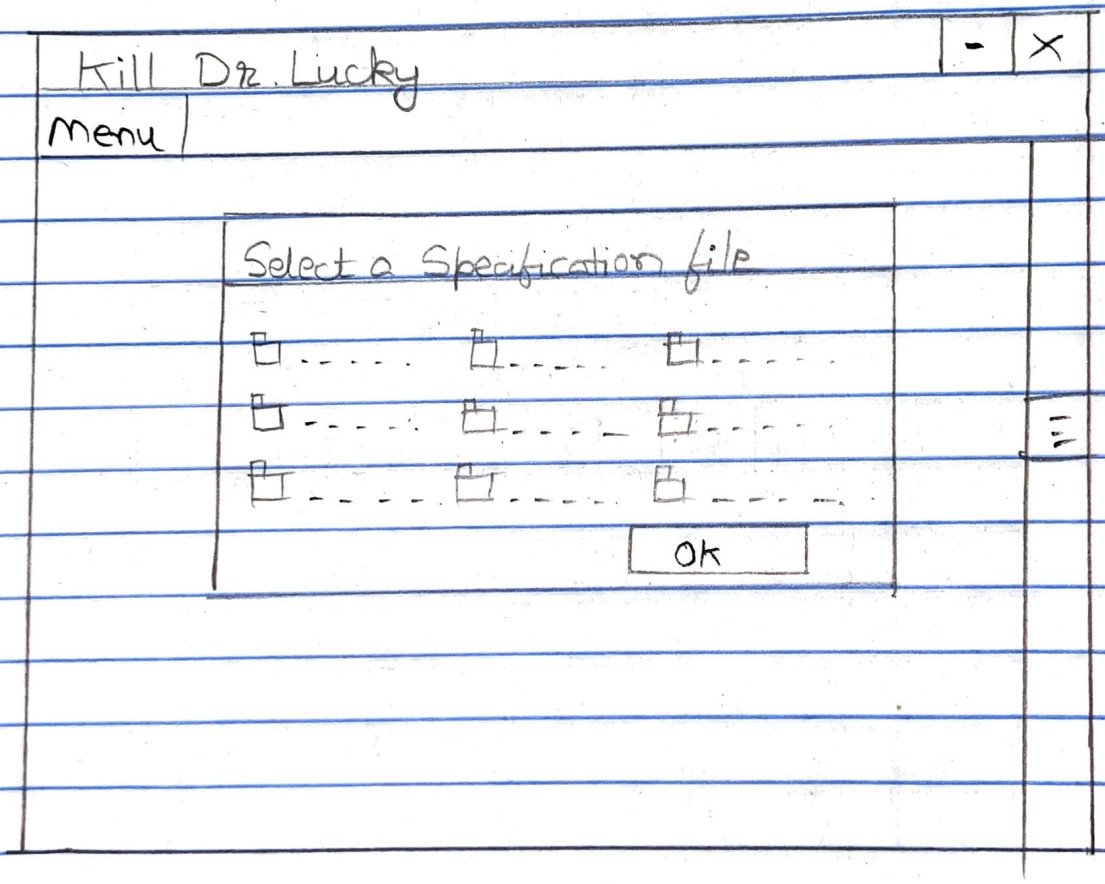
# Welcome Screen

Welcome
To
Kill Dr. Lucky

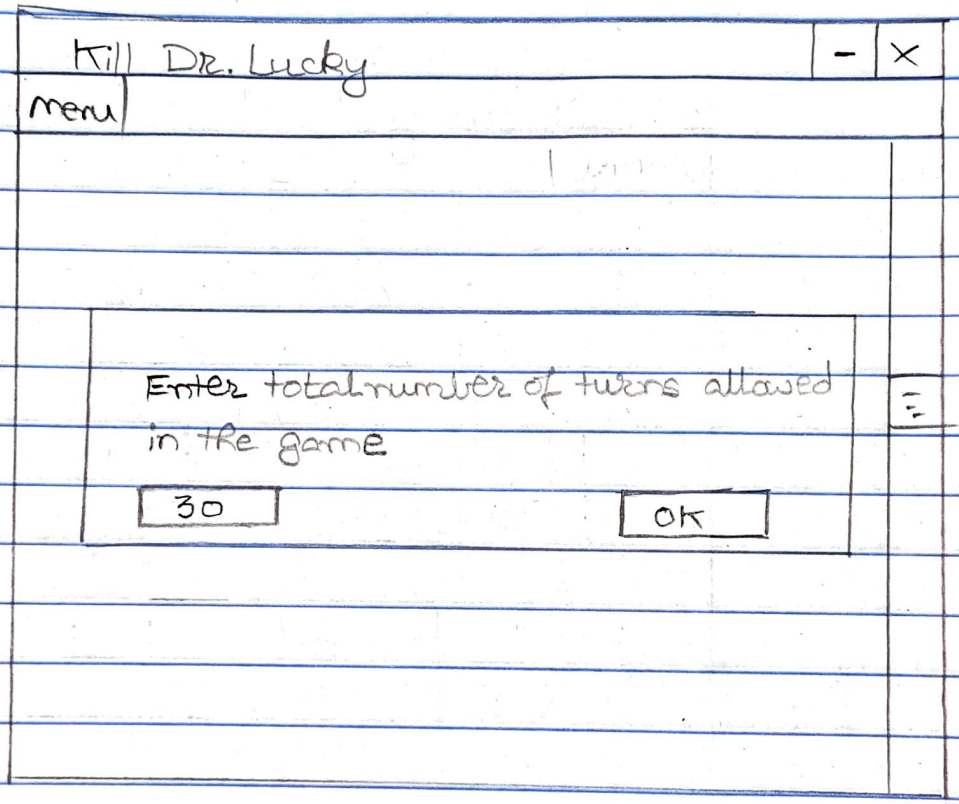
Credits:

Meet Vora  Prinjal Dave

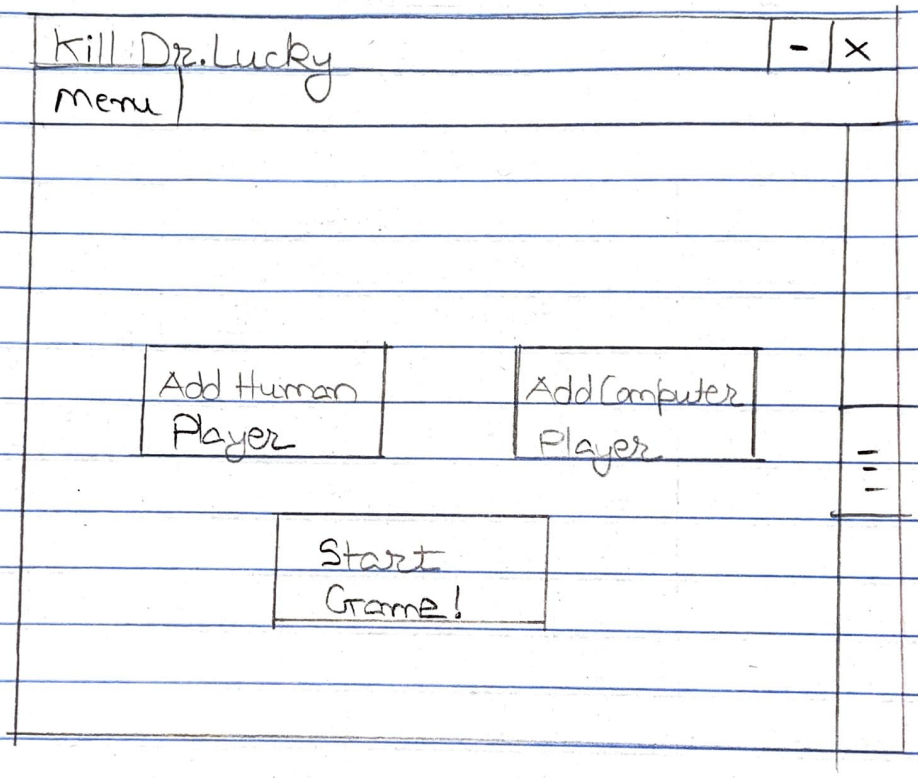
# Pick Specification File



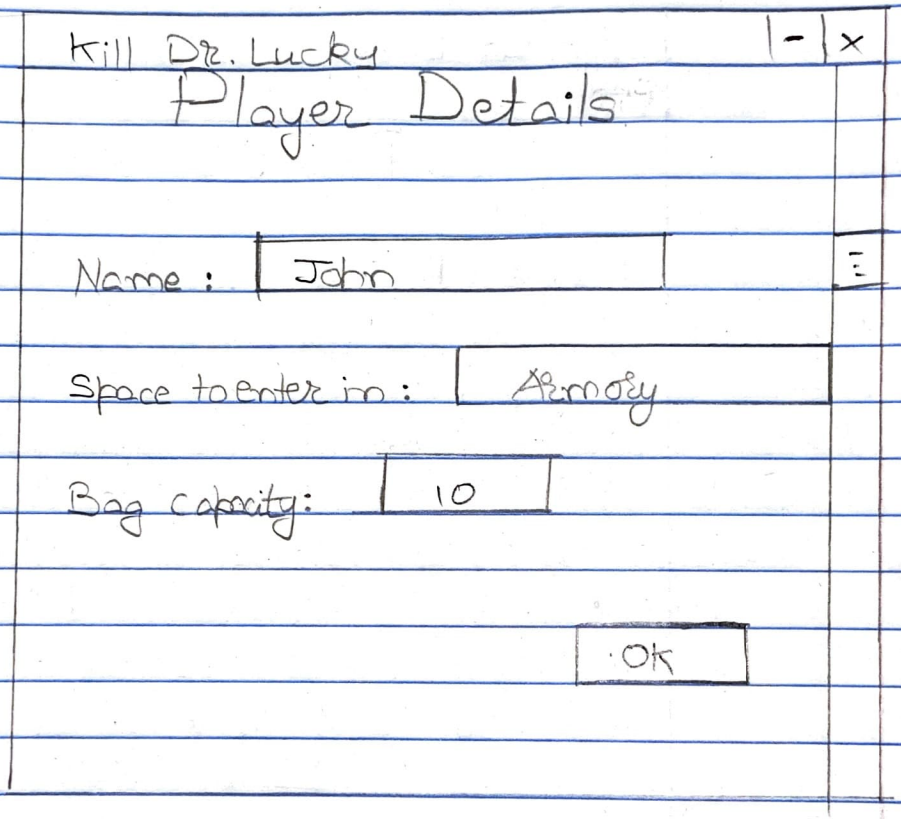
# Number Of Turns

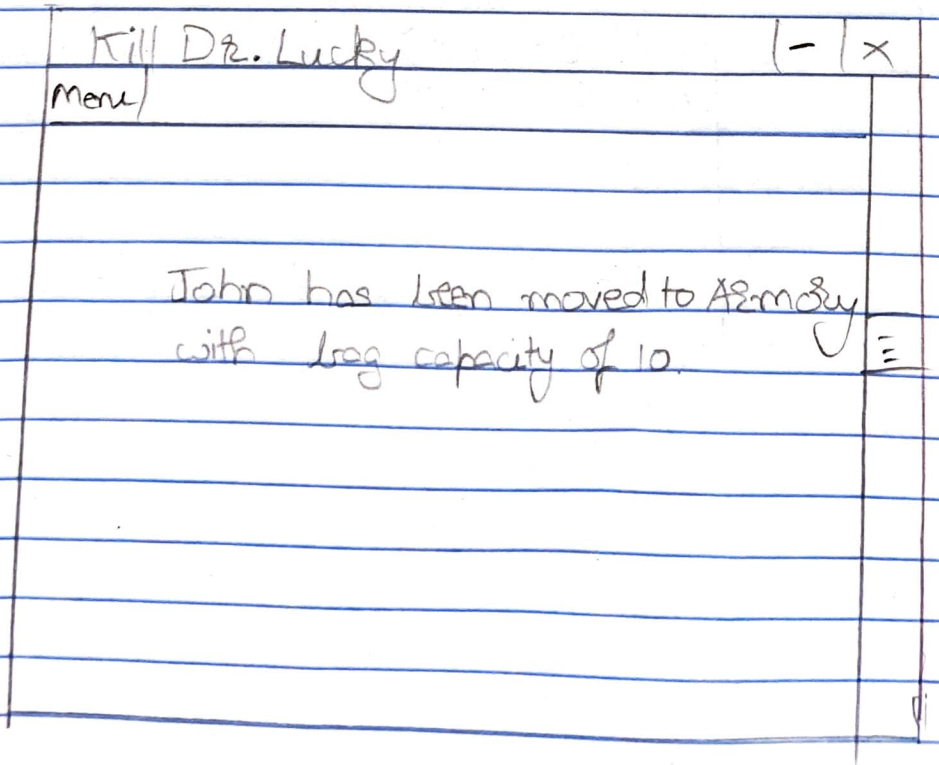


# Add Player Screen

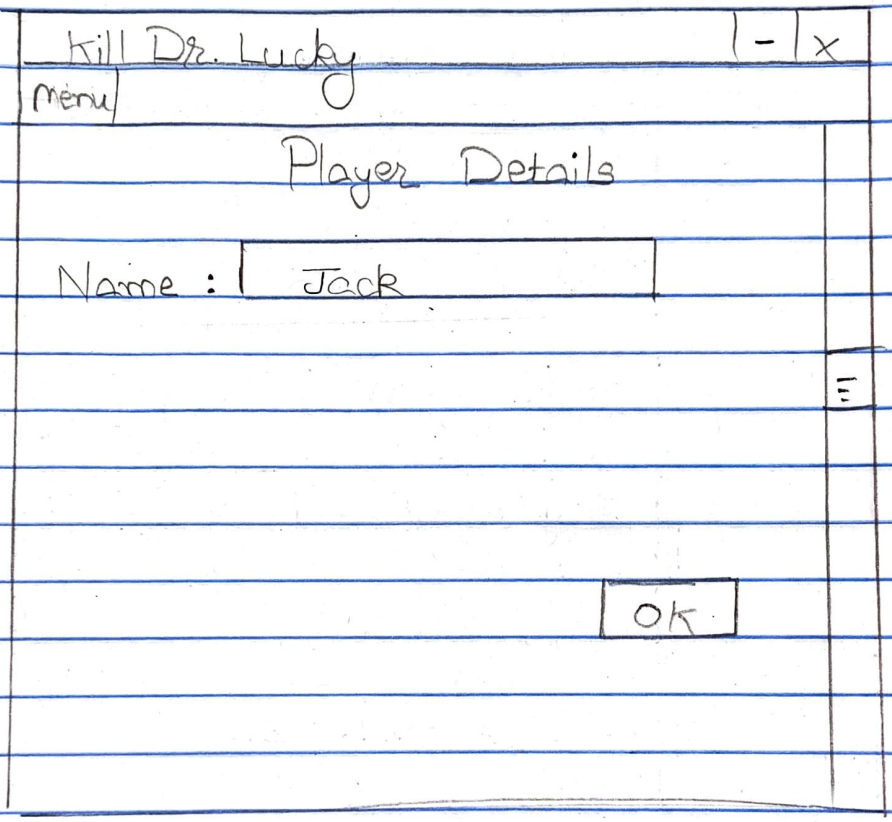


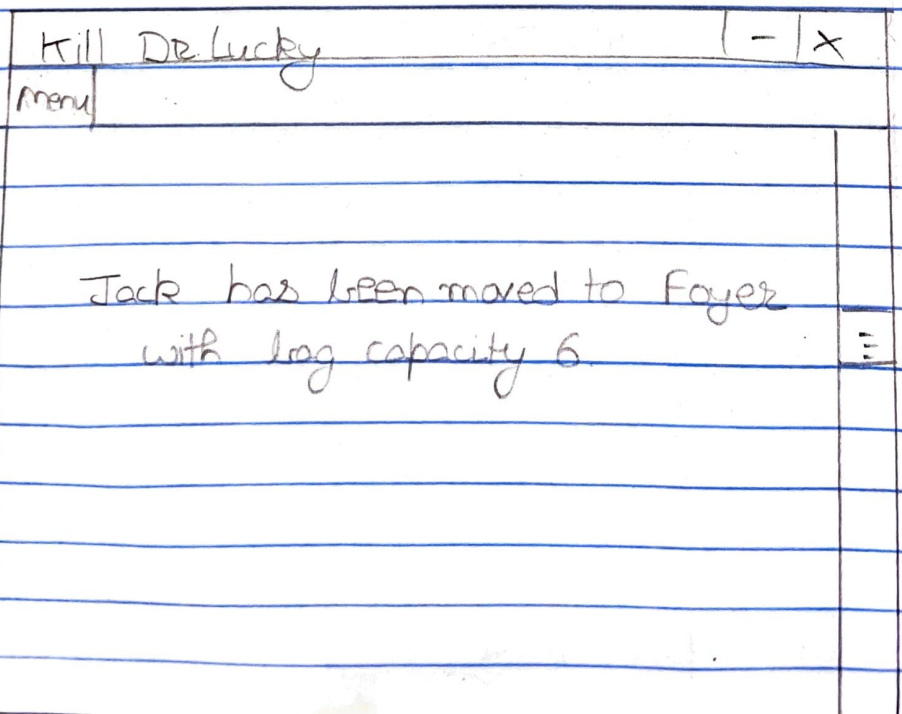
# Add Human Player



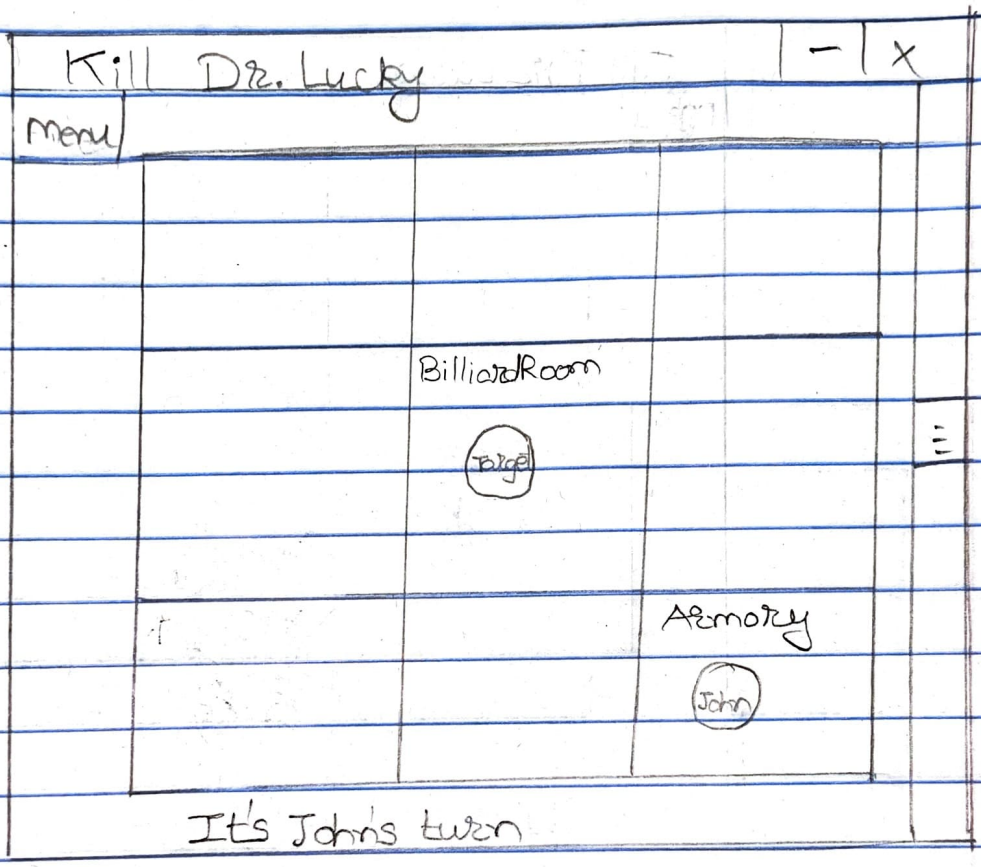



# Add Computer Player

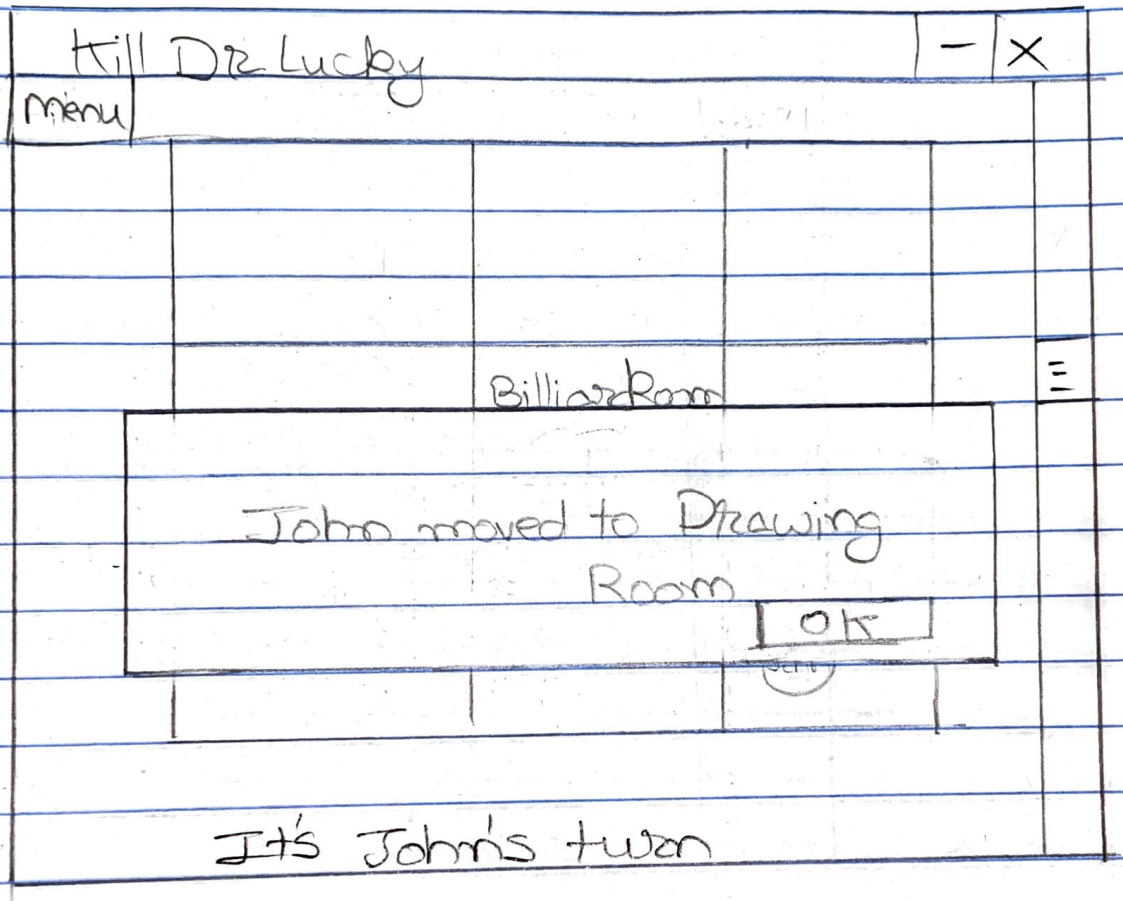


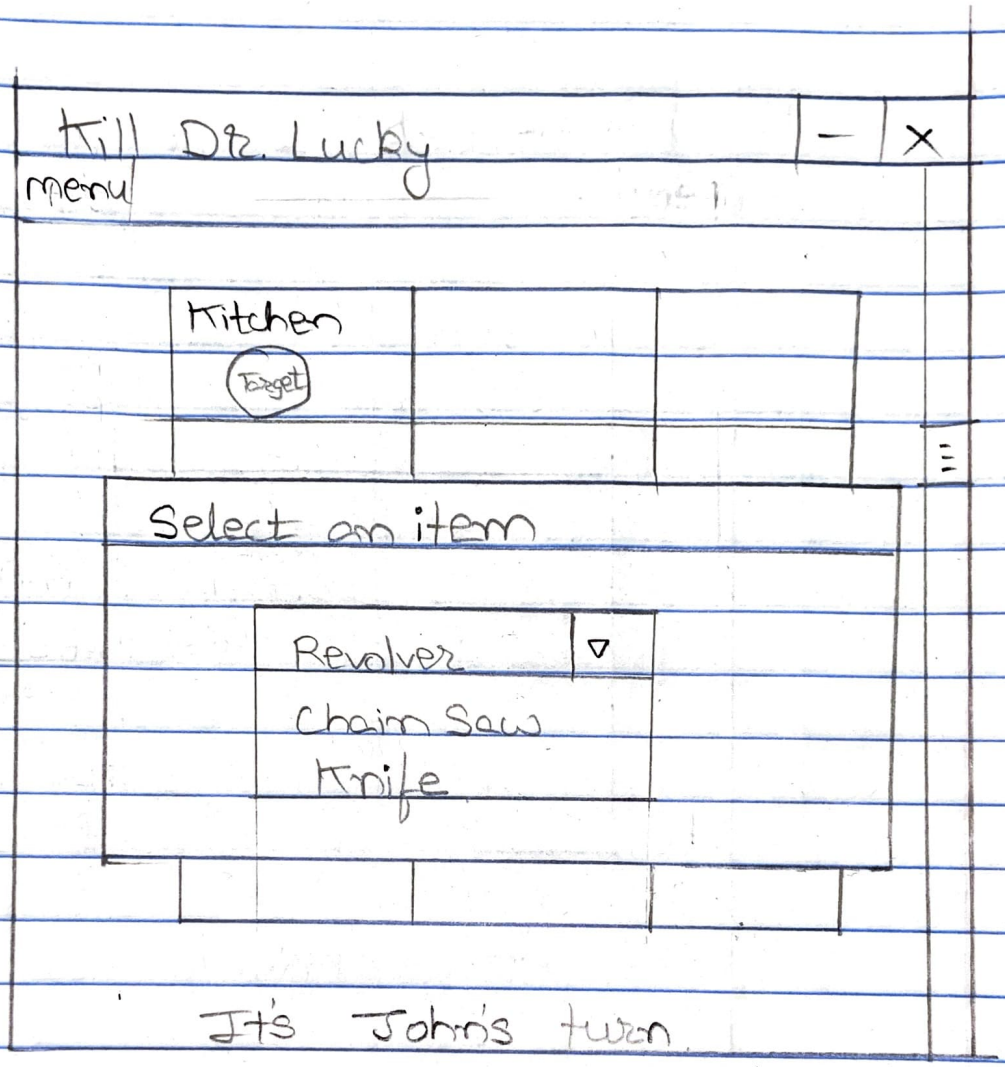
# Game Main Screen

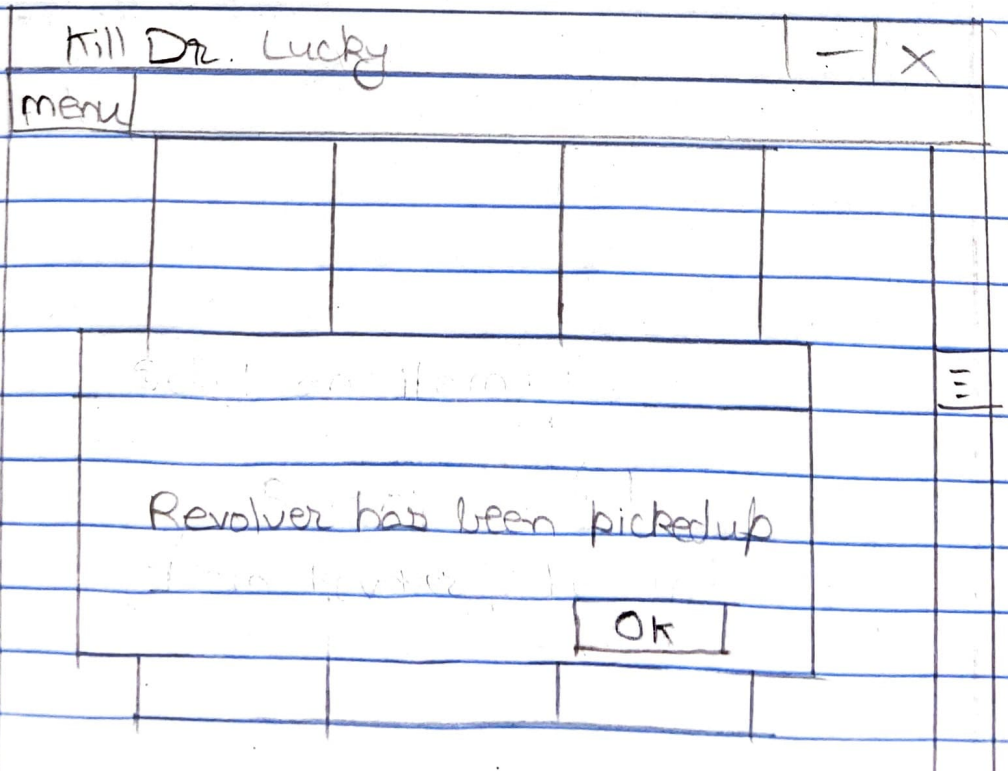


# Move Player

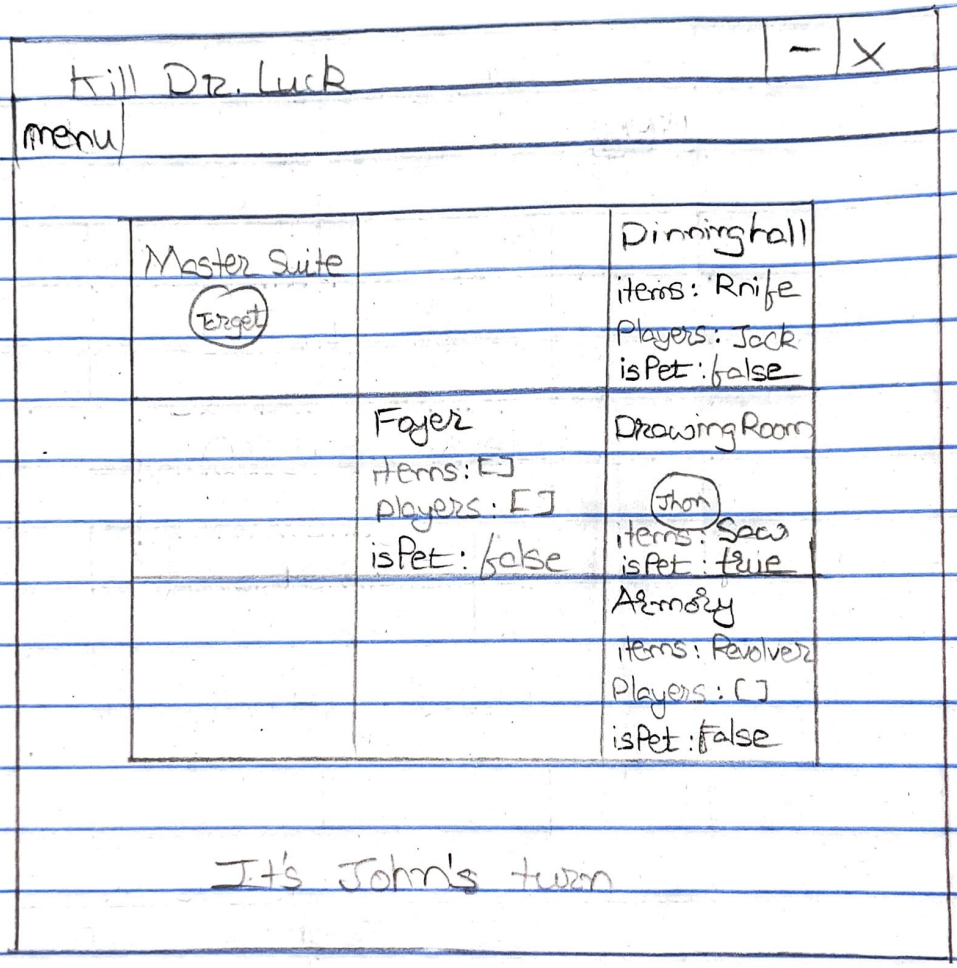

Kill Dr Lucky
Menu
Billiard Room
John moved to Drawing Room
OK
It's John's turn

# Pick Item

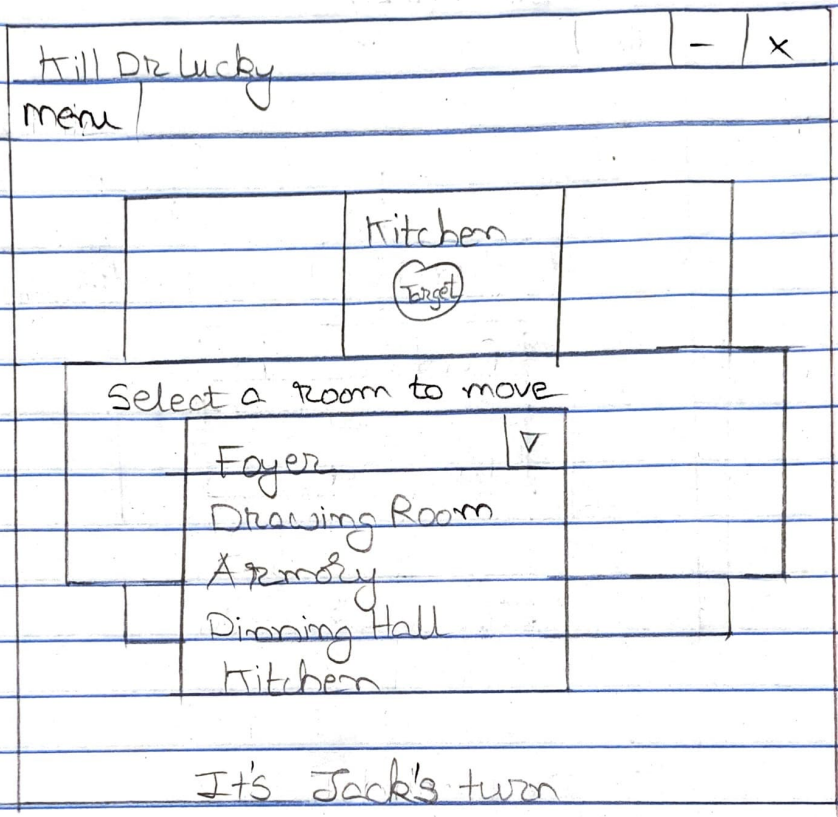

Kill Dr. Lucky
menu
Kitchen
Exget
Select an item
Revolver
Chain Saw
Knife
It's John's turn

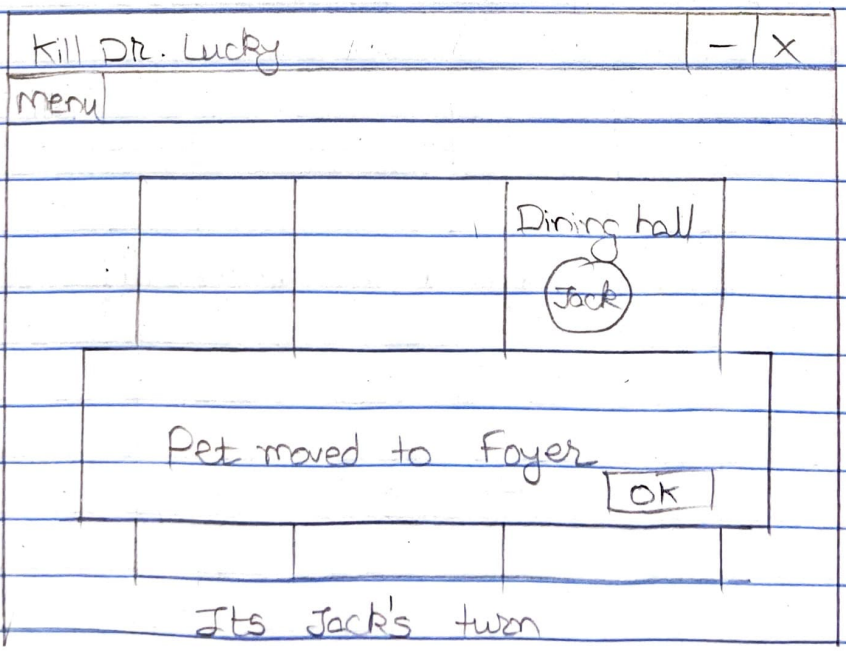

Kill Dr. Lucky
menu
Revolver has been picked up
Ok

# Look Around

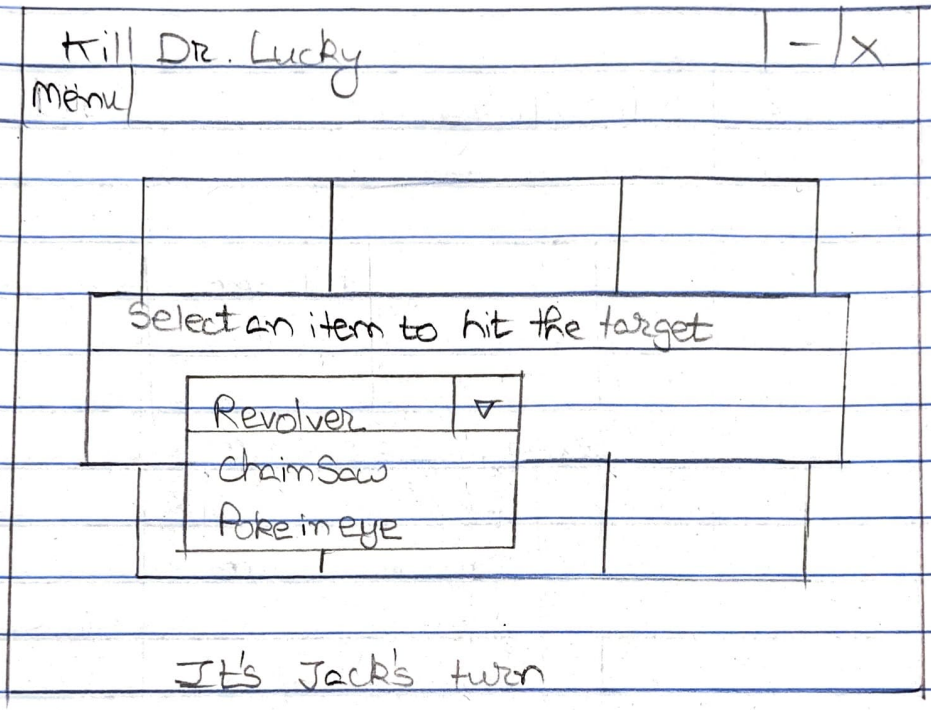

Kill Dr. Luck
menu
Master Suite
Target
Foyer
items: []
players: []
isPet: false
Dinningtall
items: Rnife
Players: Jack
isPet: false
Drawing Room
John
items: Saw
isPet: true
Armory
items: Revolver
Players: []
isPet: false
It's John's turn

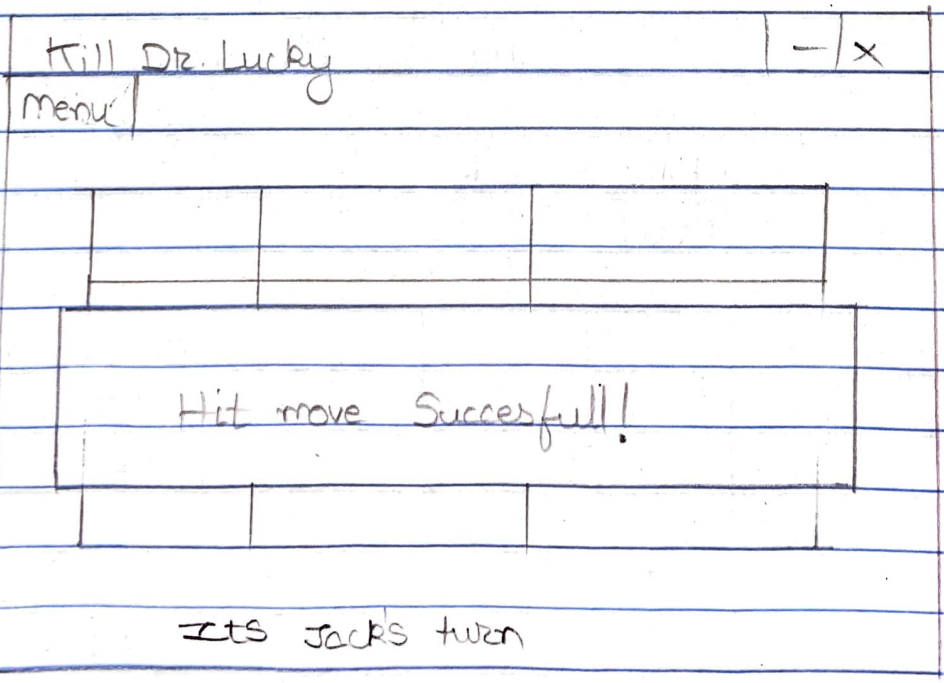
# Move Pet


Kill Dr Lucky
menu
Kitchen
Target
Select a room to move
Foyer
Drawing Room
Armory
Dinning Hall
Kitchen
It's Jack's turn

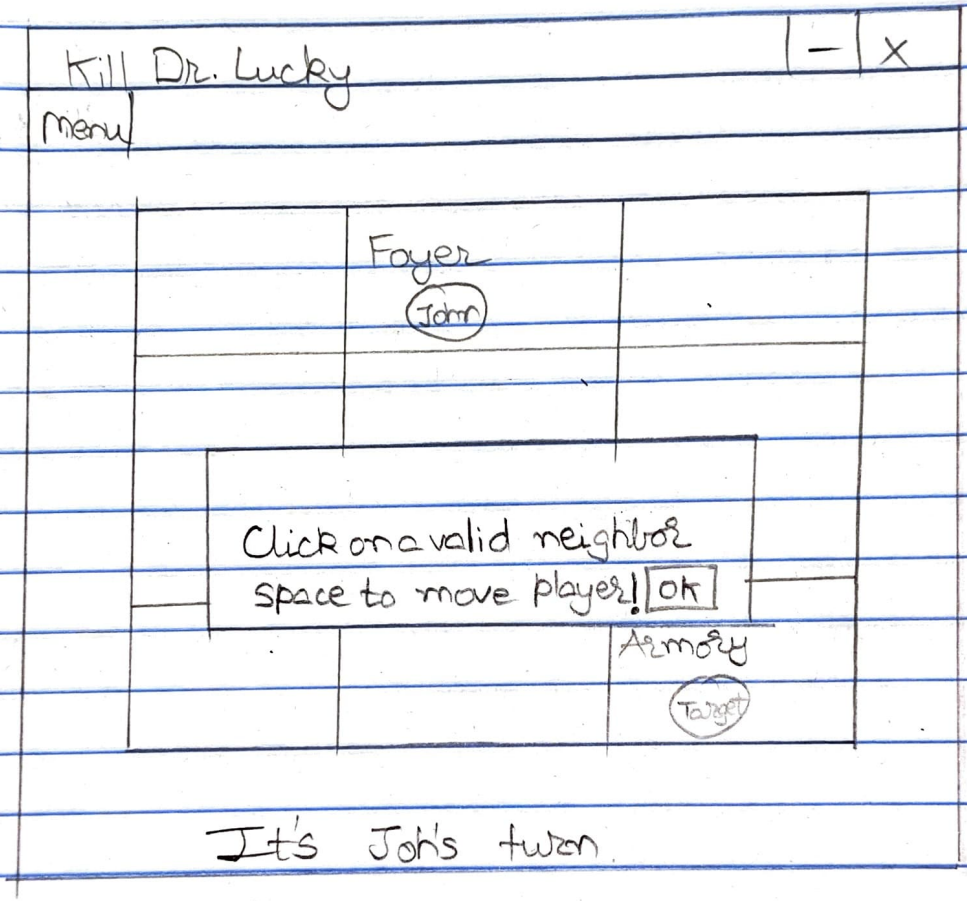

Kill Dr. Lucky
menu
Dining hall
Jack
Pet moved to Foyer
OK
Its Jack's turn

# Hit target

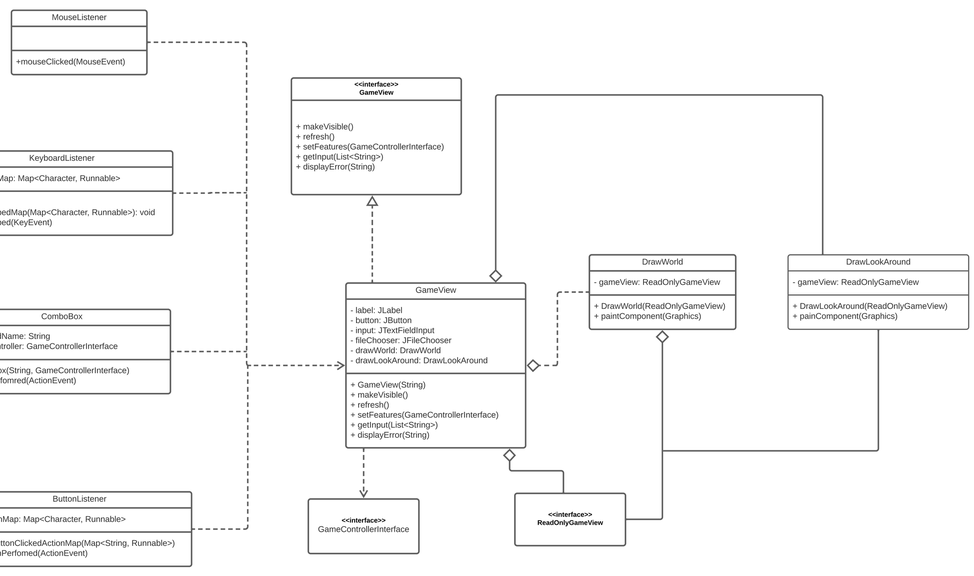

Kill Dr. Lucky
Menu
Select an item to hit the target
Revolver
Chainsaw
Poke in eye
It's Jack's turn


Kill Dr. Lucky
Menu
Hit move Succesfull!
Its Jacks turn

# Error Screens.



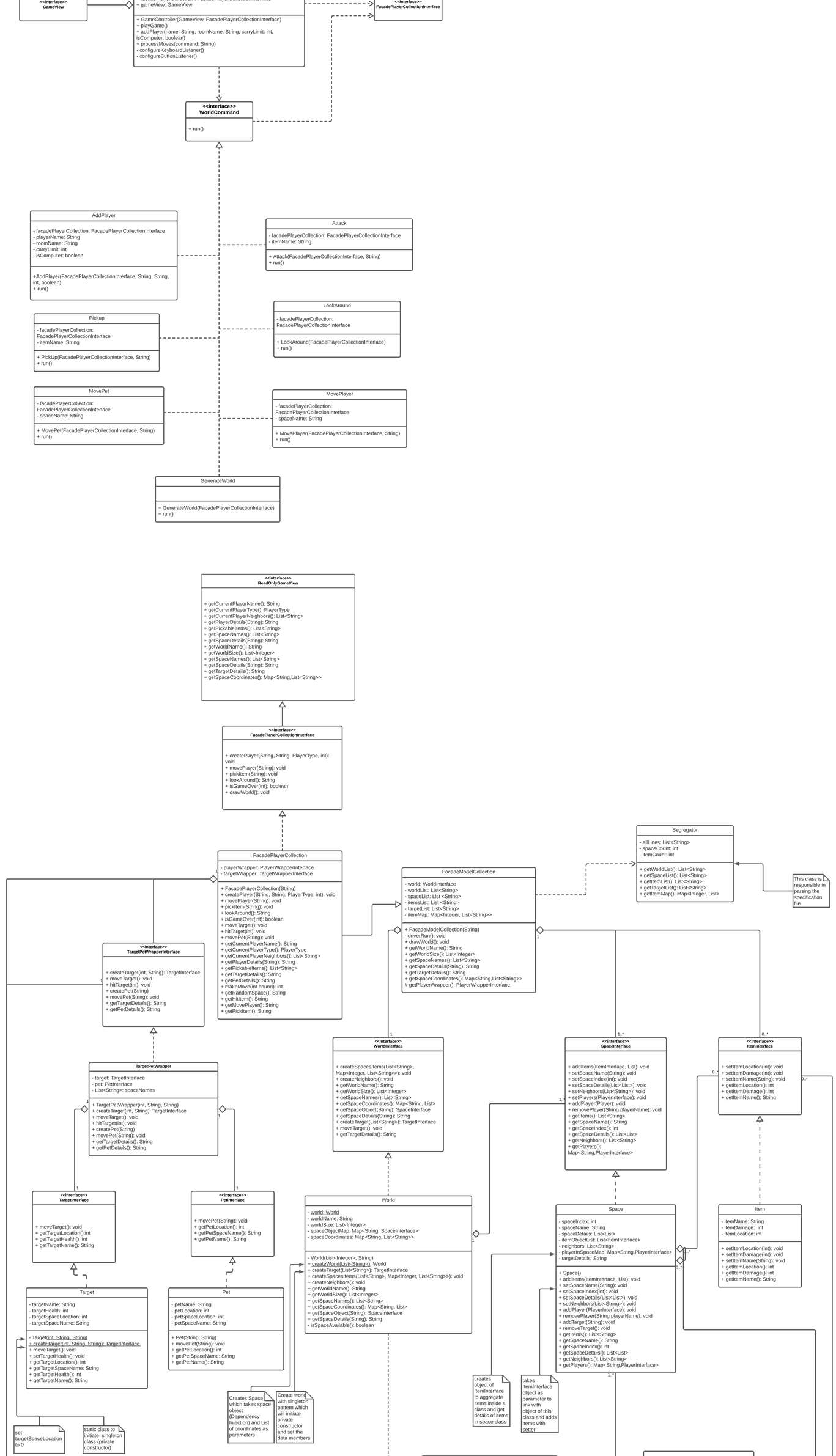
# View




# Controller




# Model



# Test Cases

# Milestone - 4 Testing Plan

## GameController using MockGameFacade and MockGameView model

| Test Description | Input | Output |
|---|---|---|
| Test Valid Constructor | GameController(GameView, FacadePlayerCollectionInterface) | Returns nothing, just creates the object. |
| Test Invalid Constructor | GameController(null, null) | Throw IAE |
| Test playGame() | playGame() | Makes the board visible. |
| Test addPlayers() | addPlayer(playerName, roomName, carryLimit, isComputer) | Adds the player in the game using the MockGameFacade model. |
| Test processCommand() for lookAround command. | processCommand("lookAround") | Sets the lookAround view on keyPressed, and unsets the view after keyReleased |
| Test processCommand() for movePet command without input. | processCommand("MovePet") | Controller asks for more input from the view, throws exception. |
| Test processCommand() for movePet command with input. | processCommand("MovePet", "Armory") | Moves the pet at the specified location and refresh the view |
| Test processCommand() for movePet command with invalid input. | processCommand("MovePet", "InvalidName") | Throws the exception and calls the displayError method from view(Mock view). |
| Test processCommand() for attack command without input. | processCommand("Attack") | Controller asks for more input from the view, throws exception. |
| Test processCommand() for attack command with input. | processCommand("Attack", "Pan") | Attacks the target with the specified item and refresh the view |
| Test processCommand() for attack command with invalid input. | processCommand("Attack", "InvalidName") | Throws the exception and calls the displayError method from view(Mock view). |
| Test processCommand() for move command without input. | processCommand("Move") | Controller asks for more input from the view, throws exception. |
| Test processCommand() for move command with input. | processCommand("Move", "Armory") | Moves the player to the specified location and refresh the DrawWorld view |
| Test processCommand() for move command with invalid input. | processCommand("Move", "InvalidName") | Throws the exception and calls the displayError method from view(Mock view). |
| Test processCommand() for pickup command without input. | processCommand("Pickup") | Controller asks for more input from the view, throws exception. |
| Test processCommand() for pickup command with input. | processCommand("Pickup", "Knife") | Picks' up the specified item and refresh the DrawWorld view. |

# Milestone 3

## Model Tests

| **TargetPetWrapper** | **Input** | **Output/Testing Methodology** |
|---|---|---|
| Create Invalid TargetPetWrapper (Invalid Target Health) | TargetPetWrapperInterface targetPetWrapper=new TargetPetWrapper(-10,"Dr Lucky","Petname") | Throws IllegalArgumentException |
| Create Invalid TargetPetWrapper (Invalid Target/Pet name) | TargetPetWrapperInterface targetPetWrapper=new TargetPetWrapper(10,null,null) | Throws IllegalArgumentException |
| Create Valid TargetPetWrapper | TargetPetWrapperInterface targetPetWrapper=new TargetPetWrapper(10,"Dr Lucky","Petname") | void<br>Method to verify:<br>Test target and pet details with getTargetDetails() and getPetDetails() |
| Move target | move() | void<br>Method to verify:<br>getTargetDetails() before and after the method call |
| Hit target invalid | hitTarget(-10) | Throw IllegalArgumentException |
| Hit Target | hitTarget(10) | void<br>Method to verify:<br>getTargetDetails() before and after the method call |
| Hit Target check Item is deleted from player | hitTarget(10) | Check item with getPlayerDetails() method before and after the method call |
| Move pet invalid | movePet("Invalid Space name")<br>movePet(null) | Throws IllegalArgumentException |
| Get target details | getTargetDetails() | Verify the ouput with expected string |
| Get pet details | getPetDetails() | Verify the ouput with expected string |
| Move Pet DFS | movePet("Armory") | Verify with getSpaceDetails() method before and after the method call |

| | | |
|---|---|---|
| Look around check player visibility | movePet(“Billiards Room”)<br>lookAround() | Verify that players in billiards room are not visible from any other room |
| Look around check item visibility | movePet(“Billiards Room”)<br>lookAround() | Verify that items in billiards room are not visible from any other room |
| Computer controlled player always hits target with most item damage | hitTarget(4) | void<br>Method to verify: getTargetDetails() before and after the method call |
| Stop attack if seen by another player | move()<br>hitTarget(4) | void<br>Method to verify: getTargetDetails() before and after the method call. Target’s health should not be decreased. |
| Kill Target | hitTarget(5) | Check isGameOver() should return true |

## Controller Tests

| Controller Methods | Input | Output |
|---|---|---|
| Invalid movePet command | movePet(null) | Throw IllegalArgumentException |
| Verify movePet command with mock class | movePet(“Foyer”) | Display affirmation in mock class and assert the same<br><br>“Pet moved to Foyer” |
| Invalid hitTarget command with mock class | hitTarget(null) | Throw new IllegalArgumentException |
| Verify hitTarget command with mock class | hitTarget(5) | Display affirmation in mock class and assert the same<br><br>“Target hit succesfull” |
| Check user input | Move player | Display affirmation in mock class and assert the same |

# Milestone 2

## Model Tests

| **PlayerWrapper** | **Input** | **Output/Testing Methodology** |
|---|---|---|
| Create Player Wrapper | world.createPlayerWrapper() | void<br>Method to verify:<br>Create world object and verify spaceList with PlayerWrapper spaceList |
| Create Player | createPlayer(“Jack”,”Armory”,”Human”) | void<br>Method to verify:<br>“Player Name: Jack, Space Name: Armory, Player Type: Human” |
| Create Player Invalid Space | createPlayer(“Jack”,”InvalidSpace”,”Human”) | Throws IllegalArgumentException |
| Create Player Invalid Name | createPlayer(20,”Armory”,”Human”) | Throws IllegalArgumentException |
| Change Turn | getTurn() | void<br>Method to verify:<br>Check current player before and after executing getTurn and validate whether the player got changed or not. |
| Move Player | movePlayer(“Billiard Room”) | void<br>Method to verify:<br>Check current player space and validate it with Billard Room. |
| Move Player Invalid Space | movePlayer(“InvalidSpaceName”) | Throws IllegalArgumentException |
| Pick Item | pickItem(“Pan”) | void<br>Method to verify:<br>Check current player item and validate it with Pan |

| | | |
|---|---|---|
| Pick Item<br>Invalid Item | pickItem(“Invalid item”) | Throws<br>IllegalArgumentException |
| See player | seePlayer(“Jack”) | “Player Name: Jack,<br>Space: Armory,<br>Neighbors: Billards Room” |
| See invalid player | seePlayer(“InvalidPlayerName”) | Throws<br>IllegalArgumentException |
| Get current player | getCurrentPlayer() | “Jack” |
| Get total turns | getTotalTurns() | 24 |
| Check target location after a move | movePlayer(“SpaceName”) | void<br>Method to verify:<br>Check target location before and after executing the movePlayer method. It should be incremented by 1. |

| **Player** | **Input** | **Output/Methodology** |
|---|---|---|
| Set Player Space name | setPlayerSpaceName(“Armory”) | void<br>verify the getPlayerSpaceName method it should return “Armory” |
| Set invalid Player Space name | setPlayerSpaceName(“InvalidSpaceName”) | Throws<br>IllegalArgumentException |

## Controller Tests

| Controller Methods | Input | Output |
|---|---|---|
| Check Move Receiver parameters | playeGame()<br>“Create Jack Armory move Billiards Room” | “Billiards Room” |
| Check Create Player Receiver parameters | playGame()<br>“Create Jack Armory” | “Jack Armory” |
| Check Display Space Receiver parameters | playGame()<br>“Display Armory” | “Armory” |
| Check Look Around Receiver Parameters | playGame()<br>“Look Jack” | “Jack” |
| Check Pick Item Receiver Parameters | playGame()<br>“Pick Pan” | “Pan” |

# Milestone1

Helper Class: WorldTest() calls World.createWorld()

| **World Creation** | **Input** | **Output** |
|---|---|---|
| World Valid | WorldTest([36,30,"Doctor Lucky's Mansion"]) | "Size=(36,30), Name= Doctor Lucky's Mansion" |
| World name Valid | getWorldName() | "Doctor Lucky's Mansion" |
| World size Valid | getWorldSize() | "36,30" |
| Space location Valid | getSpaceLocation() | "[Armory,Billiard Room, Carriage Hous…..]" |
| Space details Valid | getSpaceDetails() | "<Armory : [[23,19],[10,16]]>" |
| Space neighbors Valid | getSpaceNeighbors(String) | "Billiard Room" |
| World Invalid (Negative Values) | WorldTest([-30,30,"Doctor Lucky's Mansion"])<br><br>WorldTest([30,-30,"Doctor Lucky's Mansion"])<br><br>WorldTest([-30,-30,"Doctor Lucky's Mansion"]) | Throws IllegalArgumentException |
| World Invalid (Check 0) | WorldTest([0,20,"Doctor Lucky's Mansion"])<br><br>WorldTest([10,0,"Doctor Lucky's Mansion"]) | Throws IllegalArgumentException |
| World Invalid (Check Data Type) | WorldTest(["String not allowed here",20,"Doctor Lucky's Mansion"])<br><br>WorldTest([20,"String not allowed here" ,"Doctor Lucky's Mansion"])<br><br>WorldTest([20,10 ,20]) | Throws IllegalArgumentException |
| World Invalid (Create 2nd object/ break singleton rule)<br><br>*Let's consider that one object is already created for World* | WorldTest([36,30,"Doctor Lucky's Mansion"]) | Throws CloneNotSupportedException |

| | | |
|---|---|---|
| To String | toString() | “Name: Doctor Lucky's Mansion, Coordinate A = 30, Coordinate B = 36” |

| **Space Creation** | **Input** | **Output** |
|---|---|---|
| Space Valid | createSpace(new Space(),[[10,15],[12,21]]) | Creates space with new Space() object passed in paramater |
| Space Invalid Overlap<br><br>*Let’s say we have already set up a space at [[10,20],[8,19]]* | createSpace(new Space(), [[11,19],[9,12]]) | Throw IllegalArgumentException |
| Space Invalid outside the world size<br><br>Assume the worldSize to be [26,31] | createSpace(new Space(),[[27,12],[33,23]]) | Throw IllegalArgumentException |
| Space Invalid with negative coordinates | createSpace(new Space(),[[-10,20],[-12,21]]) | Throw IllegalArgumentException |

| **Item Creation** | **Input** | **Output** |
|---|---|---|
| Item Creation Valid | addItems(new Item(), [8, 3, “Crepe Pan”]) | Sets item properties for the new Items() object passed in parameter |
| Item Creation Invalid (Negative) | addItems(new Item(),[-10,-2,”Pan”]) | Throw IllegalArgumentException |
| Item Creation Invalid (Invalid Datatypes) | addItems(new Item(),[“String no allowed here”,”String not allowed here”,10]) | Throw IllegalArgumentException |

| **Space Methods** | **Input** | **Output** |
|---|---|---|
| Get Space Name | getSpaceName() | Returns Space name as a String |
| Get Space Index | getSpaceIndex() | Returns Space Index as int |
| Get Item with Room object Valid | getItemDetails() | Returns Map object of item details |
| To String | toString() | “Name: Armory, Index:10, Items: Revolver” |
| Equals Valid<br>Where all the data members of Space1 and Space2 are same | Space1.equals(Space2) | True |
| Equals Invalid<br>Where all the data members of Space1and Space2 are same | Space1.equals(Space2) | False |
| Hash code Valid<br>Where all the data members of Space1and item2 are same | hashCode(Space1)==hashCode(Space2) | True |
| Hash code Invalid<br>Where all the data members of Space1and Space2 are same | hashCode(Space1)==hashCode(Space2) | False |

| **Item Methods** | **Input** | **Output** |
|---|---|---|
| Get Item Name | getItemName() | “Pan” |
| Get Item Damage | getItemDamage() | 40 |
| Get Item Location | getItemLocation() | 4 |
| To String | toString() | “Name: Pan, Damage: 40, Location: 4” |
| Equals Valid<br>Where all the data members of item1 and item2 are same | Item1.equals(Item2) | True |
| Equals Invalid<br>Where all the data members of item1 and item2 are same | Item1.equals(Item2) | False |
| Hash code Valid<br>Where all the data members of item1 and item2 are same | hashCode(item1)==hashCode(item2) | True |
| Hash code Invalid<br>Where all the data members of item1 and item2 are same | hashCode(item1)==hashCode(item2) | False |

| **Target Creation and Methods** | **Input** | **Output** |
|---|---|---|
| Target Invalid (Negative Health) | Target.createTarget([-50,”Doctor Lucky”]) | Throw IllegalArgumentException |
| Target Invalid (Invalid Datatype) | Target.createTarget([“Doctor”,23]) | Throw IllegalArgumentException |
| Target Valid | Target.createTarget([23 ,“Doctor”]) | “Name: Doctor, Health: 23” |
| Get Target Location For object (Target.createTarget([23 ,“Doctor”]) | getTargetLocation() | 2 |
| Get Target Health For object (Target.createTarget([23 ,“Doctor”]) | getTargetHealth() | 23 |
| Get Target Name For object (Target.createTarget([23 ,“Doctor”]) | getTargetName() | Doctor |