

Design Document - School Playground Game. GUI for Somatic Marker Agent Development (SoMAD)

Jennifer Sandercock

0.1 Executive Summary

The player will be placed in a social environment where they interact with characters that are individual, adaptable and yet have coherent (ie believable and not random) behaviours. The game is compelling because the characters will adapt to the player (and the other characters) and so the player's experience will change the longer they play with these characters.

The player is the new kid at school. They can meet the other school children over lunch break and play games with them. Depending on what the player wants to do they can try and become the most liked person at school, or they can try and become the least liked person.

The game is based on gossiping. Characters can exchange pieces of gossip and create gossip by doing actions at the end of every interaction. They can also watch or listen in on other conversations. Characters can lie about gossip.

The unique and adaptable behaviours of the characters are built up automatically during an initialisation period (before the player joins in). The programmer only needs to decide on:

- The importance/weight the NPC places on maximising or minimising each soft goal
- Values for emotionality - thresholds, stamina, decay, uptake.

These values can be allocated randomly to make it even easier.

Contents

Chapter

0.1	Executive Summary	i
1	Game Play/Game Mechanics	1
1.1	How overall game starts	2
1.2	Conversations	3
1.2.1	Gossiping	3
1.2.2	Actions	4
1.3	Listen In	5
1.4	Watch	5
1.5	Shout	5
1.6	Messages from the Environment	5
1.7	Diary	6
1.8	How overall game finishes	6
2	Game AI	7
3	The game environment	10
4	Game Elements	14
4.1	Characters/NPCs	14
5	System Menus and Questionnaire	15
5.1	Diary	15
5.2	Questionnaire	15
6	Requirements and Input/Output	16

Chapter 1

Game Play/Game Mechanics

- The player has a top down view of the world. The main screen is a close up of the current room, there is a smaller map/radar to see what is happening in the other rooms.
- The player is able to walk around a world by using a keyboard or mouse or both.
 - * They can use the arrow keys to move forwards, backwards, or sideways
 - * Or they can use the mouse to point and click to where they want to move to.
 - * If the player right clicks on an object or the ground they can select between:
 - Move here
 - Sit here
 - Cancel
 - * Player character pathfinding doesn't have to be great. But want the player to be able to point to a location - which could be out of this room in another room, and have their character move there. When the avatar can't reach the place clicked, the avatar should just walk to the closest point it can to where the person clicked, and then stop.
 - * To move in one direction the player has to hold down the forward button. The side buttons turn the avatar (ie it doesn't walk sideways).
 - * Since there are no hills or anything in the environment, the avatar moves at a constant speed when the player is pushing the forward button. When it bumps into an object it will turn in a random(?) direction so that the user can keep pushing the forward button and get somewhere. No one is allowed to walk onto the chasey pitch unless they are playing the game.
- Player can click on the legend so that it will toggle between hidden and visible at the bottom of the screen. (Optional... only if this is easy to do in the GUI, otherwise permanently there is fine).
- The player interacts with people in the environment by:

Selecting them directly: Click on a character directly and a menu pops up offering options for the user. If the character is not already in a conversation the option will only be "talk". But if the character is already in a conversation the player can "talk", "watch" or "listen".

Shouting out: Click on the side column to shout out to everyone in the area.

Accepting an invitation to talk: A character asks the player to talk to them, so a pop-up box should appear so that the player can accept or reject. If they are in a conversation already and they accept, then they will have to finish their initial conversation first.

Clearing notices: Other agents can send messages to the player such as shouts and "Listen in", "Watch" or "Go away". These last two should encourage/discourage the player from listening in or watching another conversation. The player should be able to read the notices, but clear them away once they have been read.

- The player cannot move the other characters.
- The player is limited to the same choices as the characters, although characters can explicitly choose to lie.
- The player can pause the game at any point, but can't save.
- If the player accesses the help menu the game is paused.
- There is a access to the full help menu available during normal play, and when the window is pulled up it starts on the first page of the help menu
- When in an activity if the player asks for help the full help menu opens with info about the current activity as the first thing they see

1.1 How overall game starts

- The player starts in a randomly allocated room.
- They are able to walk around the room and to other rooms by using the mouse to click on a destination - they are then teleported to that location.
- They can find out a bit about the other characters by clicking on them.
- They interact with characters in a conversation
- The aim of the game is to make the most friends, or make the most enemies.

See the goal/plan hierarchy diagram 2.1 to see all choices available to the computer controlled characters.

1.2 Conversations

Conversations have two main components: gossiping and actions.

While in a conversation the player should still be able to see the main game screen. That way they can tell which other characters may be listening in or watching the conversation.

1.2.1 Gossiping

- Players take turns at giving and receiving gossip.
- If it is the player's turn they can choose between:

Telling Gossip Player must choose from a pull-down list for each item:

- (1) Who did the action
- (2) Who received the action or who was it about if it was a shout
- (3) What was the action (hug, shake hands, wave, ignore, insult). For a shout, what was shouted (liar/truth teller/love/hate).
- (4) Whether it was accepted or rejected (not valid for a shout)
- (5) Whether this is a true or a false piece of gossip.

Asking for gossip on someone: The player selects from a list who they would like to find gossip about.

Telling someone else to move: The player can ask another character (not in the conversation) to listen in to or watch their conversation. If a character in the area is listening in or watching already the player can ask them to stop. For this the list of possible characters is limited to the characters in this area. Note: don't need to get a response from the character in the conversation when do this.

Passing on your turn: This means that the character will start telling gossip

Ending the conversation: This will mean that the player will need to perform an action and get reply and etc... see below

- After the agent has done their turn they wait to get a reply from the character they are talking to.
- If it is not the player's turn, they must wait to hear something from the other character and respond. Then they can choose to end the conversation, but they must respond first.
- If the other character has sent through some gossip information the player can choose between:

- * Say "I'm not telling you anything"
 - * Agree with gossip told
 - * Disagree with gossip told
 - * Say "I don't know"
- If the other player has sent through a request for information on another character (x) the player chooses between:
 - * Say "I'm not telling you anything"
 - * Say "I don't know anything about them"
 - * Tell them something (need to choose what to say):
 - Was 'x' the giver or receiver?
 - Who was the other person?
 - What happened/was said
 - Was it accepted or rejected (only if not a shout action)
 - Is this true or false?

1.2.2 Actions

- The character or the player can decide to finish the conversation first.
- The person who decides to end the game sends a message to the other one with an action.
- The receiver can accept or reject the action.
- The receiver then sends its own action to the original player, who can accept or reject that.
- Actions are:
 - * Hug
 - * Shake hands
 - * Wave goodbye
 - * Ignore
 - * Insult/Be mean
- These actions then become new gossip that can be passed around.
- Therefore in every conversation two of these "gossip" items are created. the

1.3 Listen In

- A player can listen in on an existing conversation if they click on one of the characters in a conversation and then select “Listen In”.
- When listening in the player should get a text scrolling screen to show the conversation. This will show what is said during the gossip stage of the conversation
- The player won’t be able to see/hear/know about what action each of the characters took.

1.4 Watch

- A player can watch an existing conversation if they click on one of the characters in a conversation and then select “Watch”.
- The player won’t be able to see/hear/know about what gossip is being exchanged, only the action and responses at the end.
- As a result the player will probably have to be patient and wait, so maybe have some sort of animation or change or something to show that the game is still working and hasn’t frozen.

1.5 Shout

- When the player isn’t doing something else they are able to shout out to the entire area a statement.
- The statement is about a character, x, and says one of four things:
 - * X is a liar
 - * X tells the truth
 - * I love X
 - * I hate X
- if the player is not in the area when a shout is sent, they will not receive the shout.
- The play should be able to read the shout and then clear the message away.

1.6 Messages from the Environment

- Messages from other players can be one of:
 - (1) Shout: see above for details;

- (2) Request to start a conversation: Can be sent while the player is in a conversation already. If the player agrees to this conversation, they must finish the one they're doing currently with the action swapping.
- (3) Request to move: Can be sent from any character that is in the same area as the player. The player just needs to acknowledge this message. How they act on it is entirely up to them.

1.7 Diary

- Because it is probably going to be hard for the player to remember all the pieces of gossip and shouts and etc... It makes sense to store all information sent to the player in a scrolling text file so they can see what has been going on.
- this should be accessed in a similar way to the help menu.
- The diary should be able to be accessed at ANY time throughout the game and game play should be paused while the player is accessing it.

1.8 How overall game finishes

- After 15 minutes the player is able to leave the game. They can continue if they want, but they should be prompted that 15 minutes is up and asked whether they would like to leave.
- When they leave they are asked to nominate their best friend and their worst enemy
- The "result" of the game is displayed. This shows the most and least liked person, and the player's rank within this list.

Chapter 2

Game AI

I will be doing all of the reasoning behind the agents. This next bit explains a bit about my thesis if you're interested.

- This is the main focus of the game. That is the way the agents make decisions on what to do next or how to respond to input from the environment/other characters.
- Each time an NPC comes to a somatic marker (SM) decision point they:
 - * Find out what their current situation is:
 - How close are they to achieving each of their soft goals
 - What their current emotion is
 - * This current situation allows them to pull out a SM value for each of the applicable plans
 - * It then groups the plans into plans it:
 - Should try
 - Shouldn't try
 - Doesn't care
 - * Then it will randomly choose a plan to execute from the should try list, or the doesn't care list.
- After the agent has finished execution of an activity (ie a mini-game) it will evaluate how successful the plans it chose were in making it get closer to its soft goals. This will cause it to update its SM value if it finds itself in the same situation again.

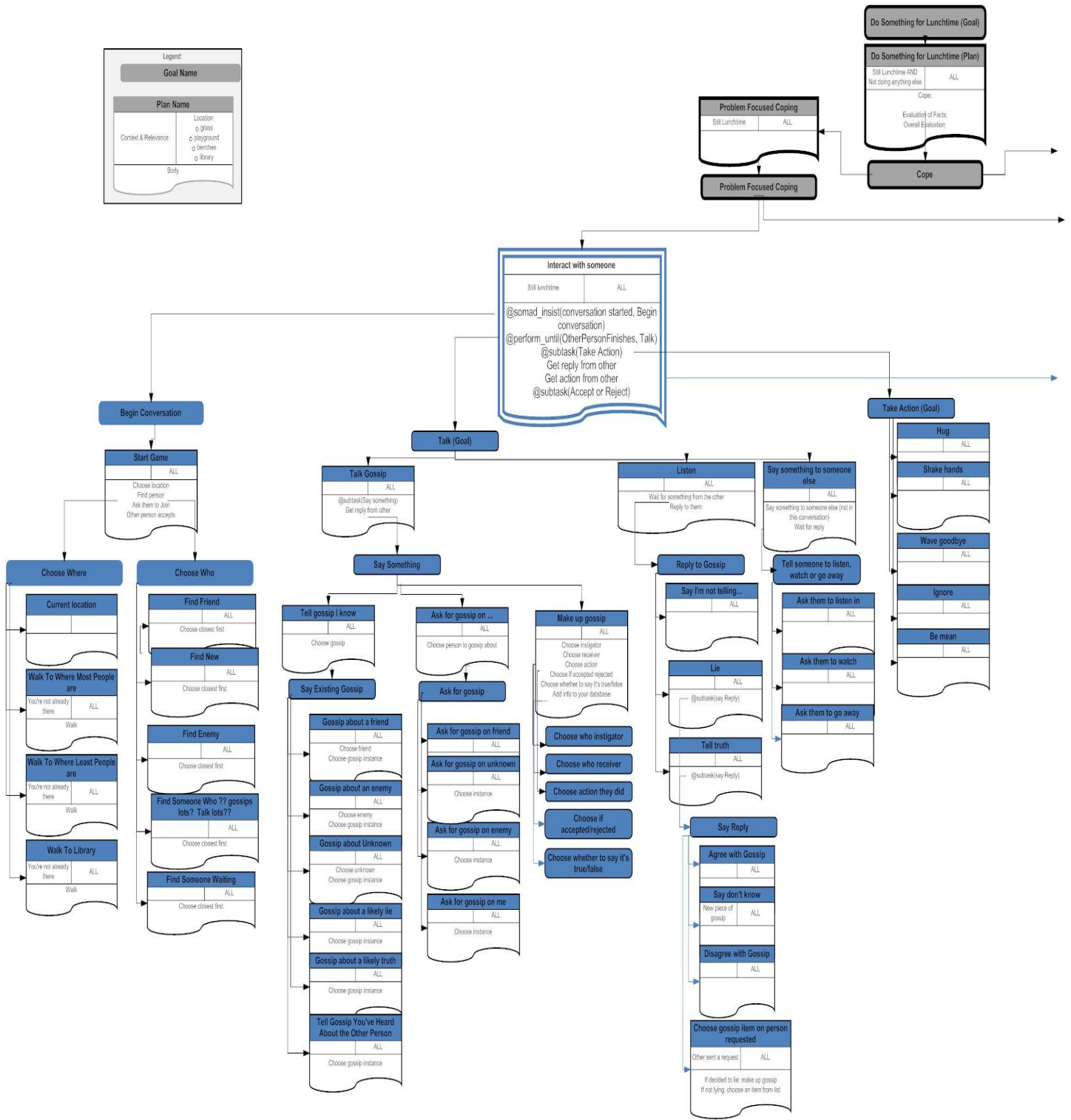
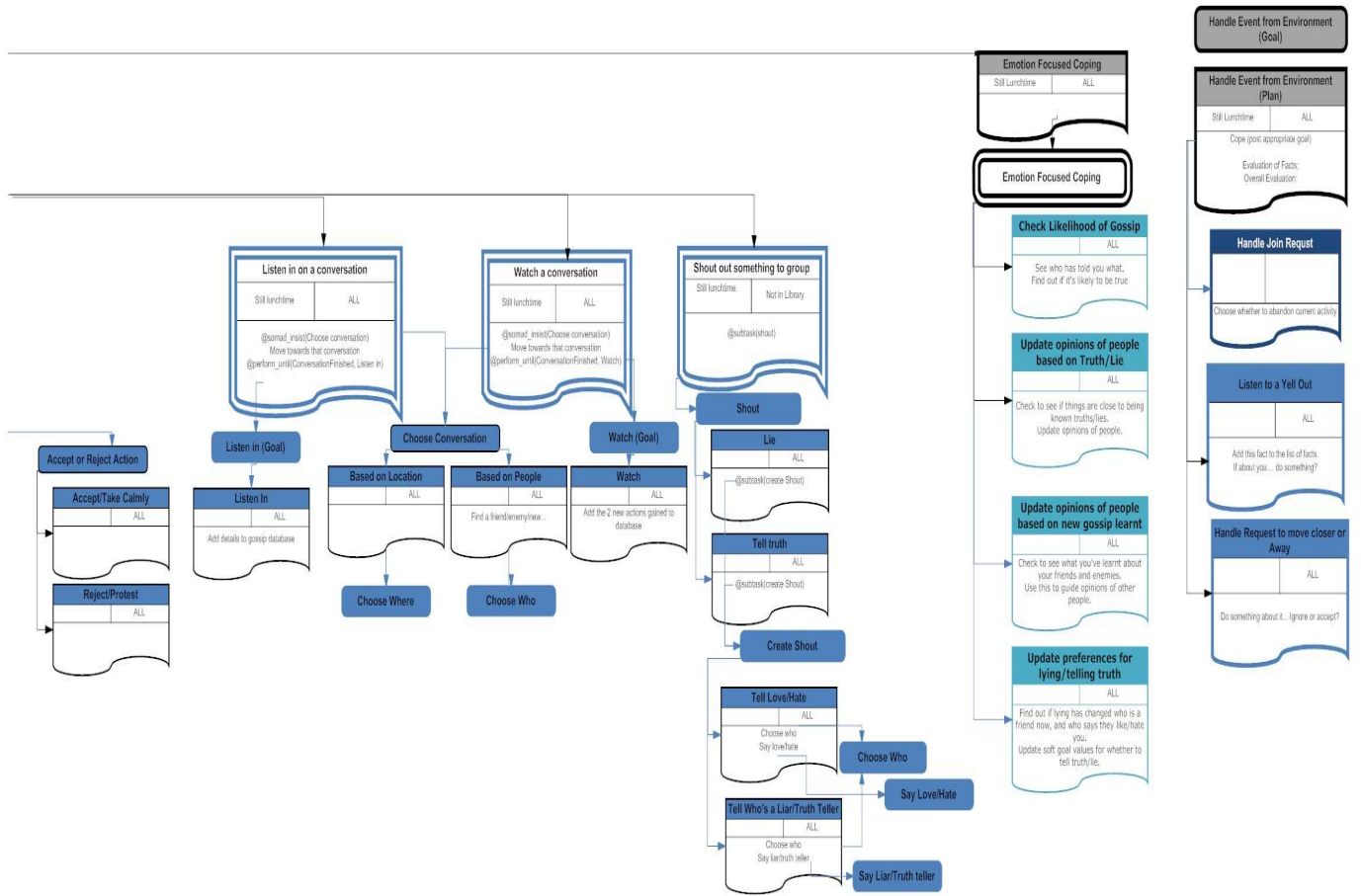


Figure 2.1: The Goal/Plan Hierarchy for the game (continued on next page)



Chapter 3

The game environment

In the next few pages are sample photos that are supposed to represent screen-shots taken at different times in the game. Not all the cases are covered. But hopefully enough so that you can understand the intent.

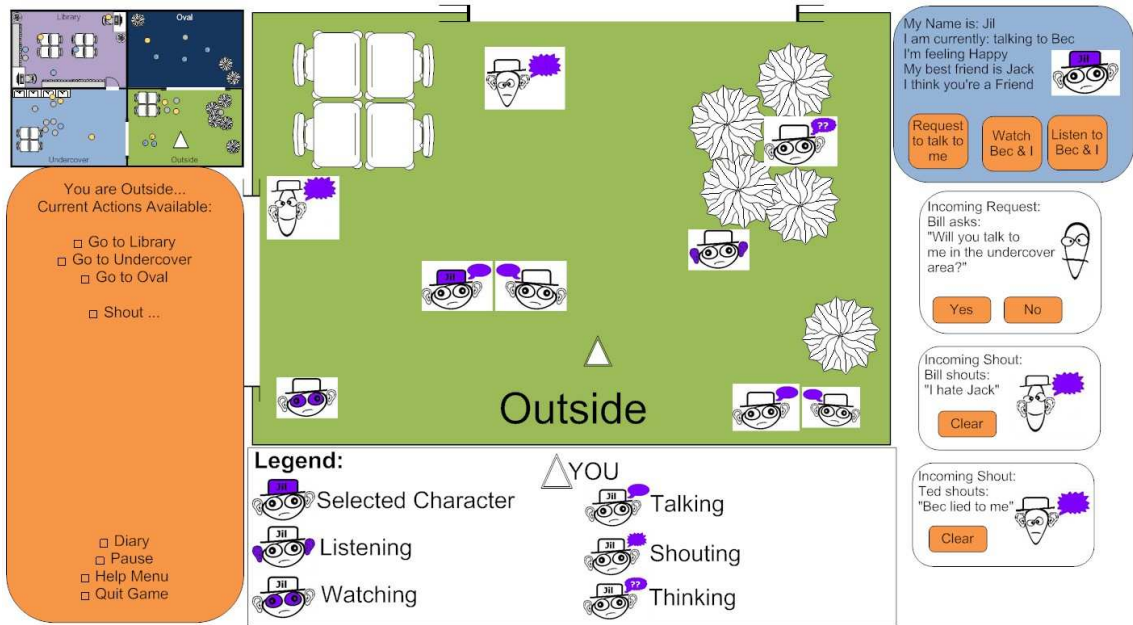


Figure 3.1: What the game looks like when walking around normally. Note: there is a request from a character on the right and two other characters have shouted out to the room. The player has selected a character (seen by the highlighted hat) and they are given some information about that character.

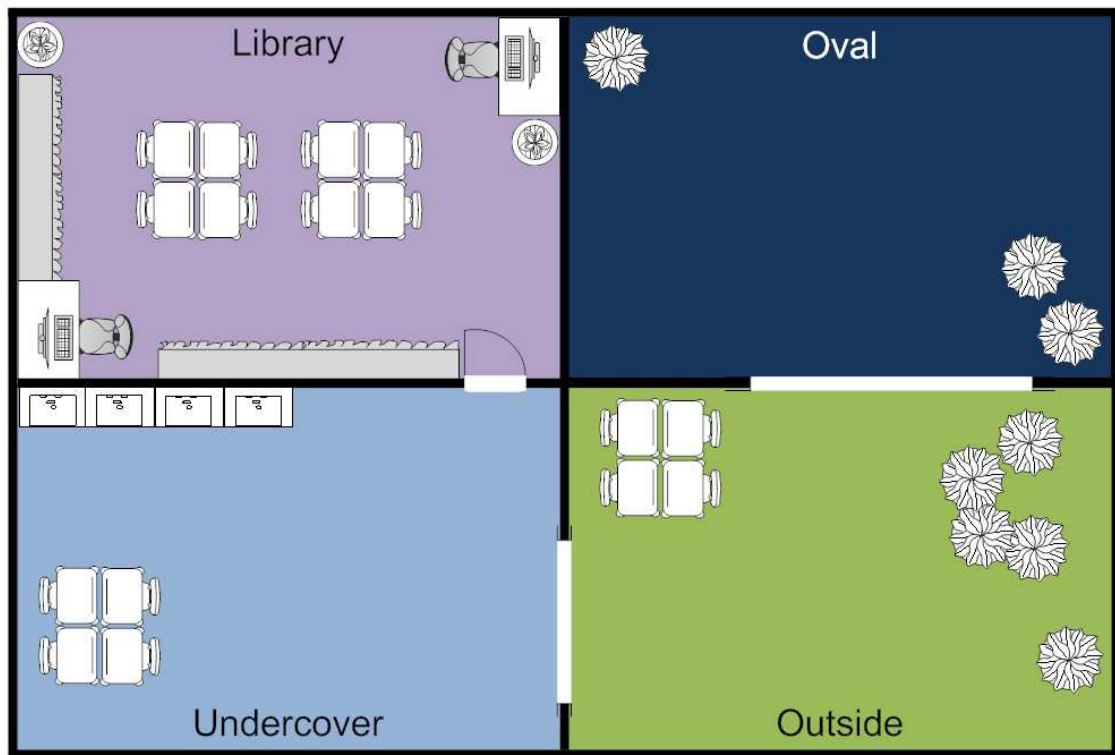


Figure 3.2: The floor plan of the game.

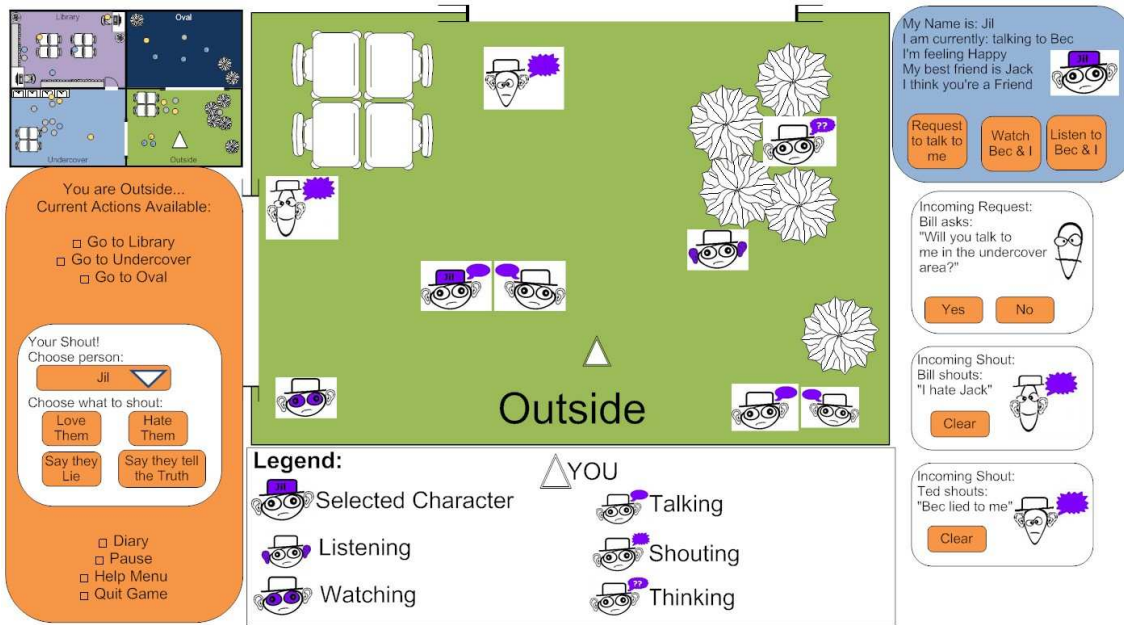


Figure 3.3: After the player has chosen "Shout". They are given options on what their shout will be. The box with an arrow represents a pull-down menu that will have the names of all the characters. Should have a cancel button too.

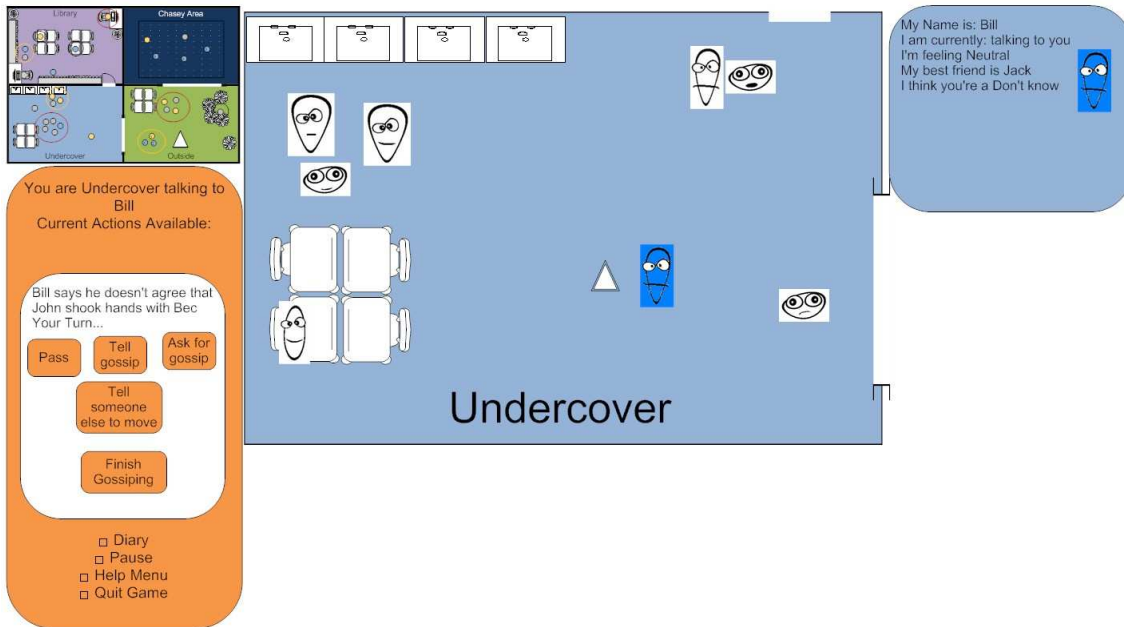


Figure 3.4: In this case the player is in a conversation with Bill. The top line shows how Bill has reacted to the last thing the player said. The player is offered a number of options since it is the player's turn to tell some gossip.

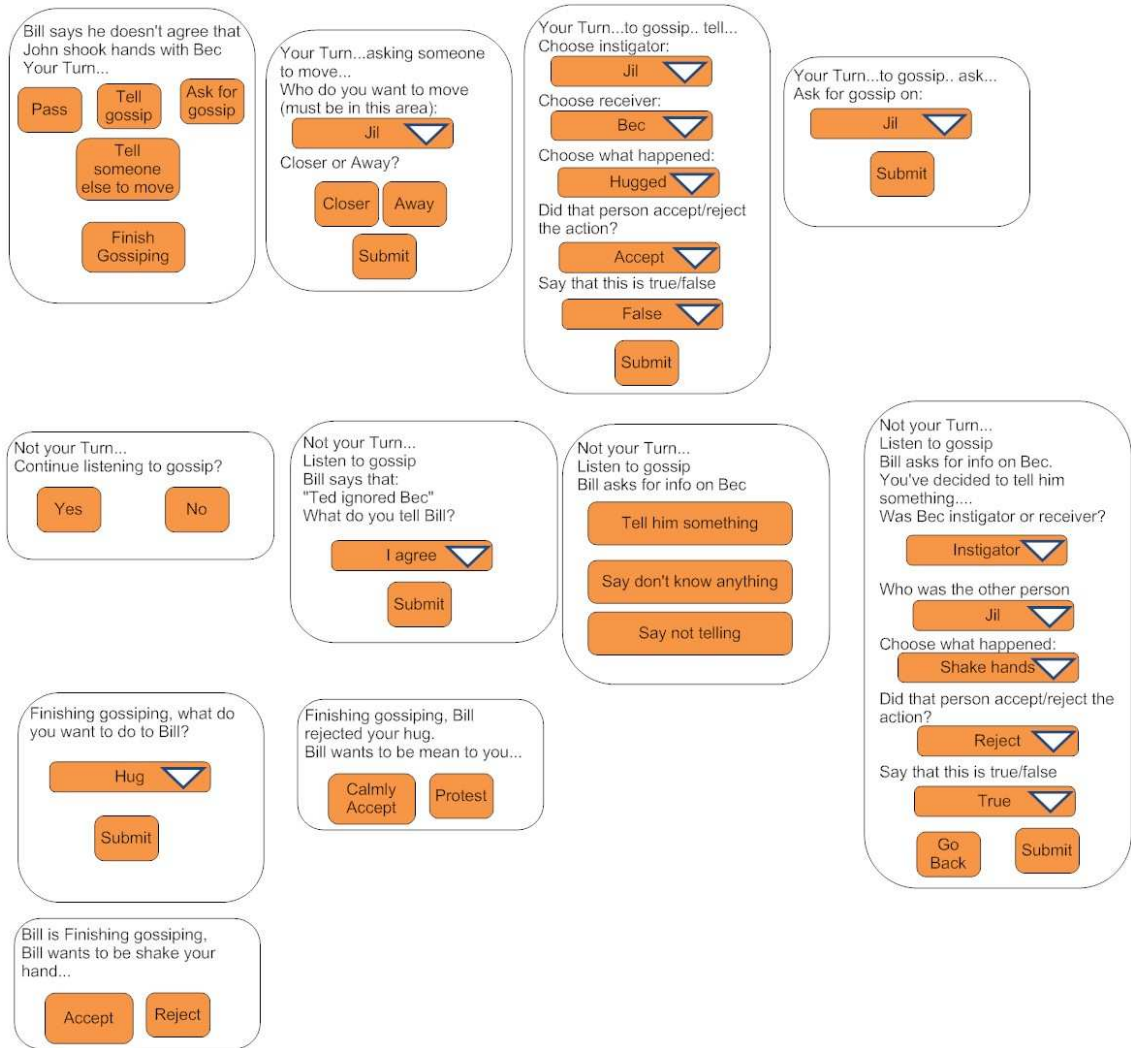


Figure 3.5: While in the conversation with Bill the player can get a number of different choices at different stages. These images show some potential things that could happen. The player should be able to quit or cancel their previous choice if a message hasn't been sent out yet.

Chapter 4

Game Elements

Objects in the game are inanimate. They can't move, be picked up or walked over.

4.1 Characters/NPCs

- The characters should look “unique”. The main focus of this work is to create characters that are different. I'm working on them having different reasoning. But if the characters don't look different to the player then it will be a lot more difficult to see the differences.
- There will be between 20-30 characters in the game.
- Each face should be slightly different. I need a minimum of 20 different faces. Ideally there would be 60 different faces since users will be playing the game 3 times with different characters. But that's probably not possible. Even if the differences are small between characters, it'd be great
- All characters have a hat on their head with their name. This hat could change for different characters (therefore giving another point of difference), or it can be the same. Whatever's easiest
- Characters need to be able to show their emotion somehow. This could be a smile, frown or neutral mouth. Or it could be 3 different colours of their face image. Whatever's easiest.

Chapter 5

System Menus and Questionnaire

It is not possible to save a game. The player can pause a game, and after the game is over they can save their questionnaire, but the game itself cannot be saved.

A help menu is needed. This will tell the player how to move in the environment, the conversation process, and information about the questionnaire. I will build the content of the help menu, but if you could provide the template or access from the GUI that would be great.

5.1 Diary

The diary shows what has been said to the player and what they have said. That is a text based list of what they've been doing.

5.2 Questionnaire

After each play session the user should be prompted to answer a questionnaire. Other than a graphic template similar to the help menu, I think I should be able to get this worked out myself.

Chapter 6

Requirements and Input/Output

I am building the core AI reasoning for the NPCs. Most of the time the interaction between the GUI and the AI will be through the game area agents. They will hold a list of who is in each area, what they are doing and with whom (if appropriate). If the list is changed a message will be sent to the GUI notifying of the change.

The area agent manages its area and will inform the GUI when one of the characters changes:

- location - if specifying to move to somewhere, may need to check with GUI to make sure can get there.
- Activity it's doing
- Who it's playing with
- Whether it is happy/sad/neutral

A suitable interface needs to be set up here. So if you just let me know what format you'd like the messages in, that would be great. The GUI will need to let the area manager's know which area the player is in currently and what they are doing. The GUI will also need to access the list of who is in a current area so that the appropriate drop down lists can be built for the player to use and to work out where to place them on the screen. Characters who are talking to each other should be placed next to each other, otherwise characters can be placed randomly within the area they are in.

All messages from a character to another character will be unseen by the player, unless the player is listening in or watching. In that case the GUI will be sent what is in the messages between the other characters, where appropriate.

Shout messages will be sent to everyone within a certain area. So if the player is in that area then they will be sent a shout message which should appear somehow on their screen.

The player should be able to request to talk to someone. This should involve sending a message directly to the character and a notice of where the player would like to talk (this can be a different location to where the player is currently).

When the player is in a conversation with a character the GUI will probably need to interact directly with the character, rather than through the area agent. The character will provide messages of what it's said and responses to what the player has said.

Pathfinding... Hopefully won't be needed. Teleporting the character from one spot to the other would be good. Although hopefully this won't be too confusing for the player when they are trying to follow their own avatar on the screen.

Ideally I would like it to be able to be run through a web interface. Or perhaps simply downloadable.

The GUI will need to notify the AI/simulation control when the player wants to quit or pause the game.

The GUI should support a template for a help menu. And should also automatically fill in the diary option.