

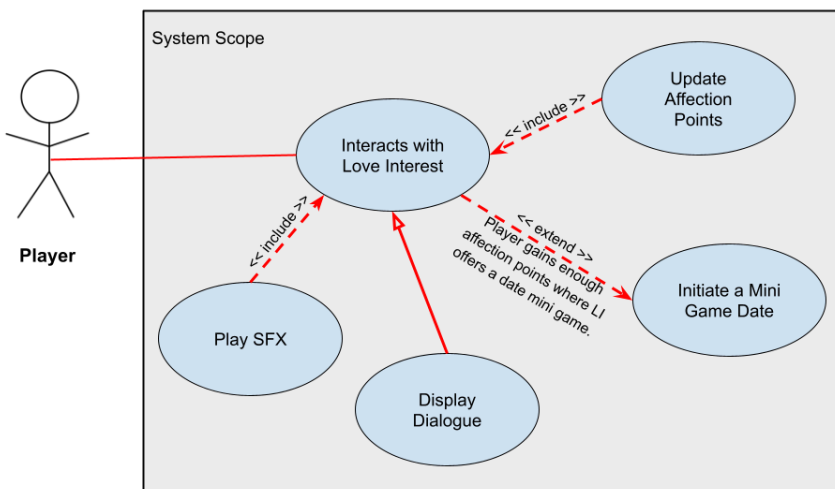
1. Brief Introduction ___/3

My feature for the game is the handling interaction with the superclass of Peanuts love interest. Each love interest will have a graphic, sound effects, text dialogue with the user to interact with, personal preferences which determine their affection levels for the main player, and the ability to initiate a minigame date with the player. The LI database which is separate from the character class will contain their graphics and affection points. The audio functions are a separate entity and will retrieve the sound effects from the LI database, and the character class can call these audio functions.

I am in charge of creating the subclasses of each specific Peanuts themed love interest. I will have to program how their affection points increase and/or decrease, the updating the affection levels in the LI database, calling the sound effects for the LIs, their text dialogues that appear and subsequently where the player in their conversation with the LI, and the option for minigames appear based on the main player input.

2. Use Case Diagram with Scenario ___/14

a. Use Case 1



b. Scenarios

i. Scenario 1

Name: Play SFX

Summary: The sound effects that play for each specific LI.

Actors: LI Character, player

Preconditions: LI must be interacted with by the player.

Basic Sequence:

Step 1: Dialogue bubble will be displayed.

Step 2: LI sound effects start playing.

Step 3: LI sound effects stop playing after set time.

Exceptions:

Step 1: Schroeder character can play sound effects for piano playing without dialogue box appearing.

Step 3: If the user exits out, the sound effect should stop then instead of waiting a set time.

Postconditions: Audio is played successfully during a given time frame.

Priority: 2*

ID: SFX2

ii. Scenario 2

Name: Display Dialogue

Summary: Dialogue text bubbles will be shown from the LI with 2-3 choice responses that the player can respond with.

Actors: LI Character, player

Preconditions: LI must be interacted with by the player.

Basic Sequence:

Step 1: Dialogue bubble will be displayed.

Step 2: Response options will appear below the LI text bubble after a set time (similar time from to the LI SFX).

Step 3: Dialogue bubble will disappear after the player selects their response.

Step 4: Steps 1-3 will loop with different dialogue options or until player reaches a minigame date as discussed in postconditions, or process will end as discussed in exceptions

Exceptions:

Step 1.1: If all 'bad' dialogue responses are chosen, the LI will not want to interact with the player again.

Step 1.2: If the user exits out, the dialogue will no longer display, but the LI will save what dialogue they were on.

Postconditions: Dialogue was successfully displayed and the player was able to select a response. Based on this response, the LI has the next dialogue or minigame queued up.

Priority: 1*

ID: DDB2

iii. Scenario 3

Name: Update Affection Points

Summary: Affection points are influenced by the player's interactions with the LI.

Actors: LI Character, player

Preconditions: LI must be interacted with by the player.

Basic Sequence:

Step 1: Good dialogue response is chosen by the player.

Step 2: Affection points are increased.

Step 3: Bad dialogue response is chosen by the player.

Step 4: Affection points are decreased

Exceptions:

Step 2: If affection points are increased to maximum, player wins (and wins over the LI).

Step 4: If affection points go below zero, the LI will stop interacting with the player.

Postconditions: LI's affection points have changed from before the interaction with the player.

Priority: 1*

ID: UAF2

iv. Scenario 4

Name: Initiate a Minigame Date

Summary: After enough good dialogue responses from the character, the LI will ask the player to go on a minigame date.

Actors: LI Character, player, Minigame Structure

Preconditions: LI must be interacted with by the player and the player must have selected good dialogue responses.

Basic Sequence:

Step 1: LI asks the player if they want to play a minigame.

Step 2: Player accepts and begins minigame

Step 3: Minigame is played, player either wins or loses.

Step 4: Affection points updated based on game result and LI preferences.

Exceptions:

Step 1: If the player does not want to play the minigame, they can refuse.

Postconditions: Player has played a minigame date with the LI, which was either positive or negative.

Priority: 1*

ID: MGD2

***The priorities are 1 = essential and 2 = nice to have**

3. Data Flow Diagrams from Level 0 to Process Description for my Feature

a. Data Flow Diagrams

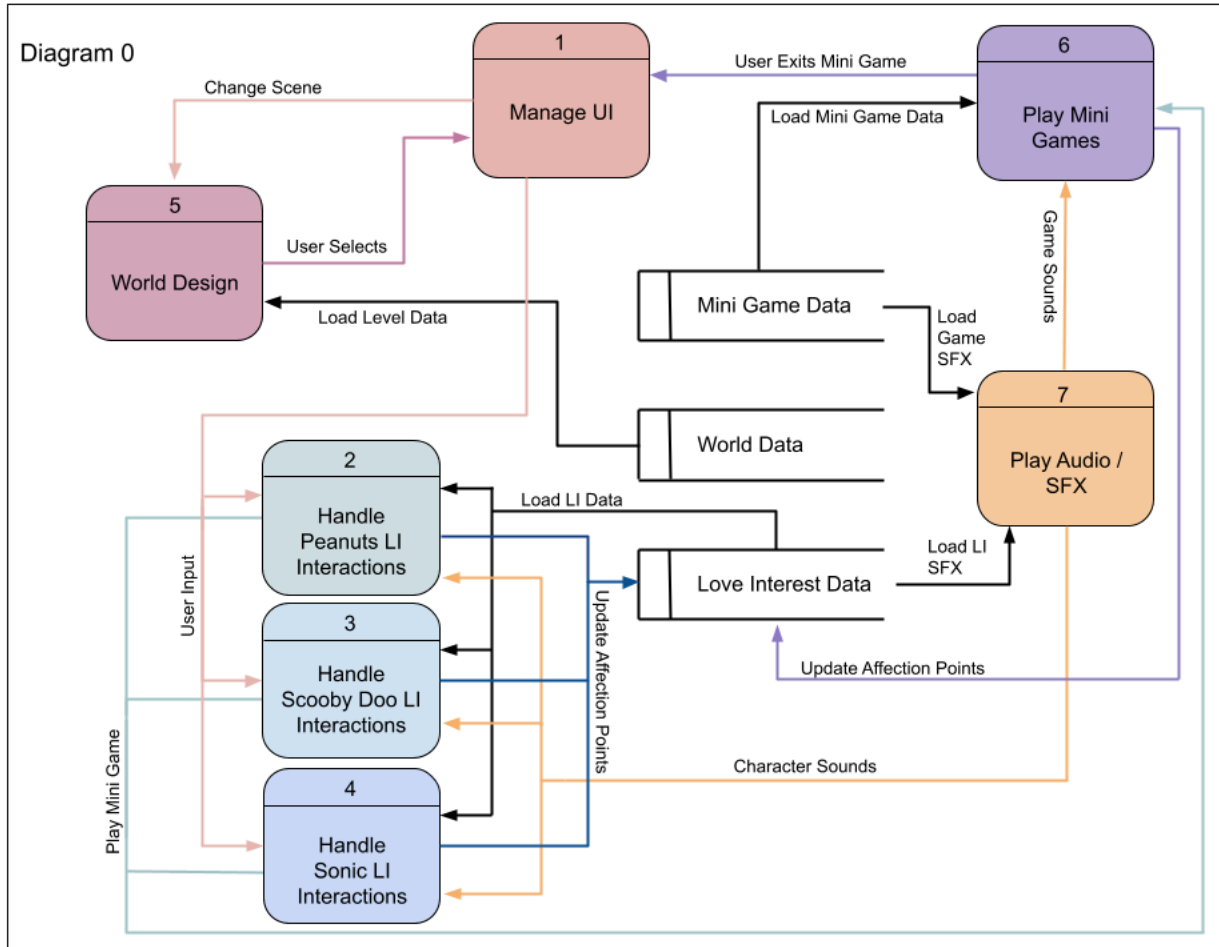


Diagram for 2

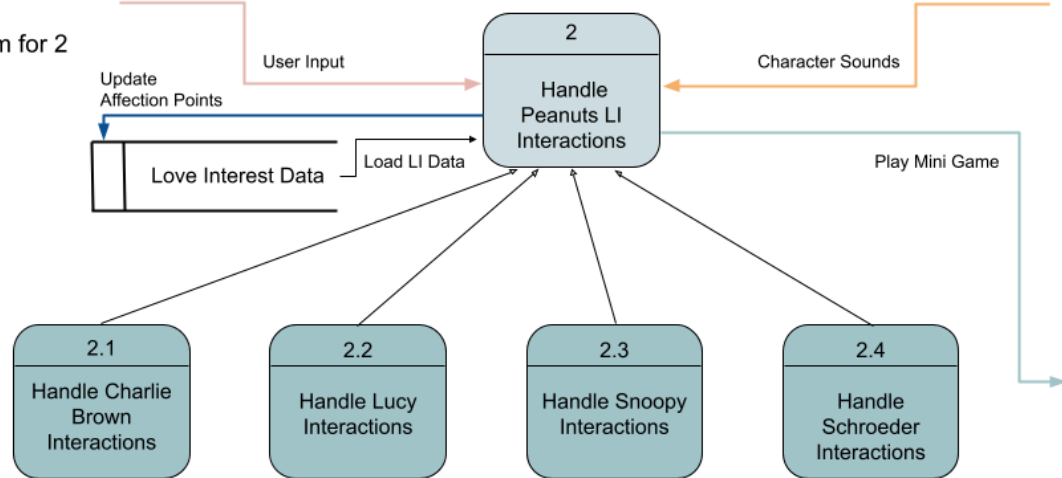
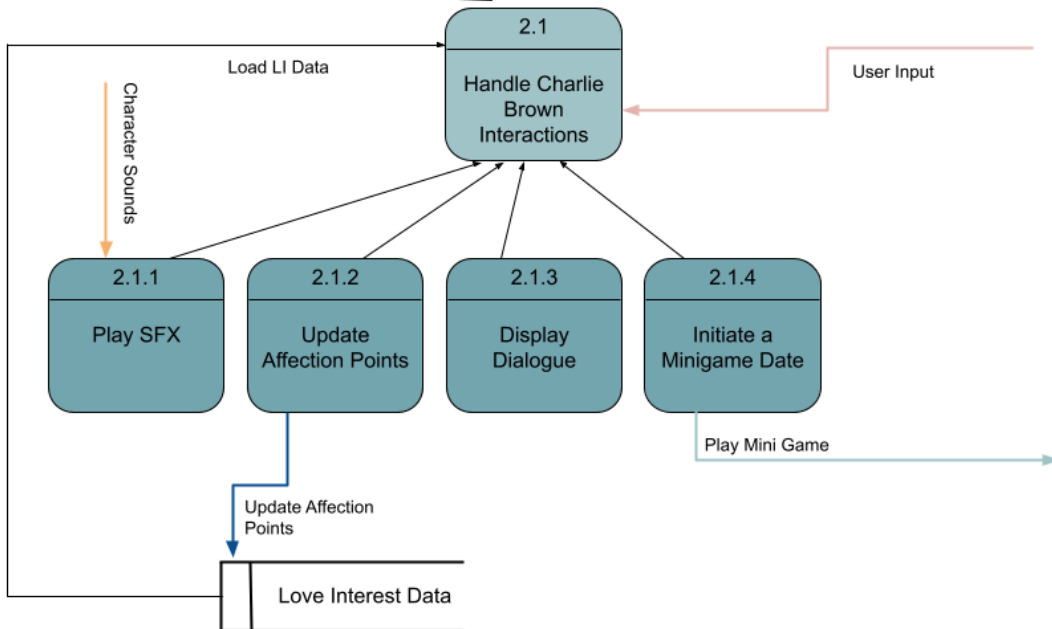


Diagram for 2.1



b. Process Descriptions

For 2.2.2 Update Affection Points:

```
//user hits some button to interact with the Charlie Brown
//the user is usually hitting a button in response to dialogue being
//displayed, but could also just be approaching him
IF (user response == what Charlie Brown wanted to hear most) {
    CharlieBrown.updateAffectionPoints(x)
    //x would be the number of points to increase by
}
ELSE IF (user response == Charlie Brown will accept this response){
    CharlieBrown.updateAffectionPoints(y)
    //y would be the number of points to increase by (less than x)
}
//otherwise, user response != what Charlie Brown wanted to hear
ELSE {
    CharlieBrown.updateAffectionPoints(z)
    //z would be a negative number to decrease the points
}
//note: the next dialogue to display or the option to initiate a game is
//also determine by user input
```

The function updateAffectionPoints() takes an int to change the affection points by. It updates the Affection Points for the Charlie Brown instance that is stored in the LI database. The function is described below:

```
Void updateAffectionPoints(int x){
    CharlieBrown.AffectionPoints += x
}
```

4. Acceptance Tests __/9

The LI interactions with the player are determined by the player's input via responses to prompts. Depending on whether the user response was positive or not, the love interest's (ex Charlie Brown's) affection points will increase or decrease. This will also impact what next dialogue is displayed or if they will go on

a minigame date. Therefore, the Handle __ (a Peanuts LI character) Interaction needs to be tested on whether or not it progresses correctly from user input.

Testing the Feature Response to User Input:

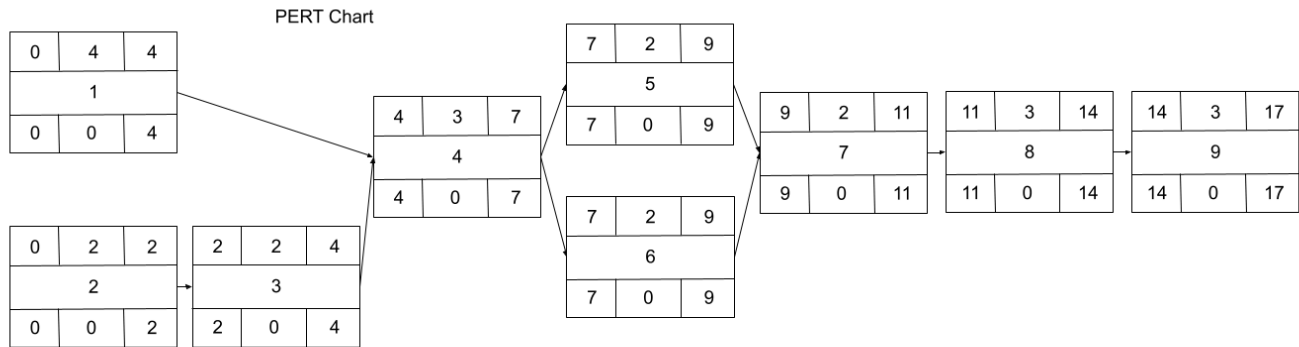
User Input	Result on Affection Points	Result on Next Dialogue Display	Notes
Charlie's Favorite Response	Increase Charlie Brown's affection points in the LI database by a lot	Next dialogue displayed is one of the more positive dialogues.	If the user chooses the 'right' response, they should see an increase in Charlie Brown's affection, and be able to continue a positive 'conversatoin'
Charlie's Accepted Response	Only increase Charlie Brown's affection points a little	Next dialogue displayed is one of the more positive dialogues	If the user is choosing the midlevel responses, they aren't gaining any points but are not losing any points or opportunities for more points.
Charlie's Disliked Response	Decrease Charlie Brown's affection points	Next dialogue displayed is one of the less positive dialogues.	If the user chooses the 'wrong' response, they should see negative affection results, and get dialogue from Charlie Brown that shows he is disappointed.

5. Timeline __/10

a. Work Items

Task	Duration (hours)	Predecessor Task(s)
1. Create Dialogue Prompts and Flow Design for each LI	4	-
2. Select Graphics and Sound for each LI	2	-
3. Implement Play SFX functionality	2	2
4. Implement Dialogue Prompt Display and Reaction to User Input (Affection Point Update / Dialogue Sequence)	3	1,3
5. Set Up Minigame Date Initiating	2	4
6. Implement Different Difficulty Levels for the Characters	2	4
7. Character Touchup	2	5,6
8. Integration	3	7
9. Testing	3	8

b. Pert Diagram



c. Gantt Timeline

