

## 1. Brief introduction \_/3

My feature for our game will be the music and sounds as well as the scoring system for our dating simulator. Our dating simulator will have considerably different characters for you to romance, so I will oversee making and applying the background music, character sounds, and mini game sounds. For the scoring system, each romanceable character will have a bar made up of hearts that fill up as you romance them through dialogue choices and mini games.

## 2. Use case diagram with scenario \_14

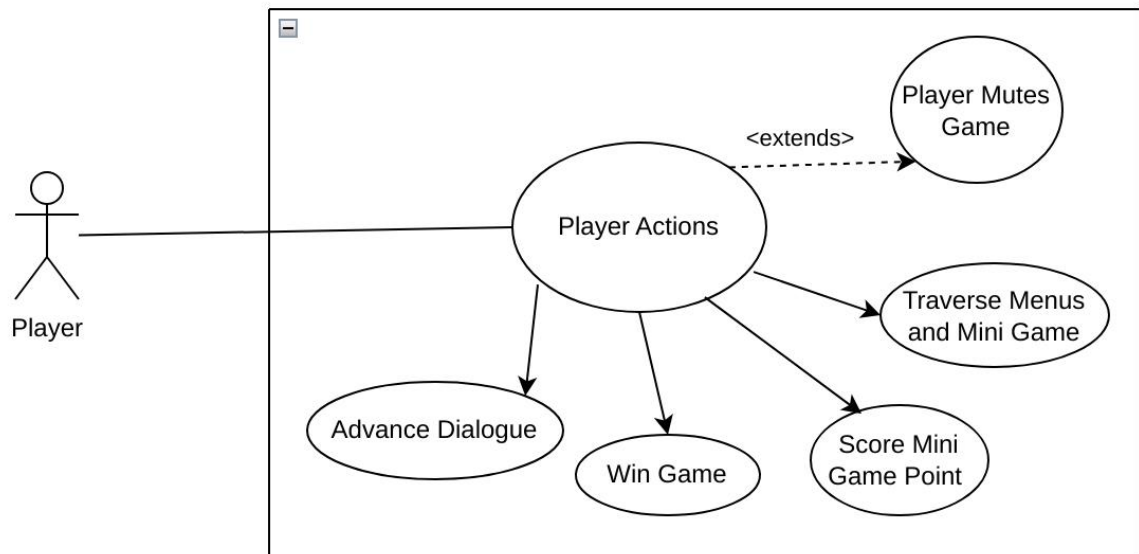
[Use the lecture notes in class.

Ensure you have at least one exception case, and that the <<extend>> matches up with the Exceptions in your scenario, and the Exception step matches your Basic Sequence step.

Also include an <<include>> that is a suitable candidate for dynamic binding]

Example:

### Use Case Diagrams



### Scenarios

**Name:** Advance Dialogue

**Summary:** The player will press a dialogue option with a button and the character you are romancing will play a sound.

**Actors:** Player.

**Preconditions:** Game has been installed and initialized.

**Basic sequence:**

**Step 1:** Player opens the game.

**Step 2:** Player sees the game menu and selects play.

**Step 3:** Player sees the opening screen and chooses which character to pursue.

**Step 4:** Dialogue with character begins.

**Step 5:** Press a dialogue button and the characters sound will play.

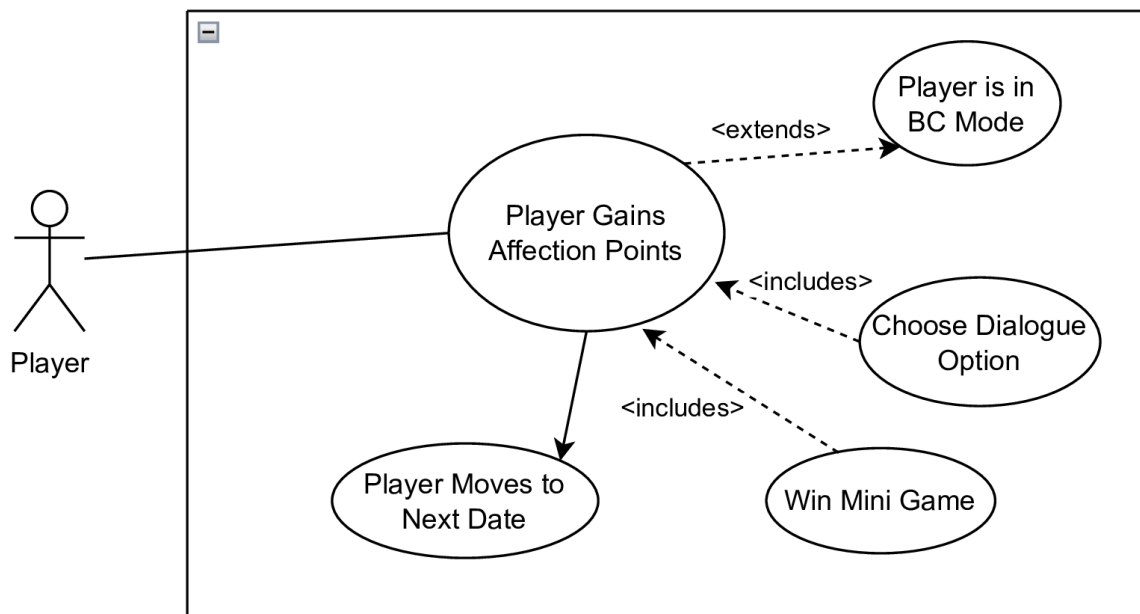
**Exceptions:**

1: Player mutes the game with button on options screen: ignore audio

**Post conditions:** The player continues play of the game

**Priority:** 2

**ID:** C03.1



## Scenarios

**Name:** Choose Dialogue Option

**Summary:** The player will press a dialogue option with a button and the character you are romancing might gain an affection point towards the player.

**Actors:** Player.

**Preconditions:** Game has been installed and initialized.

**Basic sequence:**

**Step 1:** Player opens the game.

**Step 2:** Player sees the game menu and selects play.

**Step 3:** Player sees the opening screen and chooses which character to pursue.

**Step 4:** Dialogue with character begins.

**Step 5:** Press a dialogue button and affection points might be given out.

**Exceptions:**

1: Player is in BC mode: all checks for affection points will be ignored

**Post conditions:** The player continues play of the game

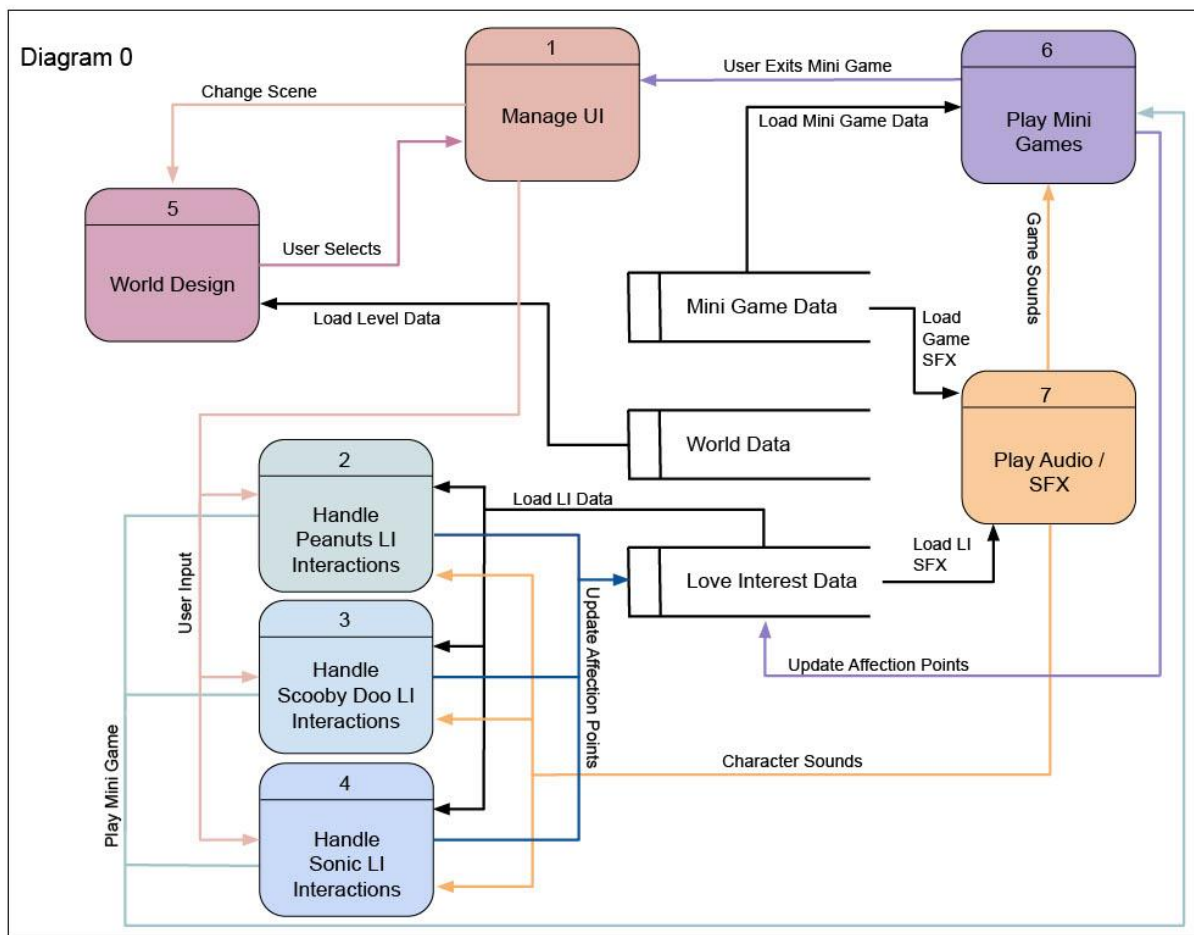
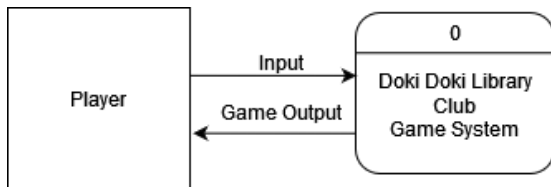
**Priority:** 1

**ID:** C03.2

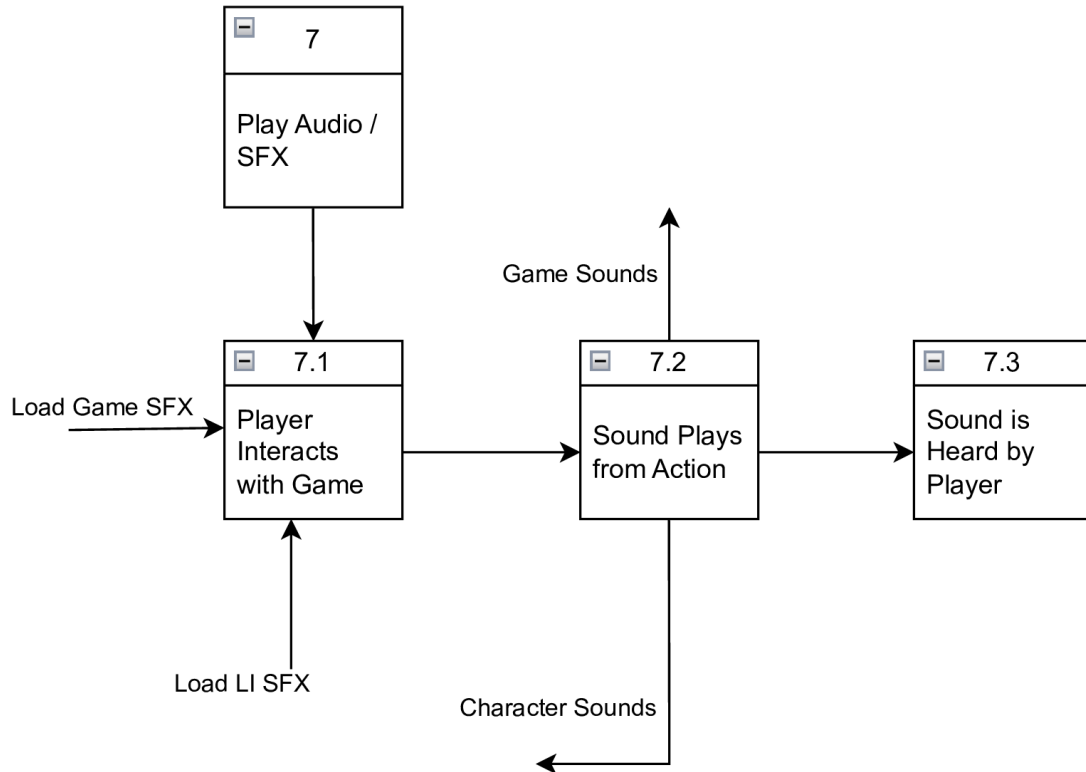
### 3. Data Flow diagram(s) from Level 0 to process description for your feature \_\_\_\_14

#### Data Flow Diagrams

Context Diagram:



I oversee process 7: Play Audio / SFX



### Process Descriptions

When the player interacts with any sort of button, goes to a new area in the game, or scores a point in a minigame, a sound will be triggered. The code below will show all the main ways sound will be triggered in the game, in order being the background music, menu buttons, dialogue for each character, and scoring in a minigame.

Pseudo Code:

```

while(!ScenesFinished){
    If (!audio.isPlaying){
        Audio.play(currentSceneMusic);
    }
}

```

```

if (Input.GetButtonDown("MenuButton")) {
    PlayMenuSound();
}

```

```

GetCurrentCharacter();
if (Input.GetButtonDown("Dialogue")){

```

```

        PlayDialogueSound(currentCharacter);
    }

    if(PlayerScored()){
        Audio.play(scoreSound);
    }

```

#### 4. Acceptance Tests \_\_\_\_\_9

##### Button Presses:

The buttons placed throughout the game sound make sounds when pressed. This will include all the buttons on the main menu, the HUD, and the buttons to advance the dialogue. Each different type of button should produce unique sounds to differentiate between them.

##### Background Music:

This won't really require any inputs besides navigating to the different areas of the game, like the main menu, character dialogue scenes, and mini games. Each area of the game should have their own background music playing.

##### Score a Point in a Minigame:

The inputs will be varied, but for one game we will have pong, so that case would be moving the paddle up and down to score against an enemy AI. The output would be a positive jingle so the user will feel more satisfied scoring a point.

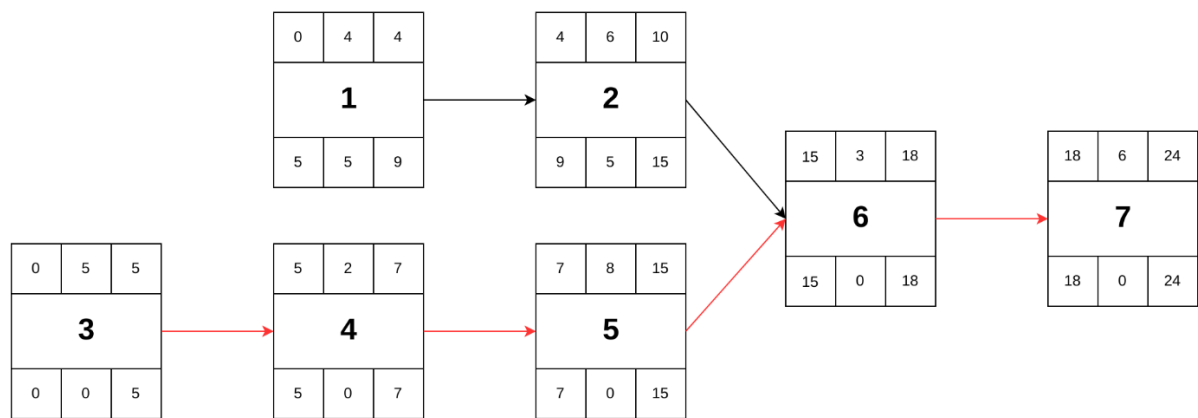
#### 5. Timeline \_\_\_\_\_/10

##### Work items

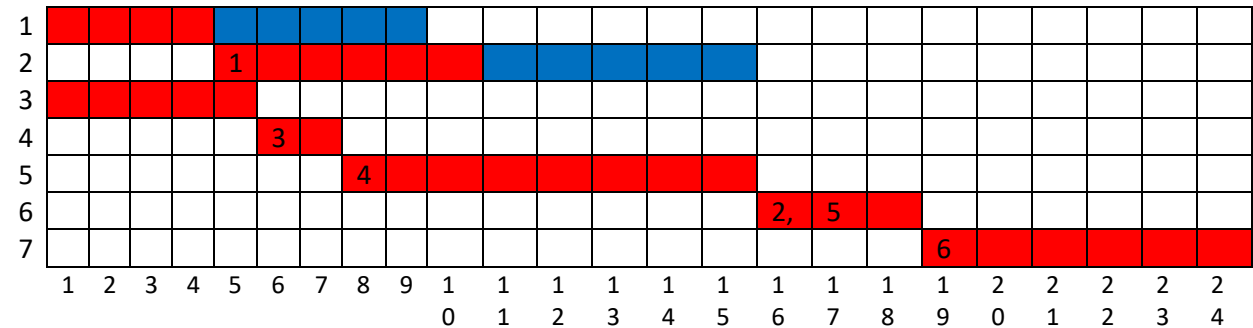
Task	Duration (Hours)	Predecessor Task(s)
1. Character Scoring Code and Functionality	4	-
2. Character Scoring Customization and Testing	6	1
3. Sound and Music Creation	5	-
4. Sound and Music Exportation	2	3
5. Sound and Music Code Implementation	8	4

6. Integration	3	2, 5
7. Testing and Fixes	6	6

Pert diagram



Gantt timeline



Key:

Work Hours	Slack