

1. Brief introduction _/3

My features encompass all sounds and music within the game. If there is any audio output sent to the user, it was my feature that generated it. Both sound and music are essential elements in a successful game that is entertaining to the user. They are not required, but do act as supporting elements that make the user's experience more realistic; therefore, it's recommended some implementation of sound and/or music occurs. Sounds are useful for letting the user know when some sort of action has taken place. Some examples of when sounds are beneficial are button selecting, moving objects within the game, and noises to draw the user's attention. Music is useful for controlling the user's emotions and thoughts in a game. Using different tones and paces for different scenes will allow for better user engagement and interest.

I will also be implementing a player class. This class will be responsible for defining player objects in our game. These objects are controlled by the user and have several traits like inventory, health, and damage.

2. Use case diagram with scenario _14

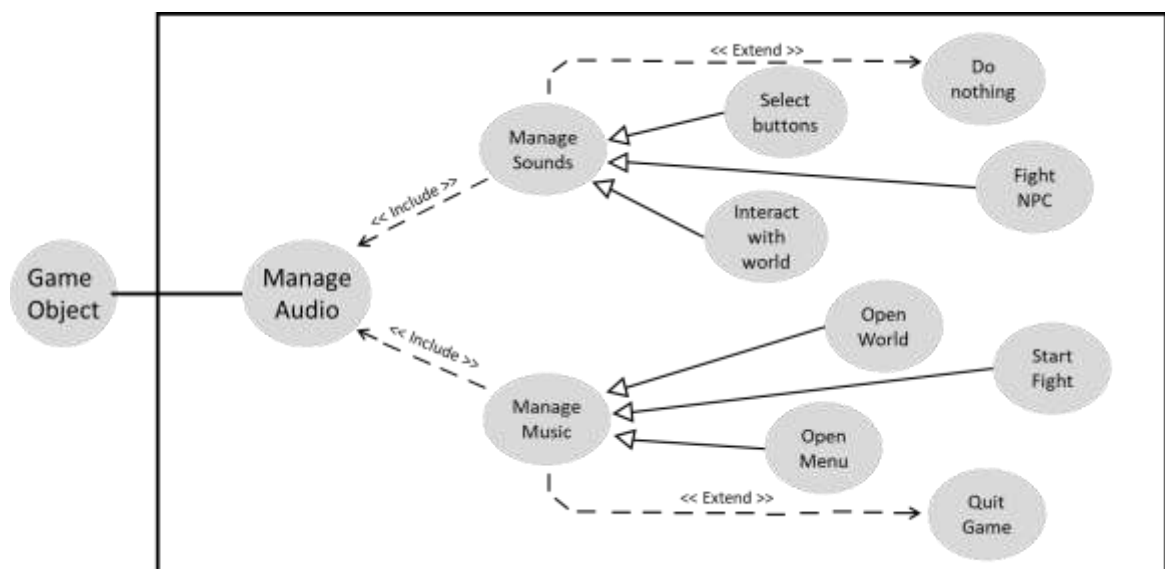
[Use the lecture notes in class.

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: Manage Sounds

Summary: The game object uses the machine to output sound.

Actors: Game Object

Preconditions: Game Object has a sound component

Basic sequence:

Step 1: Game object performs action.

Step 2: Action triggers sound.

Step 3: Sound is outputted.

Exceptions:

Step 1: Game Object performs no action

Step 2: No sound is outputted

Post conditions: Game user hears sound

Priority: 3*

ID: S01

*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

Name: Manage Music

Summary: The game object uses the machine to output music.

Actors: Game Object

Preconditions: Game Object is instantiated

Basic sequence:

Step 1: Game object displays on screen

Step 2: Music is outputted

Exceptions:

Step 1: User quits the game

Step 2: Music turns off

Post conditions: User hears music

Priority: 3*

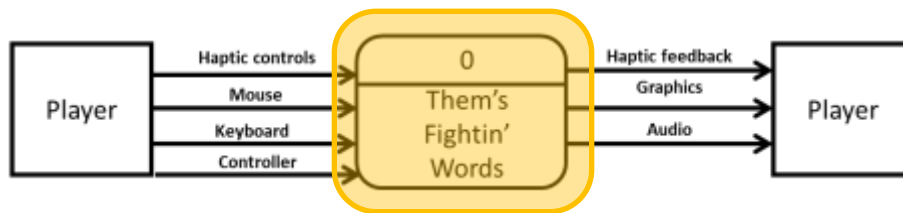
ID: M01

*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

3. Data Flow diagram(s) from Level 0 to process description for your feature ____14

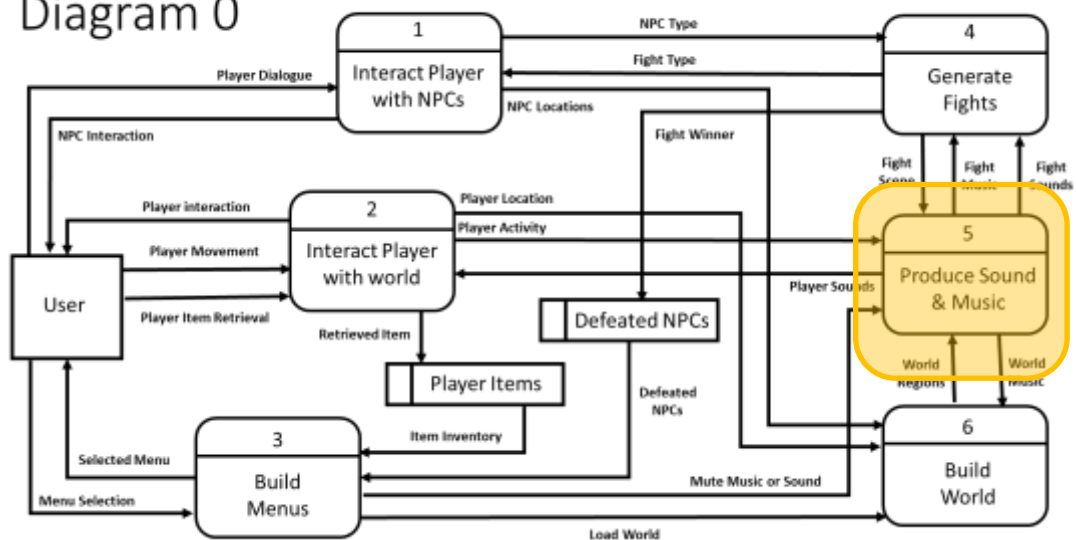
Data Flow Diagrams

Context Diagram



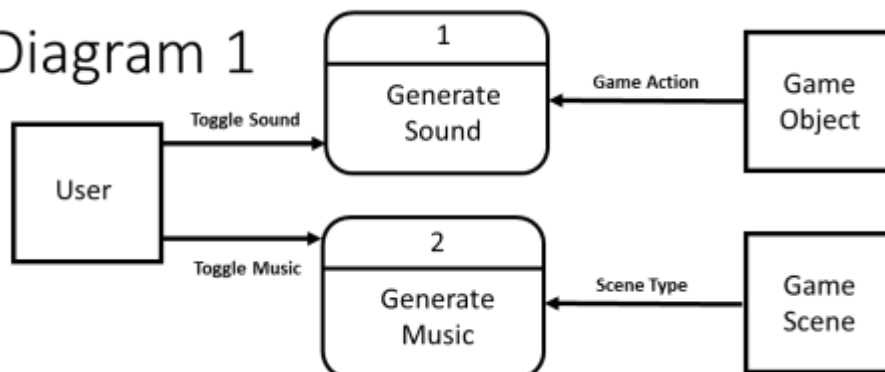
LEVEL 1:

Diagram 0



LEVEL 2:

Diagram 1



Process Descriptions

Generate Music:

IF Music Button is toggled OFF:

Mute all Music

ELSE:

IF Menu:

Play Menu Music

ELSE:

WHILE Game Scene is displayed:

Play Scene Music

Generate Sound:

IF Sound Button is toggled OFF:

Mute all Sound

ELSE:

IF Game Object Moves:

Play Movement Sound

ELSE IF Game Object Retrieves Item:

Play Accomplishment Sound

ELSE IF Game Object Attacks:

Play Attack Sound

ELSE IF Game Object Defends:

Play Defend Sound

ELSE IF Game Object sends Dialogue:

Play Dialogue Sound

ELSE:

Do Nothing

4. Acceptance Tests _____9

[Describe the inputs and outputs of the tests you will run. Ensure you cover all the boundary cases.]

Test Mute Feature (Sound and Music):

Determine if sound or music is outputted when various forms of input are generated

Input: player movement, attack, defense, button selection, menu selection, scenes

Outputs: True if sound/music is outputted

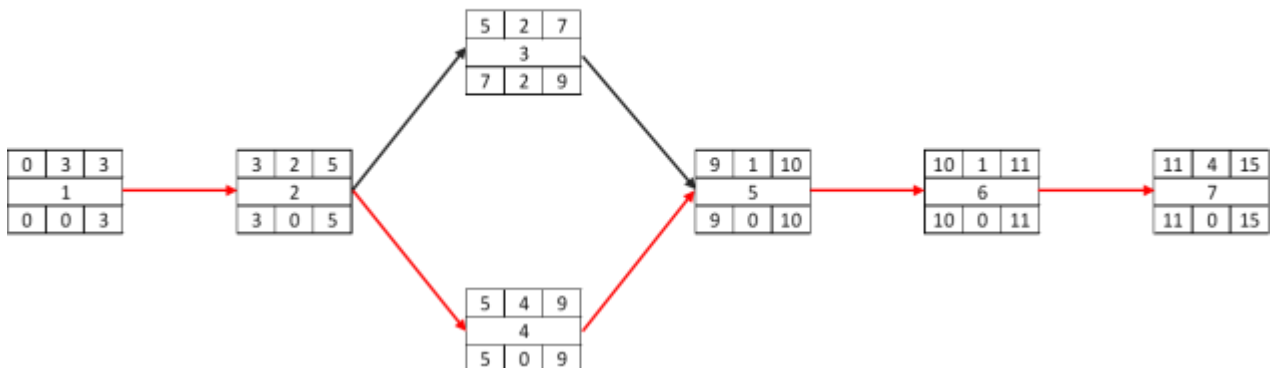
False if no sound or music is present

5. Timeline _____/10

Work items

Task	Duration (PWks)	Predecessor Task(s)
1. Feature Requirements	3	-
2. Feature Modeling	2	1
3. Develop Test Plan	2	2
4. Initial Programming	4	2
5. Programming Modifications	1	4
6. User Documentation	1	5
7. Game Presentation	4	6

Pert diagram



Gantt timeline

