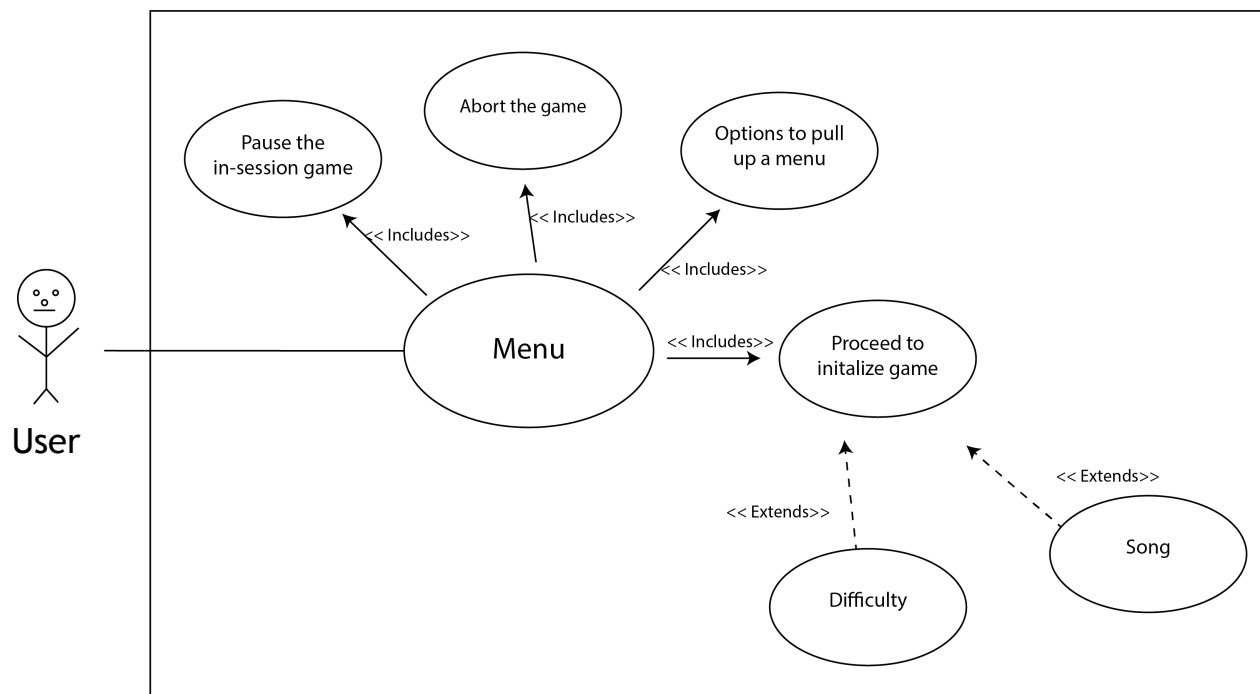


1. Brief introduction __/3

Low Effort Games will be bring a single-player, musical-rhythm game to your PC's. "Beat runner" will keep you engaged as you fight your way through every obstacle as the song progresses. Also, the game will include the top 3 greatest hits from the 80's! Now as for my role in the game, I will be in charge of creating an appealing "start menu" with a simple user-friendly interface that will include a "Start", "Options", and "Quit" panel. Now as far as the game, I will bring the ability to "Pause" and "Quit" the game incase you need to get some homework done.

2. Use case diagram with scenario __14



Scenarios

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

Name: Play

Summary: The user uses the [Start] to proceed to the game.

Actors: User

Preconditions: Game has been initialized.

Basic sequence:

Step 1: User must initialize the game.

Step 2: User will have to click on a button.

Step 3: Button will execute the game command.

Step 4: Begin to play!

Exceptions:

Step 1: User chooses [Song]

Step 2: User Chooses [Difficulty]

Post conditions: Game must be initialized.

Priority: 1

ID: A01

Name: Pause

Summary: The user uses the [Pause] to halt an in-session game.

Actors: User

Preconditions: Game has been initialized.

Basic sequence:

Step 1: User must click on [Pause].

Step 2: Will execute the pause command.

Step 3: Open a Panel with [Resume].

Exceptions:

None

Post conditions: Game must be initialized.

Priority: 1

ID: A02

Name: Options

Summary: The user uses the [Options] to adjust volume of the game.

Actors: User

Preconditions: Game has been initialized.

Basic sequence:

Step 1: User must click on [Options] at the beginning of game

Step 2: Will execute the options command.

Step 3: Open panel with toolbar

Step 4: Toolbar will match level of volume.

Exceptions:

None

Post conditions: Game must be initialized.

Priority: 2

ID: A03

Name: Choose Song

Summary: The user uses the [Song Selection] to control song.

Actors: User

Preconditions: Game has been initialized. [Play] has been pressed.

Basic sequence:

Step 1: Press [Play].

Step 2: Execute the [Song Selection].

Step 3: Open song selection panel.

Step 4: Press on [Song]

Step 5: Load song.

Exceptions:

Step 1: [Song Selection] is not pressed, it will result to the default song

Post conditions: Game must be initialized.

Priority: 3

ID: A04

Name: Choose Difficulty.

Summary: The user uses the menu to proceed to the game.

Actors: User

Preconditions: Game has been initialized.

Basic sequence:

Step 1: Press [Play].

Step 2: Go through song selection.

Step 3: Execute the [Choose Difficulty].

Step 4: Open the difficulty panel.

Step 5: Load the difficulty of game

Exceptions:

Step 1: [Difficulty] is not pressed, it will result to the default mode (Easy)

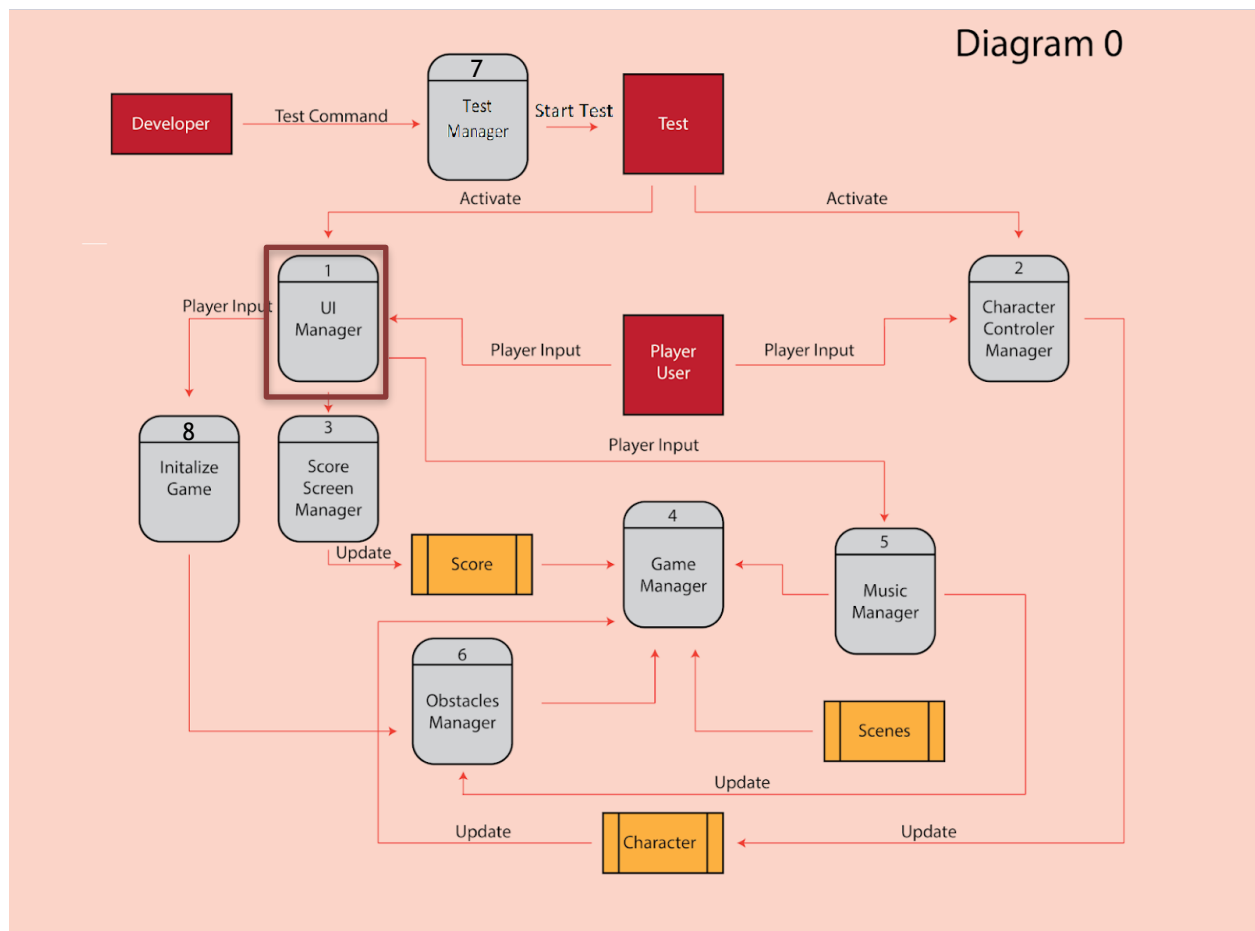
Post conditions: Game must be initialized.

Priority: 3

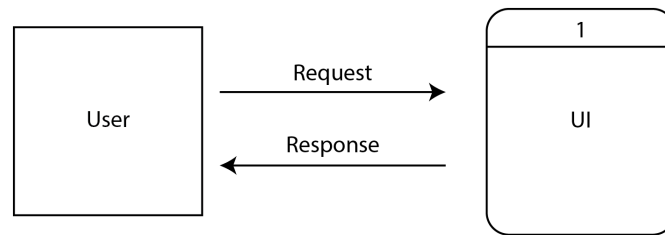
ID: A05

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

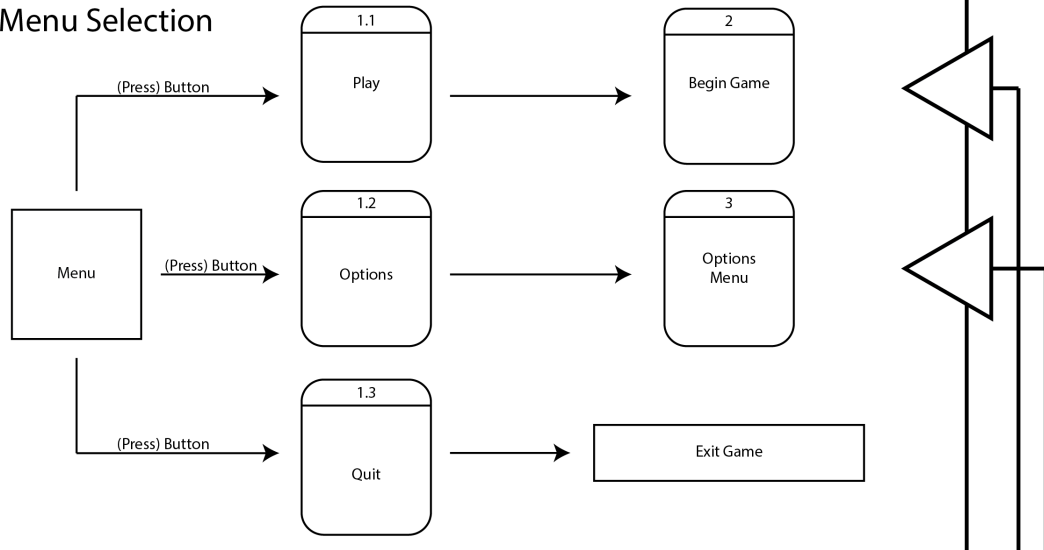
14



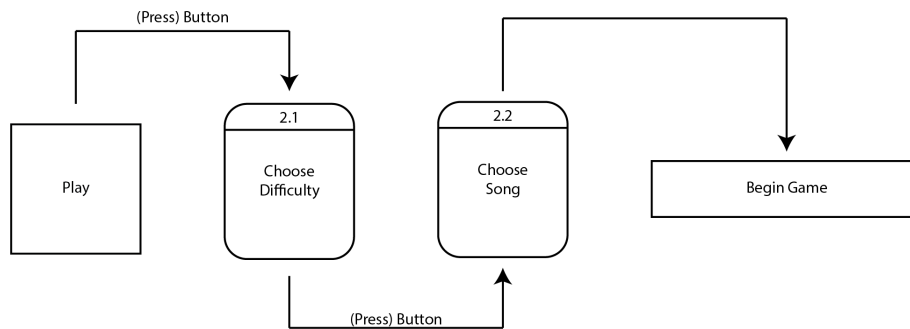
User Interface Manager



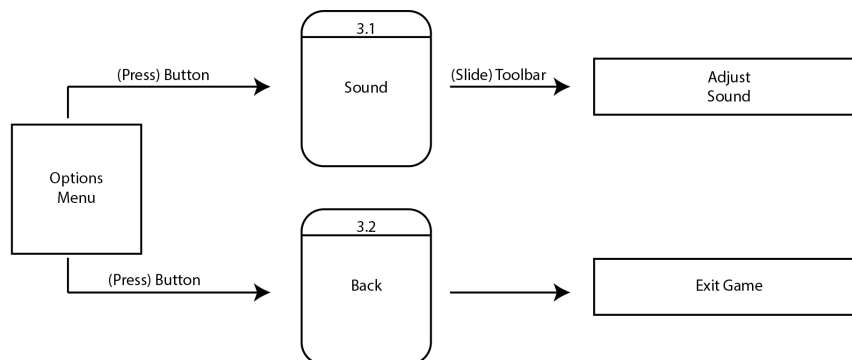
Menu Selection



Begin Game



Option Menu



Process Description:

1.1 Play (Parameter: Player Input){

 IF (Player Input == 1)

 Execute play();

 Else IF

 Do nothing //Remain on the main page

}

1.2 Options (Parameter: Player Input){

 IF (Player Input == 2)

 Execute options();

 Else IF

 Do nothing //Remain on the main page

}

1.3 Quit (Parameter: Player Input){

 IF (Player Input == 3)

 Execute quit();

 Else IF

 Do nothing //Remain on the main page

}

2.1 Choose Difficulty (Parameter: Player Input){

 IF (Player Input == 3)

 Execute quit();

 Else IF

 Do nothing //Remain on the main page

}

2.2 Choose Song (Parameter: Player Input){

IF (Player Input == Button 1 || Player Input == Button 2 || Player Input == Button 1)

 Load Song();

Else IF

 Do nothing //Remain on the main page

}

3.1 Sound (Parameter: Player Input){

IF (Player Input moves Toolbar)

 Execute sound();

Else IF

 Do nothing //Remain on the main page

}

3.2 Back (Parameter: Player Input){

IF (Player Input == 3)

 Execute Back();

Else IF

 Do nothing //Remain on the main page

}

4. Acceptance Tests _____9

Inputs: Click on the buttons to perform action

- Play
 - Input: Click on [Play]
 - Output: Game will be initialized.
- Options
 - Input: Click on [Options]
 - Output: Display Menu for user to access.
- Quit
 - Input: Click on [Quit]
 - Output: Aborts the game.
- Pause
 - Input: Click on [Pause]
 - Output: Halts the in-session game.
- Resume
 - Input: Click on [Resume]
 - Output: Resume the game activity.
- Song Selection (Extensions)
 - Input: Click on [Song]
 - Output: Experience the game with the song selected.
- Difficulty Selection (Extensions)
 - Input: Click on [Difficulty]
 - Output: Experience the game at the difficulty specified.

Output: Carry out action.

Example for [Option] button:

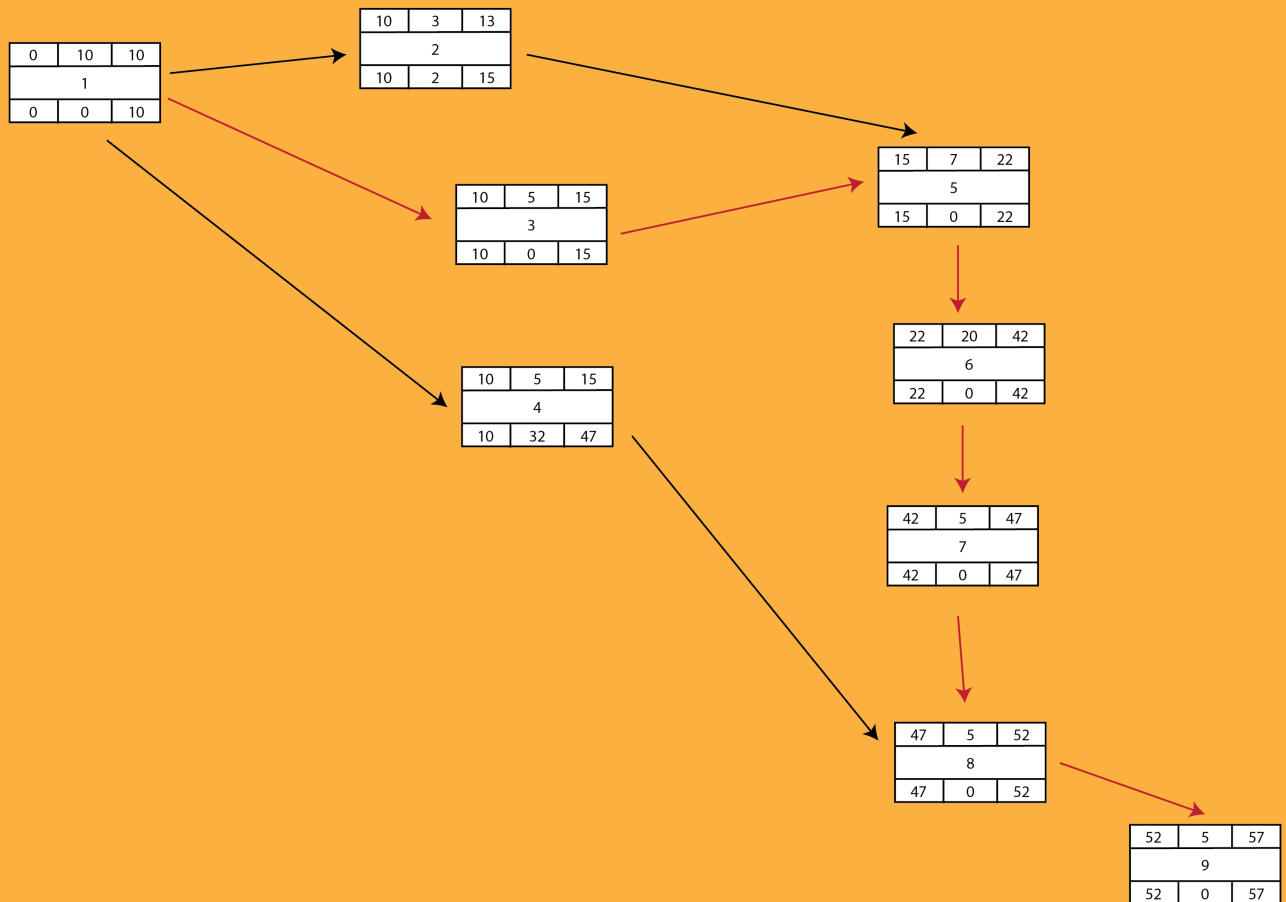
Option button can be connected to a hot-key. For example, if the user presses the 'o' on the keyboard then it will bring up the "Option Menu"

5. Timeline _____/10

Work items

Task	Duration (PWks)	Predecessor Task(s)
1. Requirements Collection	10	-
2. Menu1	3	1
3. SongList	5	1
4. Character List	5	1
5. In-game Menu 2	7	2,3
6. User Documentation	20	5
7. Programming	5	6
8. Testing	5	4
9. Installation	5	8

Pert diagram



Gantt timeline

