Software Design Document Echo

1. Introduction

This document is organized by first giving an overview of Echo, the audio game, and what it accomplishes followed by descriptions of its views via application flowcharts and mockups in Section 2, followed by a description of its backend via process flowcharts and UML diagrams in Section 3.

1.1 Purpose and Scope

Echo is a based audio game that is intended for those who have visual impairments. A more detailed description is available in the *Echo (Audio Game) Vision and Scope Document* [3], along with a list of features intended for initial, full or partial implementation release.

1.2 References

1. Hasty, Tyler; Bolanos, Eduard; Pelton, Charles; Iseri, Timothy. Trello
2. Hasty, Tyler; Bolanos, Eduard; Pelton, Charles; Iseri, Timothy. Echo Software Requirements Specification
3. Hasty, Tyler; Bolanos, Eduard; Pelton, Charles; Iseri, Timothy. Echo (Audio Game) Vision and Scope Document

2. User Interfaces

2.1 Overall Application Flow

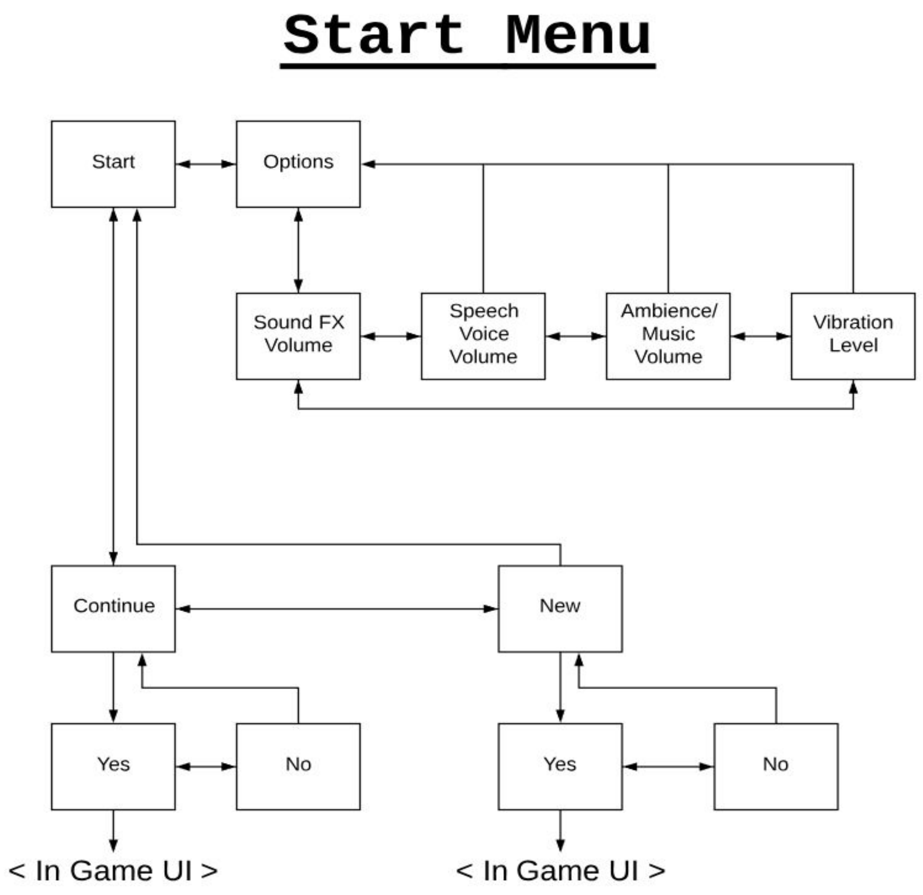


Figure 1: Start Menu Application Flow Diagram

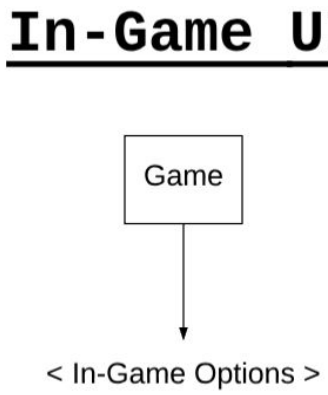


Figure 2: In-Game UI Application Flow Diagram

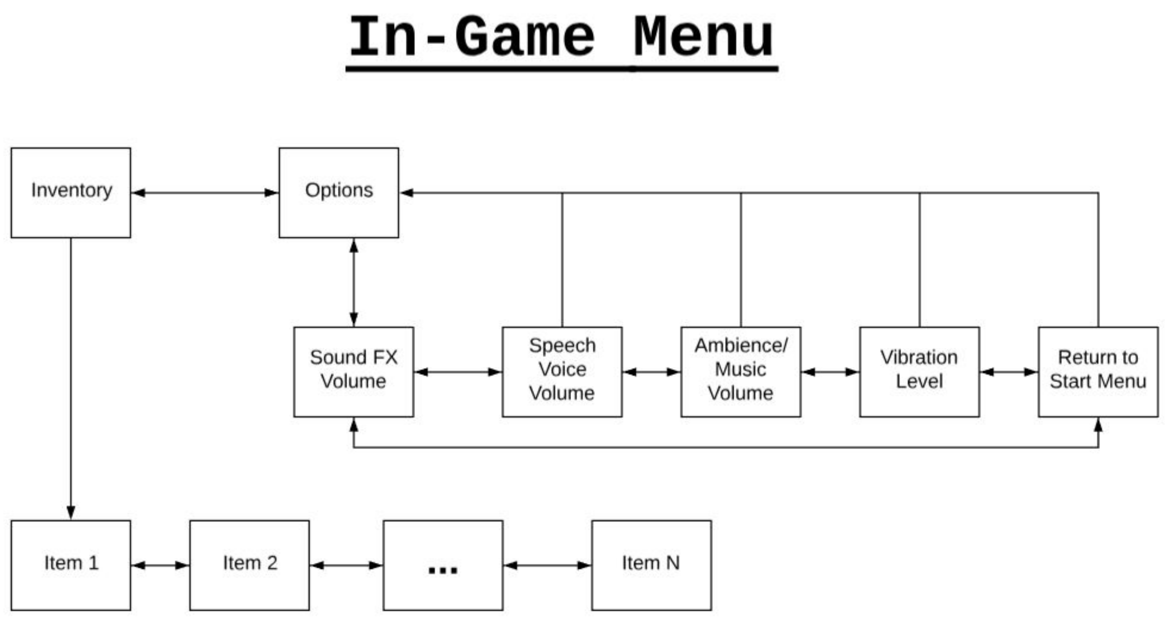


Figure 3: In-Game Menu Application Flow Diagram

2.2 Mockups Shared Characteristics:

In order to permit system interface (which contains back button, home button, etc) to exist comfortably at the bottom of the screen when the device and application are in portrait orientation, the input surface area for portrait orientation with consist of the upper ¾ of the screen area. For the same reason, the input surface area for landscape orientation will consist of the leftmost 90% of the screen area. All interfaces will be labeled with a bordered object at the center of the screen, labeled with the option name in size 30 Times New Roman font. A small logo indicating the audio interface with exist within the upper-center of this central border object. Double-directional-arrow icons will be featured on each side of this border object in order to indicate swipe input.

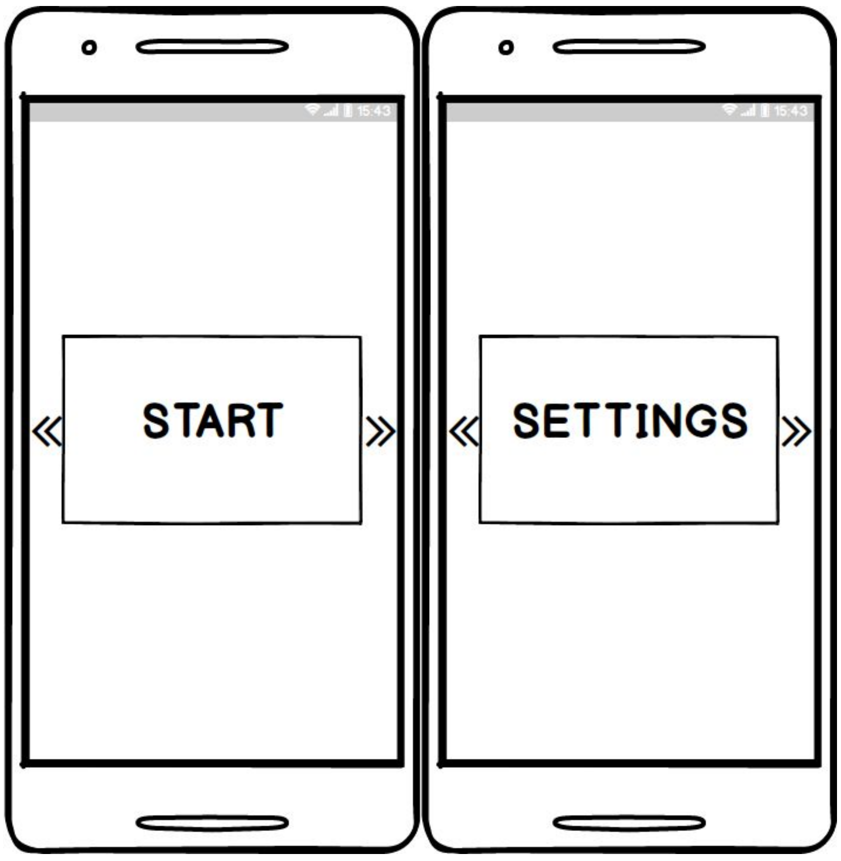
2.3.1 Start Menu

2.3.1.1 Purpose

- This view serves as the introductory/ initial application interface and affords access to all proceeding application interfaces

- The user can continue an existing save game or create a new save game. The user can also manipulate audio volume and vibration magnitude features in the settings sub-menu.
- The user's desire is to play the presented game, this menu is the interface that connects the user's desire to the product.
- The user can reach the main menu by starting the application on their device.
- The user can continue into the main game interface(2.3.2).
- The user swipes left and right to go between Start and Settings buttons and tap to confirm.
- The user when entering this menu hears the introduction of "Welcome to Echo. Swipe left and right to change between options, tap to confirm. You are at:" as it will play an audio clip and the audio clip "Start". The game will play audio clips of "Start" or "Settings" based on 'Visual' feedback.

2.3.1.2 Wireframe Screenshot



2.3.1.3 Mockup Screenshot

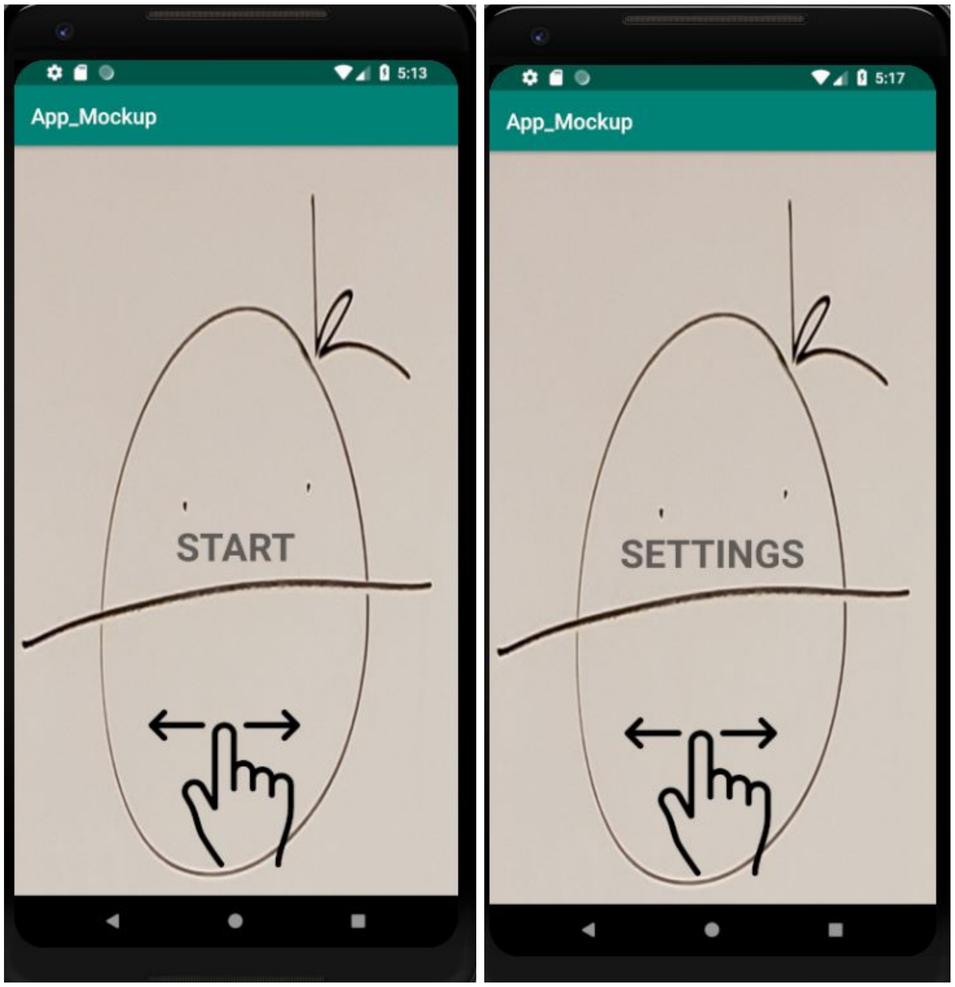
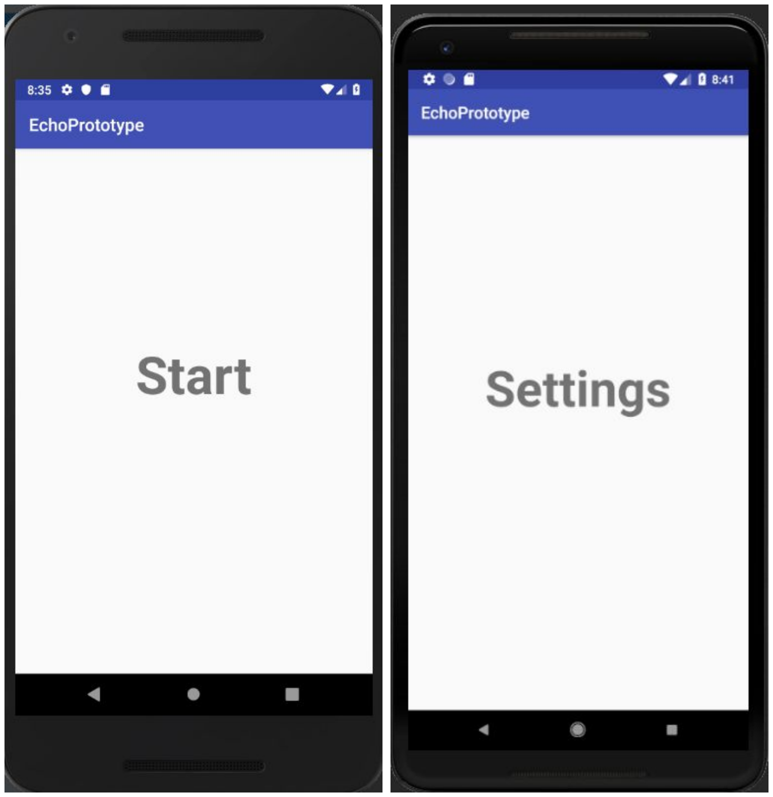


Figure Start Menu: The hand gestured represents swiping between START and SETTINGS

Sounds:

- 2.3.1.A.1: Intro
- 2.3.1.A.2: Start
- 2.3.1.A.3: Settings

2.3.1.4 Prototype Screenshot



All Sound Files for the Prototype: raw.zip

2.3.1.5 Design Commentary

Originally tapping on the sides of the screen would move to the next screen in that direction, but now swiping in a direction changes screens and tapping selects that option.

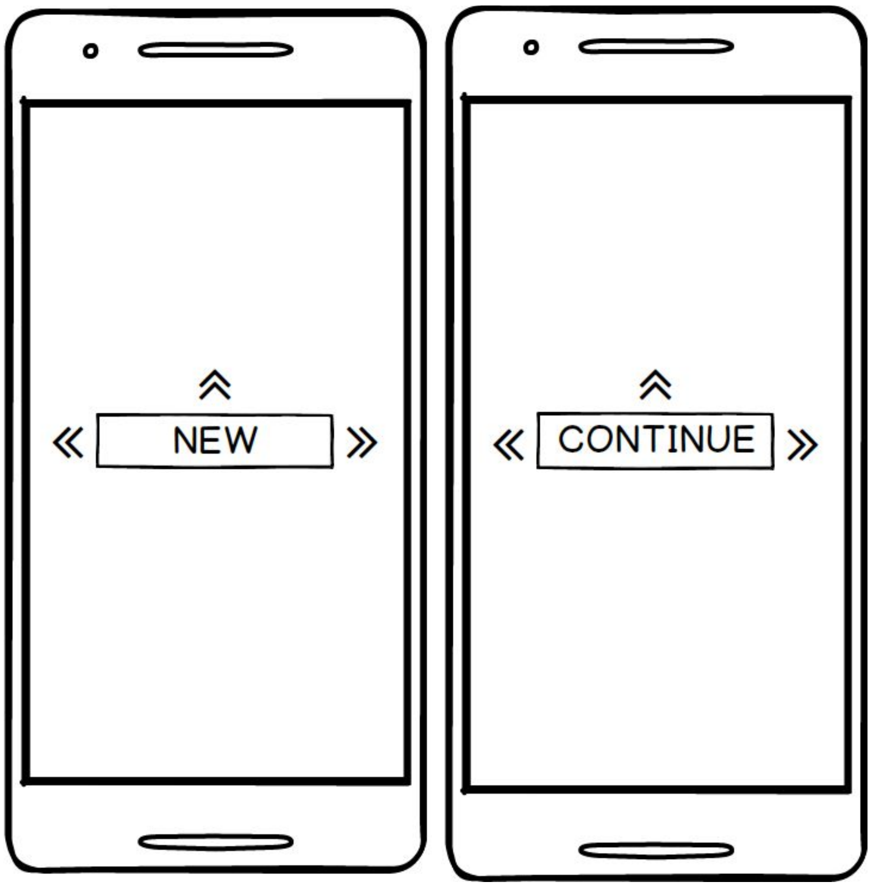
2.3.2 Game Initialization

2.3.2.1 Purpose

- The user can start a new game or continue an existing game using this interface.
- This view allows the user to select which action is needed to be initialized before going into the game interface.
- The user comes here from the Start Menu(2.3.1).
- The user can return to the Start Menu(2.3.1) or go onto confirmation interface of their choice.
- The user swipes left or right to go between New and Continue, swipes up to return to the Start Menu(2.3.1) and taps to confirm.

- The user when entering this menu hears the introduction of “Select: Start” as it will play an audio clip. The game will play audio clips of “Continue” or “New” based on ‘Visual’ feedback.

2.3.2.2 Wireframe Screenshot



2.3.2.3 Mockup Screenshot

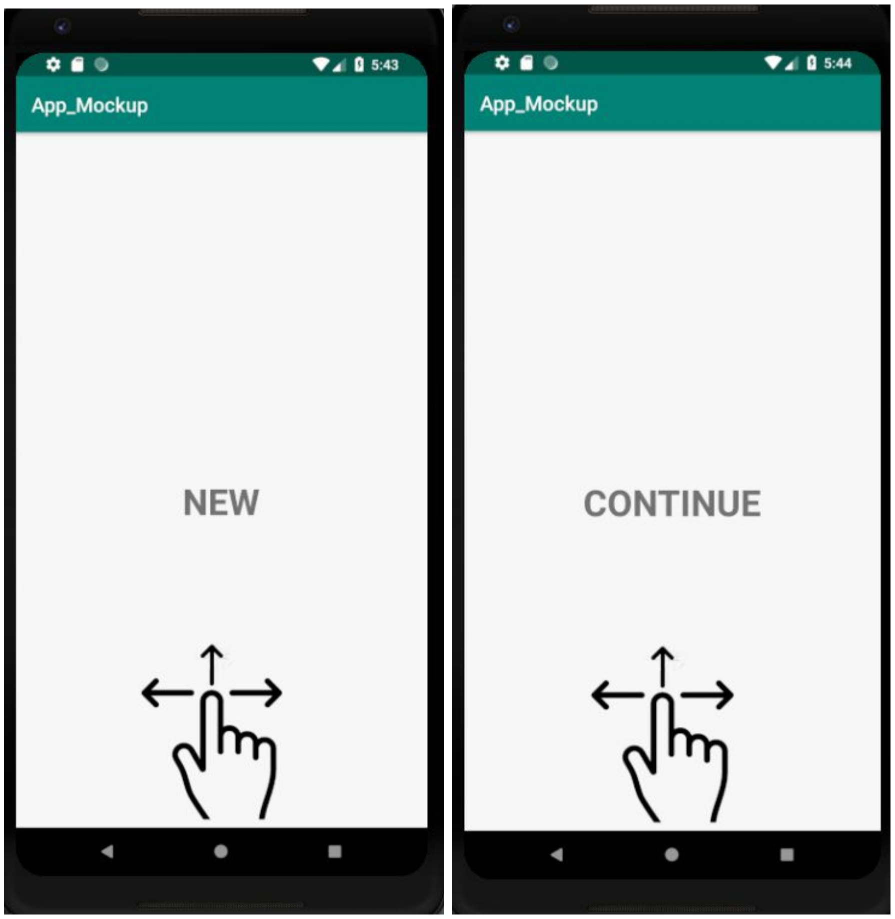
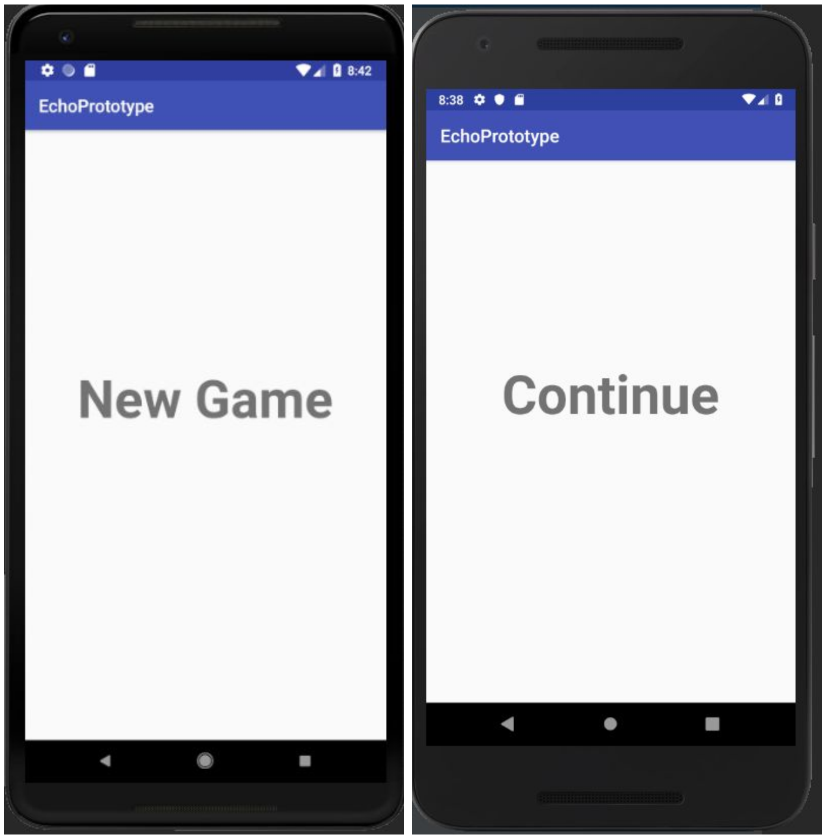


Figure Game Initialization: The hand gestured represents swiping between NEW and CONTINUE, the up arrow, when swiped, will return you back to the main menu

Sounds:

- 2.3.2.A.1: New
- 2.3.2.A.2: Continue

2.3.2.4 Prototype Screenshot



By 2.3.1.4: All Sound Files for the Prototype: raw.zip

2.3.2.5 Design Commentary

Originally tapping on the sides of the screen would move to the next screen in that direction, but now swiping in a direction changes screens and tapping selects that option.

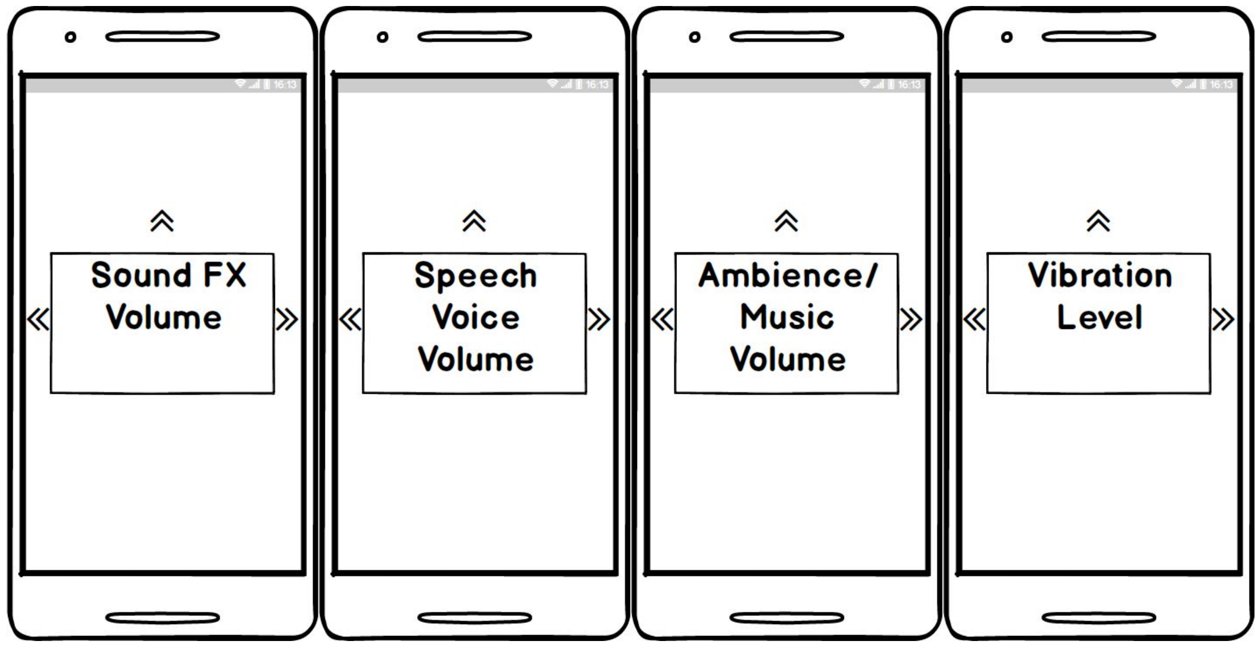
2.3.3 Settings

2.3.3.1 Purpose

- The purpose of this menu is to adjust settings to suit the individual level of tolerance and comfort to audio levels.
- The user can change sound settings, change dialog
- The user gets to the settings via Settings in the Start Menu(2.3.1).
- The user can go back to the Start Menu(2.3.1) or configure their respective setting of choice.

- The user can traverse the options of Sound FX Volume, Speech Voice Volume, Ambience/Music Volume and Vibration level by swiping left and right. The user can go back to the Start Menu(2.3.1) by swiping up and the user can confirm by tapping.
- The user when entering this menu hears the introduction of “Select: Settings” as it will play an audio clip. The game will play audio clips of “Sound FX Volume”, “Speech Voice Volume”, “Ambience/Music Volume” and “Vibration Level” based on ‘Visual’ feedback.

2.3.3.2 Wireframe Screenshot



2.3.3.3 Mockup Screenshot

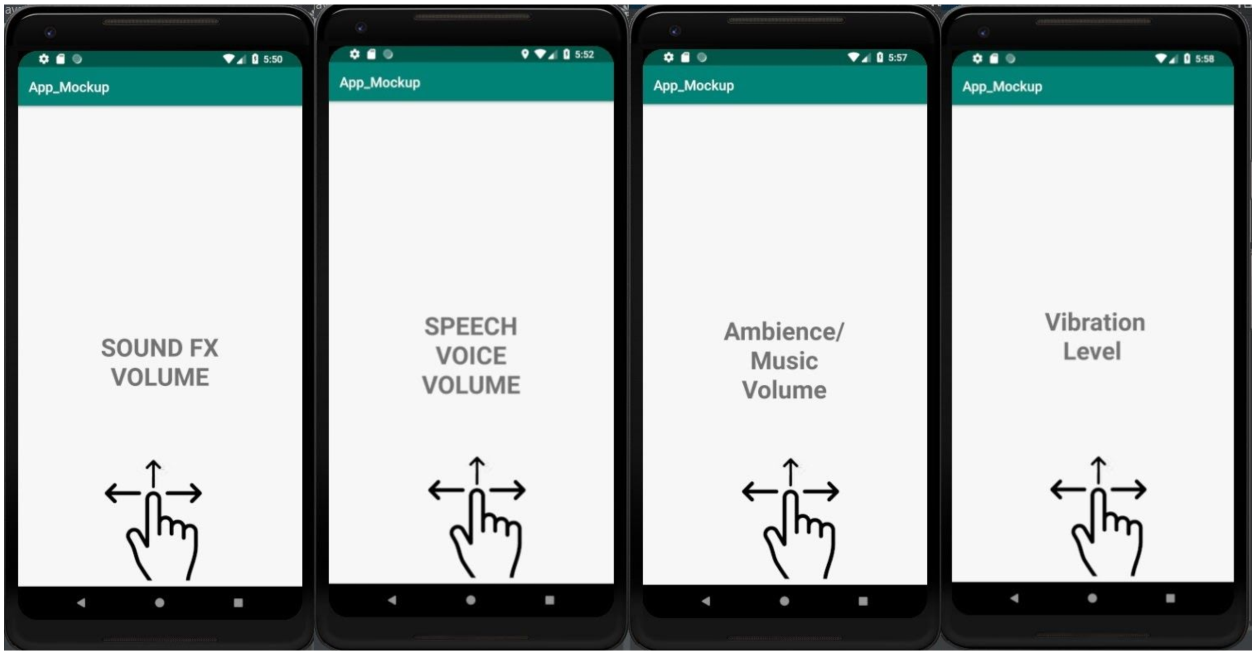
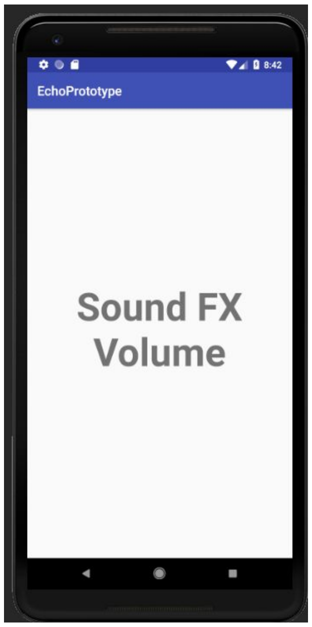


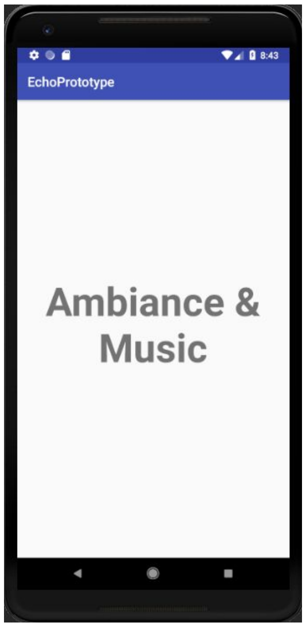
Figure Settings: The hand gestured represents swiping between the settings of the game, the up arrow, when swiped, will take you back to the main menu

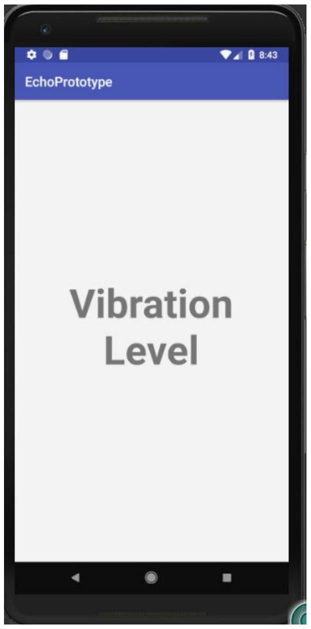
Sounds:

- 2.3.3.A.1: Sound FX
- 2.3.3.A.2: Speech Voice Volume
- 2.3.3.A.3: Ambience/Music Volume
- 2.3.3.A.4: Vibration Level

2.3.3.4 Prototype Screenshot







By 2.3.1.4: All Sound Files for the Prototype: [raw.zip]

2.3.3.5 Design Commentary

The UI barely changed between the wireframe, mockup and prototype. Some design features were removed, text was enlarged and the prototype was given new audio files.

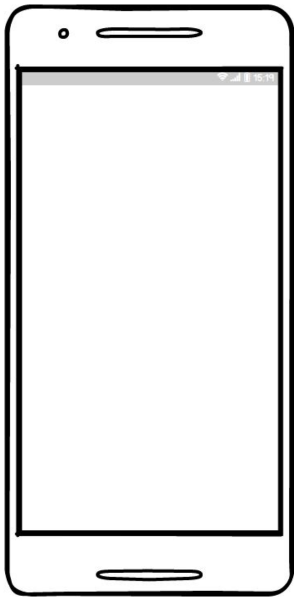
2.3.4 Game Interface

2.3.4.1 Purpose

- This view serves as the primary gameplay interface for players as well as the screen featuring developer diagnostics for testers/ developers
- The user can control the protagonist by providing screen inputs (tapping and swiping). This can navigate the program state to the ingame menu, move the character forward by swiping up, turn by swiping left or right, echolocate by tapping while not touching an object , and interact with objects by tapping.
- This interface allows the user to play the game by engaging with primary game features and audio feedback.
- The user can get to this screen from the start new game or continue new game screens(2.3.1).
- The user can get to and back from the in-game options menu by swiping down which includes an inventory menu(2.3.2)
- If the user encounters an object that cannot be passed through (ie. wall, door,) a sound will play indicating that the path is obstructed, the sound will be different to identify the differing types of obstructions.

- The player can activate echolocate to play a sound that will identify objects in front of them and how far away they are.

2.3.4.2 Wireframe Screenshot



2.3.4.3 Mockup Screenshot

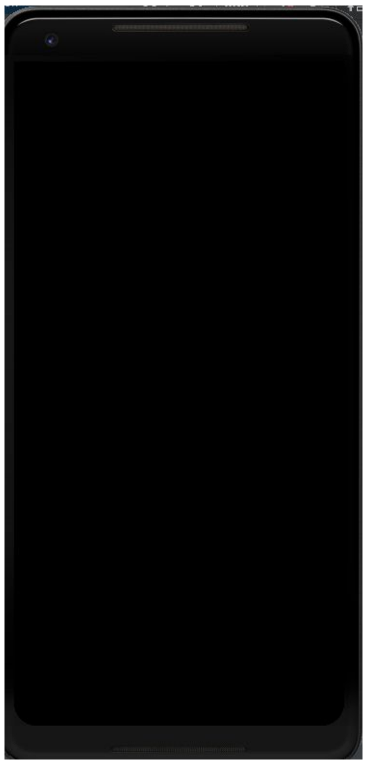


Figure Game Interface: It's an audio game, so there is no visuals for the actual game
Dialog Sample from characters:

2.3.4.A.1: Antagonist - The Supreme One
2.3.4.A.2: Beginnings - Alice
2.3.4.A.3: Cardinality - Igor
2.3.4.A.4: Defeat - Ricardo, Grand Explorer
2.3.4.A.5: Exploring - Ricardo, Grand Explorer

2.3.4.5 Prototype Screenshot

By 2.3.1.4: All Sound Files for the Prototype: raw.zip

2.3.4.6 Design Commentary

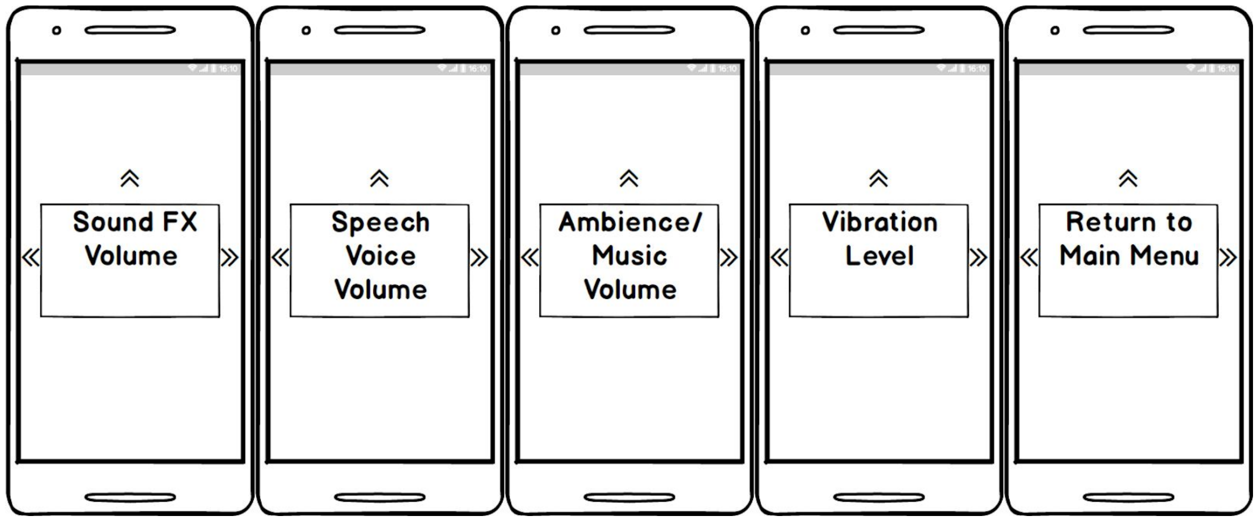
Original design was cluttered with symbols describing actions the player could take and information the the player won’t necessarily be able to read, current design is minimal with nothing on screen and an implementation of audio cues. The echolocate function in its current state is primitive.

2.3.5 In-Game-Settings

2.3.5.1 Purpose

- The user may access inventory(2.3.6), change game settings or return to Start Menu(2.3.1) interface.
- Thu user can make adjustments to such attributes as audio volume and vibration magnitude, or end the current game and return to the Start Menu(2.3.1)
- The in-game options allows to user to perform more complex interactions to continue playing the game as well as return to previous menus to tweak their experience and comfort.
- The user can get to this menu from the game-interface(2.3.2)
- The user can get back to the Game-Interface (2.3.2) and the main Start Menu (2.3.1).

2.3.5.2 Wireframe Screenshot



2.3.5.3 Mockup Screenshot

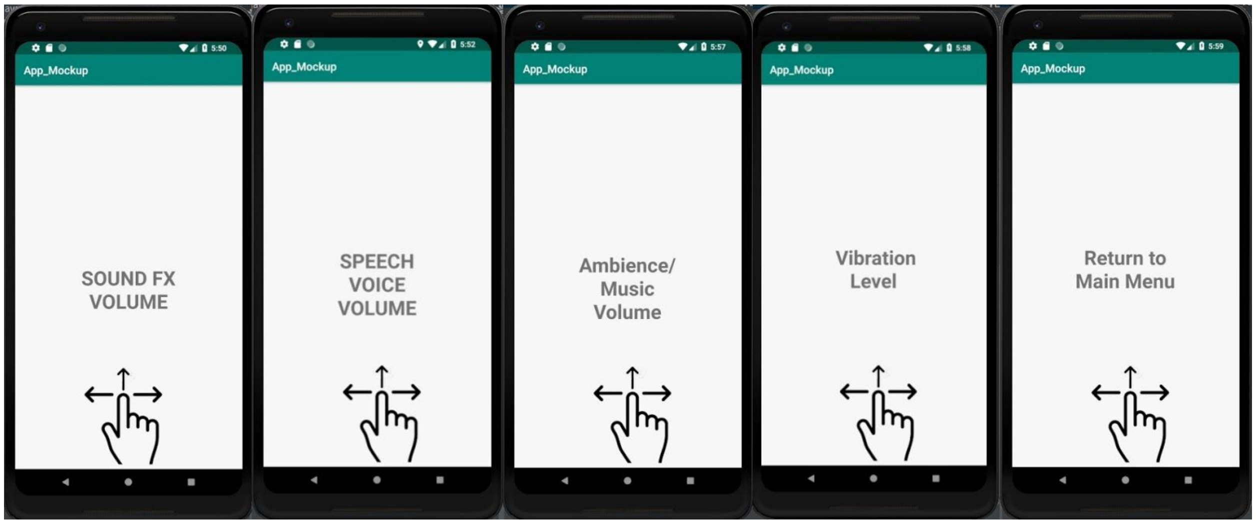


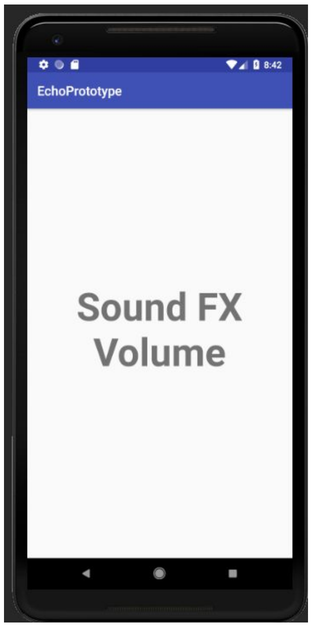
Figure In-Game Settings: The hand gestured represents swiping between the settings of the game, the up arrow, when swiped, will take you back to the in game menu

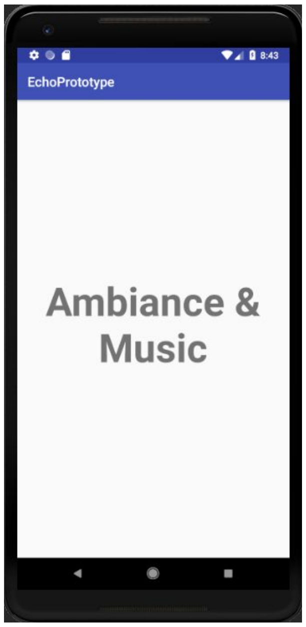
Sounds:

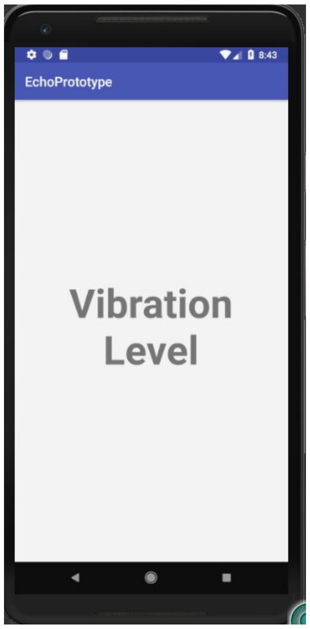
- 2.3.5.A.1: Sound FX
- 2.3.5.A.2: Speech Voice Volume
- 2.3.5.A.3: Ambience/Music Volume
- 2.3.5.A.4: Vibration Level
- 2.3.5.A.5: Go Back; Nick Version

2.3.5.4 Prototype Screenshot

Design for Implementation, Same Concept as 2.3.3.4; **Not in gameplay prototype version 1.0.0**







By 2.3.1.4: All Sound Files for the Prototype: [raw.zip]

2.3.5.5 Design Commentary

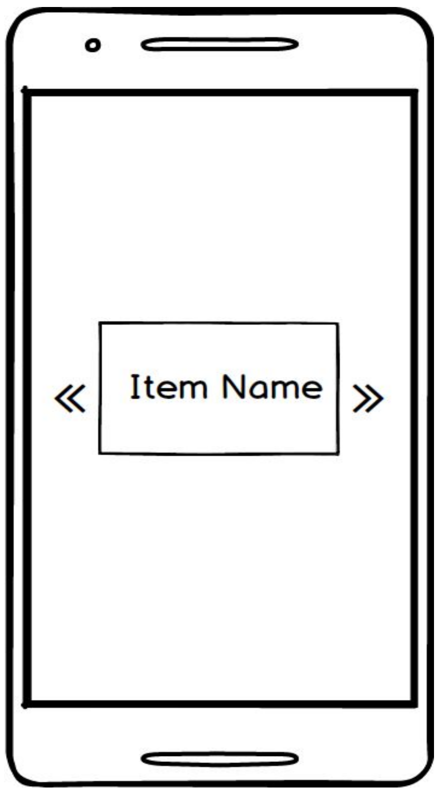
Originally tapping on the sides of the screen would move to the next screen in that direction, but now swiping in a direction changes screens and tapping selects that option.

2.3.6 Item Screen

2.3.6.1 Purpose

- The user will be able to scroll through all items in their possession.
- Each item will play an identifying sound signifying the specific type of item (ie. key, lever, etc.). Additionally a vibration pattern will be played to identify where and how the item will be used.
- The name of the item can be read aloud if further identification is needed by the user.
- The user can get to this screen from another item screen or the in-game interface(2.3.4) and can get back to (2.3.4) by swiping up.

2.3.6.2 Wireframe Screenshot



2.3.6.3 Mockup Screenshot

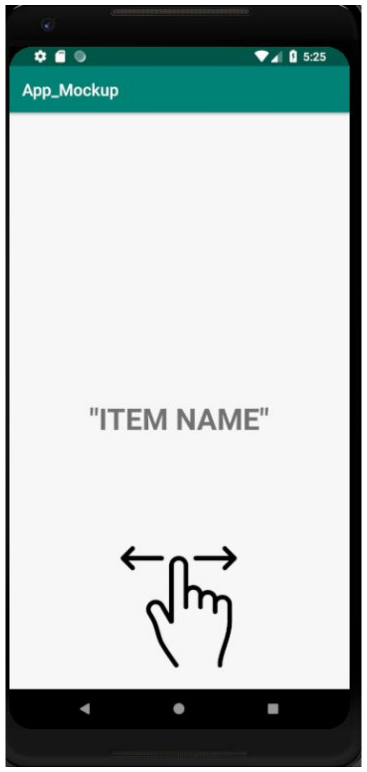


Figure Item Screen: The hand gestured represents swiping between items

Example of Traversing through the item screen

Sounds:

- 2.3.6.A.1: Items

2.3.6.4 Prototype Screenshot

Not in Gameplay Prototype Version 1.0.0

2.3.6.5 Design Commentary

Originally tapping on the sides of the screen would move to the next screen in that direction, but now swiping in a direction changes screens and tapping selects that option.

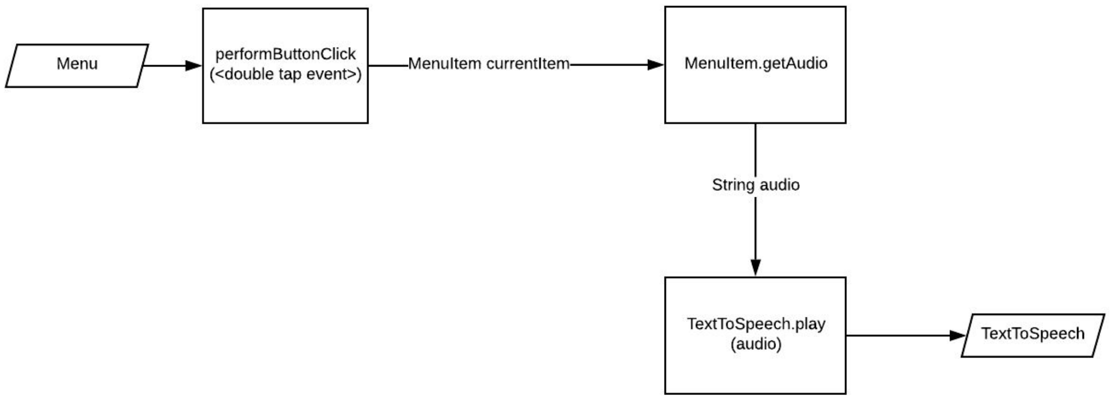
3. Software Interfaces

3.1 Text to Speech

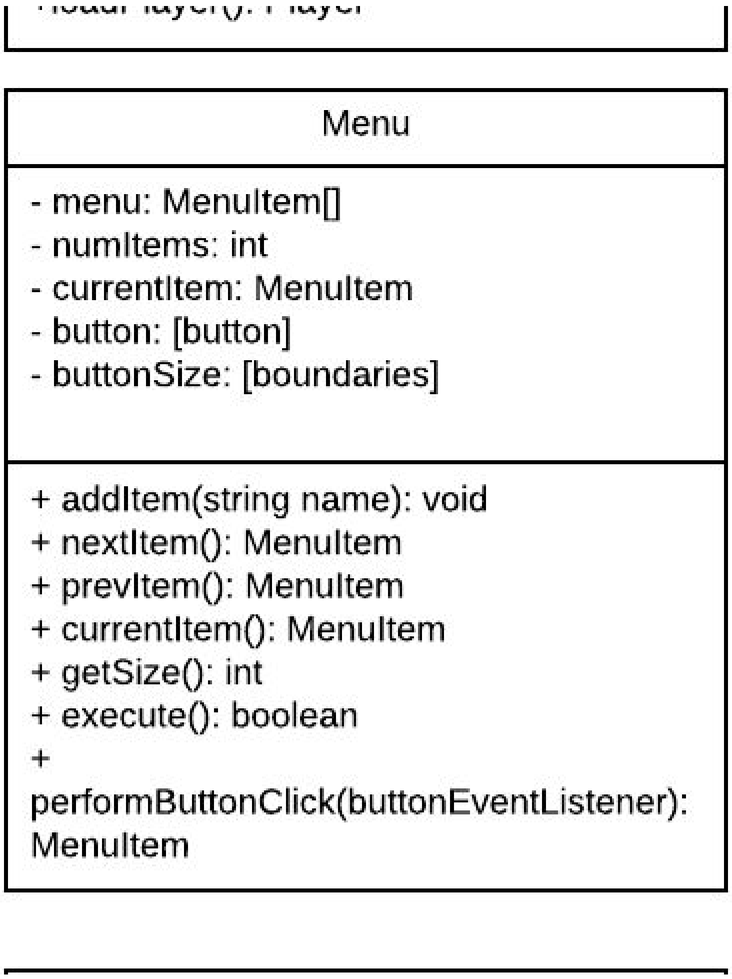
3.1.1 Process Description

This process occurs when the user provides a double-tap screen input on a menu item of a menu. The “button” member of the “Menu” object registers a double-tap event and returns a MenuItem object representing the currently selected menu item (the currentItem attribute), calling its getAudio() method which in turn passes the audio corresponding to the name of the cuurent menu item to the TextToSpeech object’s play() method, which uses text-to-speech processing to read the menu item name aloud to the user.

3.1.2 Process Flowchart



3.1.3.1 Menu

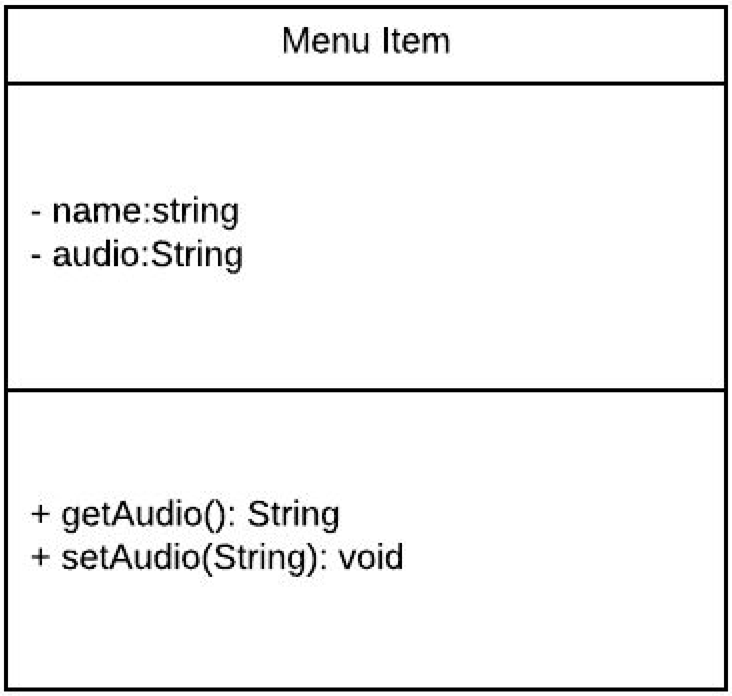


Menu

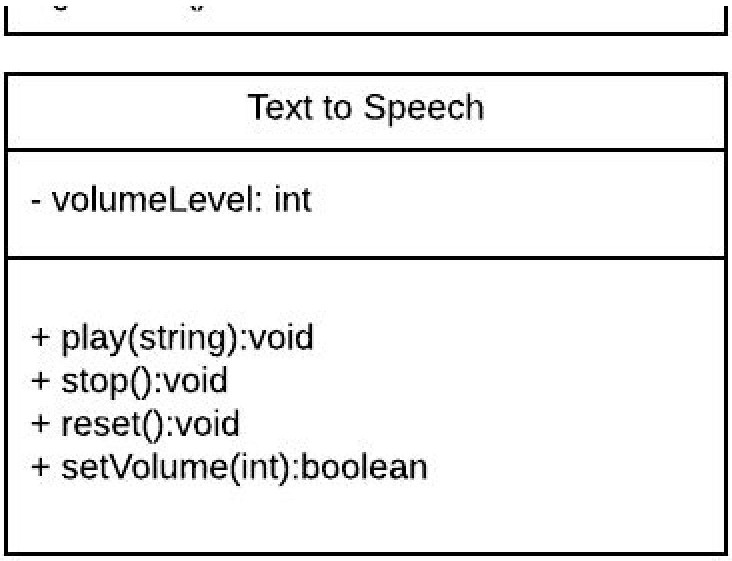
- menu: MenuItem[]
- numItems: int
- currentItem: MenuItem
- button: [button]
- buttonSize: [boundaries]

+ addItem(string name): void
+ nextItem(): MenuItem
+ prevItem(): MenuItem
+ currentItem(): MenuItem
+ getSize(): int
+ execute(): boolean
+ performButtonClick(buttonEventListener): MenuItem

3.1.3.2 Menu Item



3.1.3.3 Text to Speech



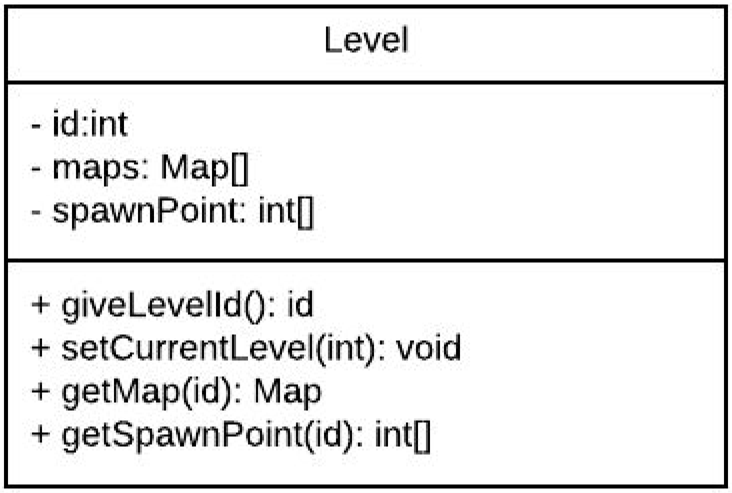
3.2 Player Spawn

3.2.1 Process Description

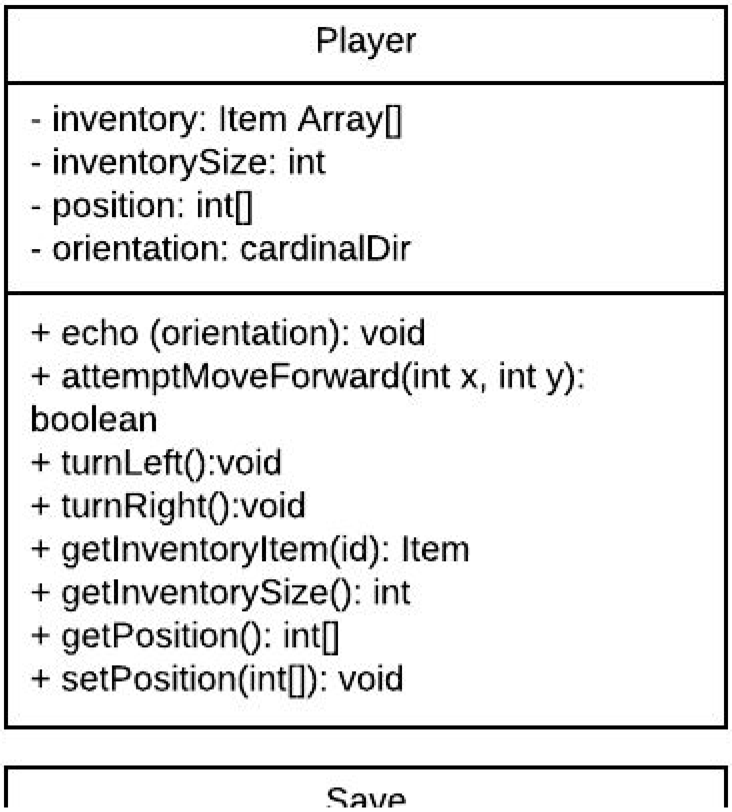
This process occurs at the beginning of each level, upon initialization of the level.

3.2.2 Process Flowchart

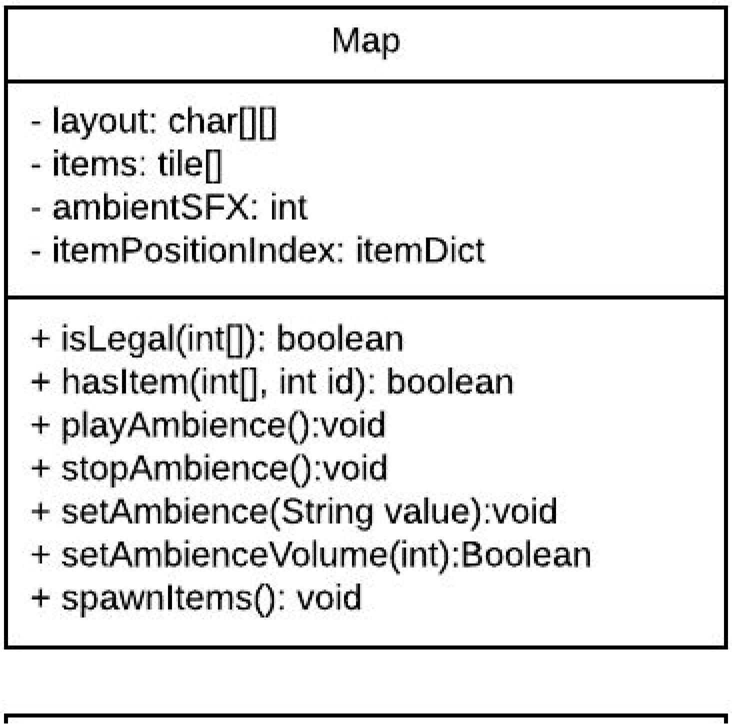
3.2.3.1 Level


Level
- id:int
- maps: Map[]
- spawnPoint: int[]
+ giveLevelId(): id
+ setCurrentLevel(int): void
+ getMap(id): Map
+ getSpawnPoint(id): int[]

3.2.3.2 Player


Player
- inventory: Item Array[]
- inventorySize: int
- position: int[]
- orientation: cardinalDir
+ echo (orientation): void
+ attemptMoveForward(int x, int y): boolean
+ turnLeft():void
+ turnRight():void
+ getInventoryItem(id): Item
+ getInventorySize(): int
+ getPosition(): int[]
+ setPosition(int[]): void
Save

3.2.3.3 Map

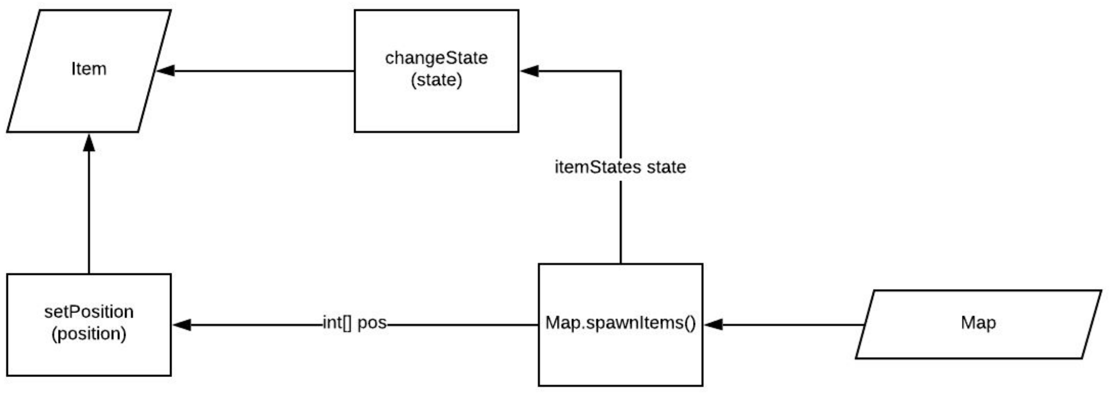

Map
- layout: char[][]
- items: tile[]
- ambientSFX: int
- itemPositionIndex: itemDict
+ isLegal(int[]): boolean
+ hasItem(int[], int id): boolean
+ playAmbience():void
+ stopAmbience():void
+ setAmbience(String value):void
+ setAmbienceVolume(int):Boolean
+ spawnItems(): void

3.3 Item Spawn

3.3.1 Process Description

This process occurs at the beginning of each level, upon initialization of the level.

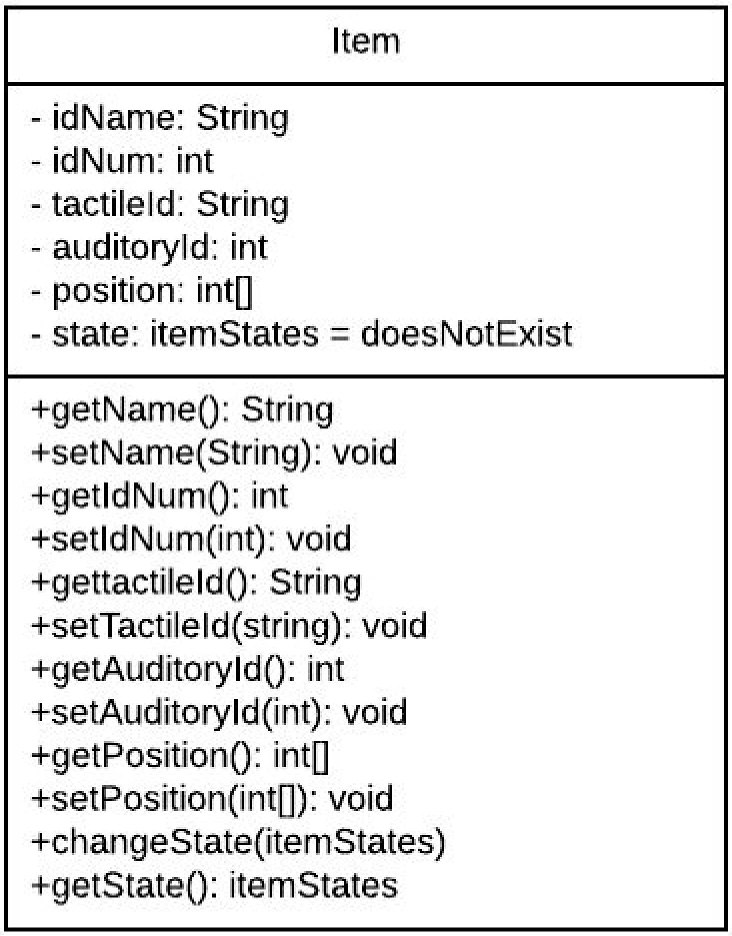
3.3.2 Process Flowchart



3.3.3.1 Map

[See 3.3.2.3]

3.3.3.2 Item



3.4 Interact With Interactive Tile

3.4.1 Process Description

This process occurs when the player provides a specific input when oriented towards an "interactive tile".

3.4.2 Process Flowchart

Screenshot of diagram for some application process goes here.

3.4.3.Y UML for object involved

UML diagrams for any involved objects in this process flow.
If the UML diagram was presented earlier in this list simply state the reference number (3.X.3.Y).

3.5 Player Movement

3.5.1 Process Description

3.5.2 Process Flowchart

Screenshot of diagram for some application process goes here.

3.5.3.Y UML for object involved

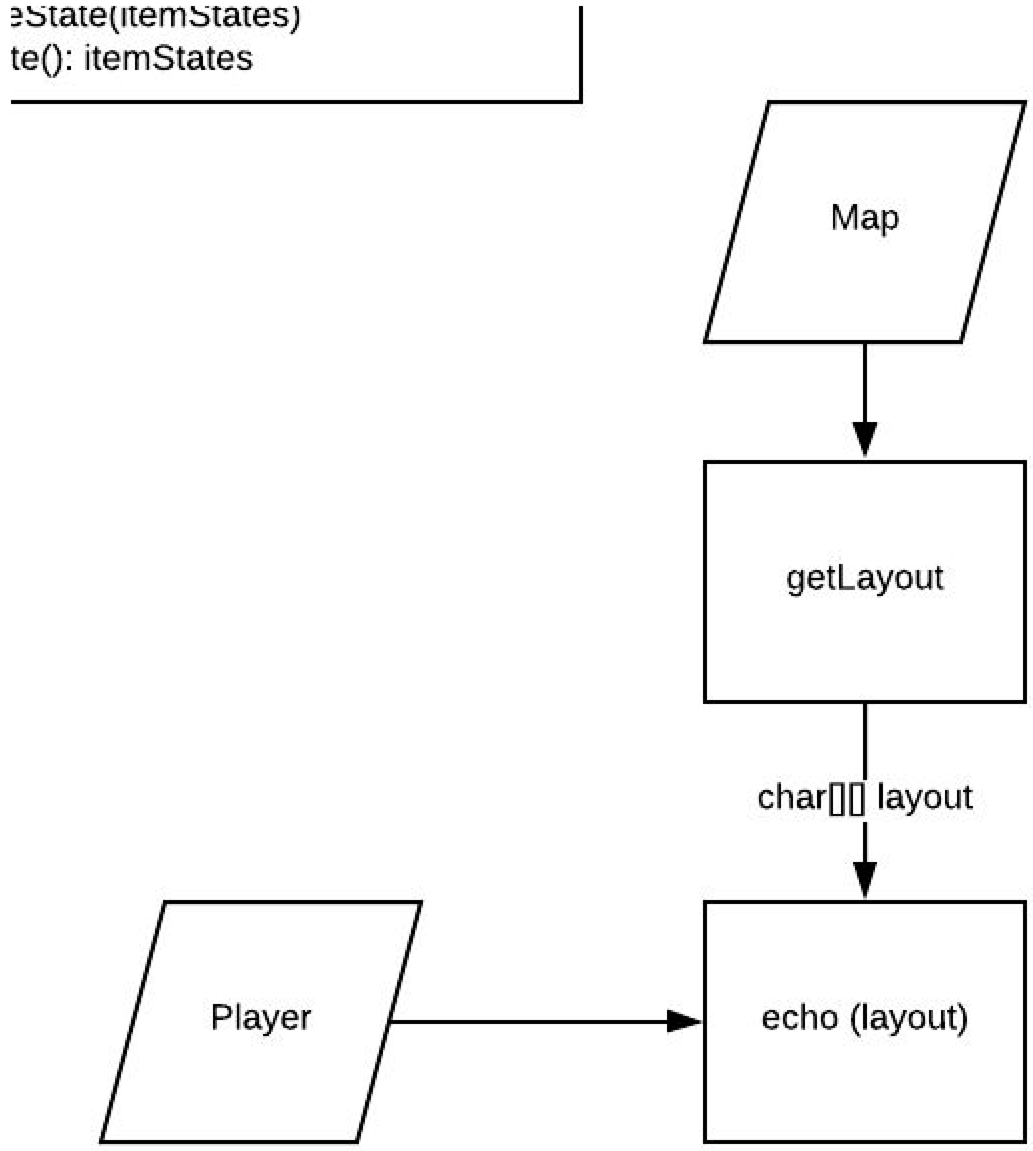
UML diagrams for any involved objects in this process flow.
If the UML diagram was presented earlier in this list simply state the reference number (3.X.3.Y).

3.6 Echolocation

3.6.1 Process Description

This process occurs when the player provides input that initiates the echo process to determine their relative position and surroundings.

3.6.2 Process Flowchart



3.6.3.Y UML for object involved

UML diagrams for any involved objects in this process flow.
If the UML diagram was presented earlier in this list simply state the reference number (3.X.3.Y).

3.7 Add Item to Inventory

3.7.1 Process Description

3.7.2 Process Flowchart

Screenshot of diagram for some application process goes here.

3.7.3.Y UML for object involved

UML diagrams for any involved objects in this process flow.
If the UML diagram was presented earlier in this list simply state the reference number (3.X.3.Y).

3.8 Save Game Data

3.8.1 Process Description

3.8.2 Process Flowchart

Screenshot of diagram for some application process goes here.

3.8.3.Y UML for object involved

UML diagrams for any involved objects in this process flow.
If the UML diagram was presented earlier in this list simply state the reference number (3.X.3.Y).

3.9 Load Game Data

3.9.1 Process Description

3.9.2 Process Flowchart

Screenshot of diagram for some application process goes here.

3.9.3.Y UML for object involved

UML diagrams for any involved objects in this process flow.
If the UML diagram was presented earlier in this list simply state the reference number (3.X.3.Y).