HSB
Hochschule Bremen
City University of Applied Sciences

Mobile Computing

Prof. Dr. Thorsten Teschke

Woven Verses

Studiengang: Internationaler Studiengang Medieninformatik,
Internationaler Frauenstudiengang Informatik

Fachsemester: 6, 4

Autoren: Sophie Brand (5194731), sbrand@stud.hs-bremen.de
Shirin Erol (5194335), serol@stud.hs-bremen.de
Lena Müller (5257164), lenmueller@stud.hs-bremen.de

Bremen, den 26. Juni 2024

Table of contents

1. Conceptualization

"Woven Verses" is a puzzle game utilizing different sensors and the control center of a smartphone. The game, designed in the style of a book with page numbers and a bookmark-shaped main menu button, consists of eight riddles; the solution of each revealing a hint to a treasure. Players can log in or register, track their progress, and compare their times on a leaderboard.

2. Functionality

Based on the concept, the app utilizes a narrative style resembling that of a book, where a story is integrated into the riddles posed in each chapter. The game also incorporates sensors such as, but not limited to, the accelerometer, the magnetic field sensor and the light sensor. Each chapter is dedicated to a different sensor or control center element. An authentication function is implemented which identifies users by an unique username and allows users to save their progress. The user's profile displays their time for each chapter, which is also used in the Leaderboard to calculate their position in a global ranking of all Woven-Verses-players. Lastly, the application offers Credits which contains a list of developers and used assets.

The Intro Chapter introduces the player to the game's functionalities by challenging them with a riddle that must be solved by turning on the flashlight. The application listens to changes in the flashlight state using the Camera Manager API.
Following up is Chapter 1 which involves the device's volume. Here, a Content Observer is used to monitor changes in the system and music volume. As the player increases the volume, the riddle text grows larger exposing the previously unreadable solution.
In Chapter 2, the player must point their phone west aided by the embedded compass. Since there are two sensors which can trigger a relevant event, an "onSensorChanged" method calculates the phone's orientation by checking if the accelerometer (values are being stored in the "gravity" array) or magnetic field sensor (values are being stored in the "geometric" array) triggered it. The rotation is converted into orientation angles, which must be in the range of 260°-280°, which describes an orientation to the west.
In Chapter 3, Android's sensor framework is used for the light sensor to detect the brightness of the room, and the goal is to darken it so a light value of less than 20 is achieved. However, the Android emulator starts with a light value of 0, which is why this should be set higher at the beginning for a better playing experience. This also becomes a problem if one would play it in a dark room on a physical device, preventing this makes no sense though.
In order to solve Chapter 4, the darktheme should be turned on. This is checked regularly using a native Compose function. In addition, this puzzle was created in such a way that it is checked whether the non-darktheme switches to darktheme, to prevent an accidental immediate solution.
To solve Chapter 5, the phone has to be put on a charger. This is logged through a Broadcast receiver that listens for changes in the battery's state. It triggers the "onReceive" method, which checks if the device is charging or fully charged.
In Chapter 6, the microphone is used to solve the puzzle. The Android Speech API was used in combination with an ActivityResultLauncher to convert the user's speech input into a string and check whether the correct code word was used. To avoid possible variations in output, several strings were used to check against to avoid problems with word-number conversion.
Lastly, take a screenshot to solve Chapter 7. A content observer and resolver is used to detect changes in the storage location of the images. A screenshot in the emulator can only be triggered via the screen and not the camera button, as the screenshot will be saved on the laptop otherwise.

3. Architecture

The architecture of "Woven Verses" is divided into two main components: Frontend and Backend. The frontend handles the user interface and sensor interactions, while the backend manages user data, authentication, and leaderboard. The project is divided into several directories, “manifests” for the project’s .xml file, which defines the structure of the application, declares all activities and specifies which features and permissions are required. The ui directory contains all graphic elements such as chapters in /chapters. Valueobjects directory contains the entity from the database. The “res” directory with several subfolders contains materials such as images and audio and some more configurations. To summarize, it can be described as a combination of modular app architecture and MVVM architecture. This structure provides a clear separation of responsibilities (Activity, Composable, Sensor-Listener, etc.) and allows for easy maintenance and expandability. The use of technologies such as Jetpack Compose and Firebase supports this architecture.

3.1. Frontend

The frontend is built using Kotlin and Android Jetpack Compose. Its key components include Activities and Fragments, which manage the different views such as the MainMenu and Chapter screens; the Composables, our reusable UI elements. We also used frameworks such as the Android Sensor Framework, the Android Speech API for microphone use or the BatteryManager API to work with the battery status. Furthermore, intents are used for the navigation of these activities. The repository can be cloned here: https://github.com/Shaygi/WovenVersesFrontend, whereas a more detailed documentation and operational manual can be found in the GitHub repository’s README: https://github.com/Shaygi/WovenVersesFrontend/blob/newmain/README.md

3.2. Backend

The backend is responsible for data storage (MySQL database), user authentication and the leaderboard. It communicates via REST with the frontend. It is currently accessible through https://mobbackend.elster.dev, but can also be hosted locally. The database stores a user's user-id (primary key), username and mail (unique), their password, the time for each chapter, whether the user completed the intro and their placement in the ranking. For a more detailed documentation including mappings, more information about the database and the process of public hosting, visit the GitHub repository’s README at: https://github.com/val8elster/WovenVersesBackend/blob/main/README.md.

4. UI

The user interface is designed to be intuitive and engaging, with a focus on a narrative-driven experience. The Key Elements of the UI are the Main Menu, the Chapter Screens, the Profile Screen and the Leaderboard. The Main Menu represents the starting point for the users. There players can navigate through the Contents, which leads to the chapter navigation, the Leaderboard, the Profile and Credits of the game. The Chapter screens are designed in a similar structure. The header contains the menu button, below is the page heading followed by the puzzle’s description, sometimes with an image or animation. The footer contains page numbers. Purple accents are used to mark significant buttons. Elements such as pop-ups, appearing buttons and animations signify that puzzles have been successfully solved and promote the user experience.

5. Sources

5.1 Images

Image 1:Patchakorn Phomin,"Lens flare light", Plattform: Vecteezy,
Url: https://www.vecteezy.com/png/8507495-lens-flare-light-special-effect

Image2: Pisut Tardging, "Treasure png with AI generated", Plattform: Vecteezy,
Url: https://www.vecteezy.com/png/26758496-treasure-png-with-ai-generated

5.2 Sounds

Sound 1: Floraphonic, "Handle Paper Foley 4", Platform: Pixabay, 3. January 2024,
Url: https://pixabay.com/de/sound-effects/handle-paper-foley-4-184013/

5.3 Tutorials

Tutorial 1: Android, "Sensor Overview", Last updated: 4. April 2024, Platform: Android Developers,
Url:
https://developer.android.com/develop/sensors-and-location/sensors/sensors_overview?hl=de

Tutorial 2: Android, "Animatable", Last updated: 4. April 2024, Platform: Android Developers,
Url:
https://developer.android.com/reference/kotlin/androidx/compose/animation/core/Animatable

Tutorial 3: Android, "Content Observer", Last updated: 4. April 2024, Platform: Android Developers,
Url: https://developer.android.com/reference/kotlin/android/database/ContentObserver

Tutorial 4: Nicolas Mage, "Jetpack Compose: How to change theme from light to dark mode programmatically onClick", 8. December 2020, Platform: Stackoverflow,
Url:
https://stackoverflow.com/questions/65192409/jetpack-compose-how-to-change-theme-from-light-to-dark-mode-programmatically-on

Tutorial 5: Android, "Voice Input", Last updated: 1. June 2024, Platform: Android Developers,
Url: https://developer.android.com/training/wearables/user-input/voice?hl=de

Tutorial 6: Android, "Monitor the Battery Level and Charging State", Last updated: 12. April 2024, Platform: Android Developers,
Url: https://developer.android.com/training/monitoring-device-state/battery-monitoring

Tutorial 7: GeeksforGeeks, "How to Build a Simple Flashlight/TorchLight Android App?", Last updated: 29. November 2022, Platform: GeeksforGeeks,
Url: https://www.geeksforgeeks.org/how-to-build-a-simple-flashlight-torchlight-android-app/

Tutorial 8: Android, "Handling changes in audio output", Last updated: 5. January 2024, Platform: Android Developers,
Url: https://developer.android.com/media/platform/output