Educational Ham Radio Simulator

Facilitating CW communication learning, one dit and dah at a time

Team **Heron:** Yasmine El Mezouari, Mehdi Mamas, Lizzy Tubbs and Youssef Chekkori

Date: December 11th, 2024

Table of contents

01
Intro

Context of our project

Target Audience

- Beginners and hobbyist interested in amateur HAM radio transmission involving CW message communication.

Goal

- Provide an engaging simulator that supports a smooth learning and practice of HAM communication skills, especially for very beginners.

Our Educational HAM Radio Simulator

- Simple and intuitive software that facilitates HAM radio communication training with no equipment purchase necessary. Anyone can chose to start learning with no financial hassle.
- No license required to start, only motivation!

- Besides allowing realistic HAM radio communication. Our software also proposes personalized training to support CW learning across multiple levels.

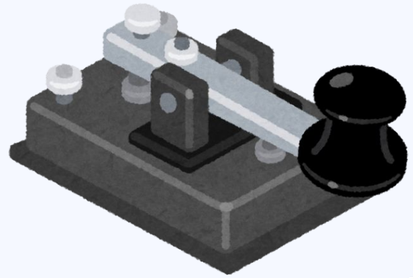
02

Key Features

What we bring to the table

Live HAM Radio

- Quick CW transmission and **peer-to-peer communication** with other users from **different computers**
- Users can access the **translation** of the message they entered
- Morse code input (straight key/paddle simulation).



Practice CW

- CW can be **practiced** at different levels with our software
- User can choose to either translate english to CW or listen to message and type it in CW
- Softwares gives **feedback** (Correct or false) to user
- **Control speed** of practice CW audio

Scenario Builder

- Users can **create, save and open** “Scenarios” to play that are **entirely customizable**.
- User can choose to add **AI Bots** that operate on specific frequencies just like actual humans would and “send” messages relating to the scenario description.

03 Demo

Let’s show off our cool software!

04 Design Principles

Design Principles

Single Responsibility Principle

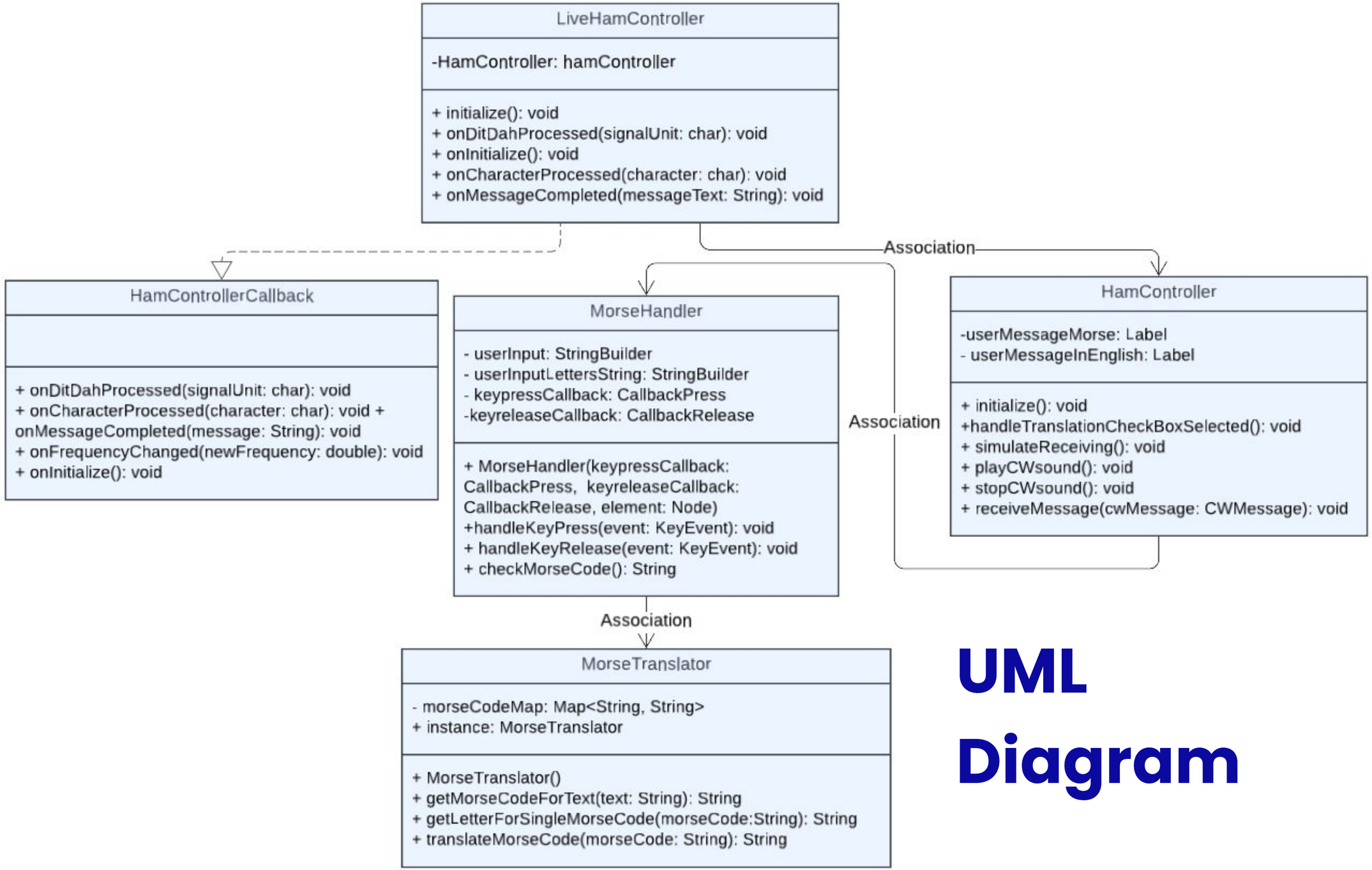
- Each class is focused on it’s own specific responsibility
- Dividing responsibility makes the codebase modular and easier to maintain
- Examples
  - TonePlayer: manages sound playback
  - MorseTranslator: translates cw into text
  - MorseHandler: manages user input and state transitions

Design Principles

Open/Closed Principle

- Extensible format
  - Easily add new callbacks
  - Expand cw dictionary without altering core logic
- Upgrade software without requiring major changes to existing code

05
UML Diagram


LiveHamController
-HamController: hamController
+ initialize(): void
+ onDitDahProcessed(signalUnit: char): void
+ onInitialize(): void
+ onCharacterProcessed(character: char): void
+ onMessageCompleted(messageText: String): void
Association
HamControllerCallback
+ onDitDahProcessed(signalUnit: char): void
+ onCharacterProcessed(character: char): void +
onMessageCompleted(message: String): void
+ onFrequencyChanged(newFrequency: double): void
+ onInitialize(): void
MorseHandler
- userInput: StringBuilder
- userInputLettersString: StringBuilder
- keypressCallback: CallbackPress
-keyreleaseCallback: CallbackRelease
+ MorseHandler(keypressCallback:
CallbackPress, keyreleaseCallback:
CallbackRelease, element: Node)
+handleKeyPress(event: KeyEvent): void
+ handleKeyRelease(event: KeyEvent): void
+ checkMorseCode(): String
Association
HamController
-userMessageMorse: Label
- userMessageInEnglish: Label
+ initialize(): void
+handleTranslationCheckBoxSelected(): void
+ simulateReceiving(): void
+ playCWsound(): void
+ stopCWsound(): void
+ receiveMessage(cwMessage: CWMessage): void
Association
MorseTranslator
- morseCodeMap: Map<String, String>
+ instance: MorseTranslator
+ MorseTranslator()
+ getMorseCodeForText(text: String): String
+ getLetterForSingleMorseCode(morseCode:String): String
+ translateMorseCode(morseCode: String): String

UML Diagram

06 Teamwork Insights

- **Collaboration:** distributed work load is always more manageable than individual work.
- **Communication and visualization**: everyone understands concepts differently, so communicating about how to move forward prevents confusion when deadlines approach.
- **Agile development:** having a working version of a software to expand and develop is better than waiting to release a perfect version.
- **Push frequently!**

07

Acknowledgements

08

Conclusion

Concluding and Q&A session

User Engagement

Our simulator allows beginners to practice HAM radio skills effectively

No equipment necessary

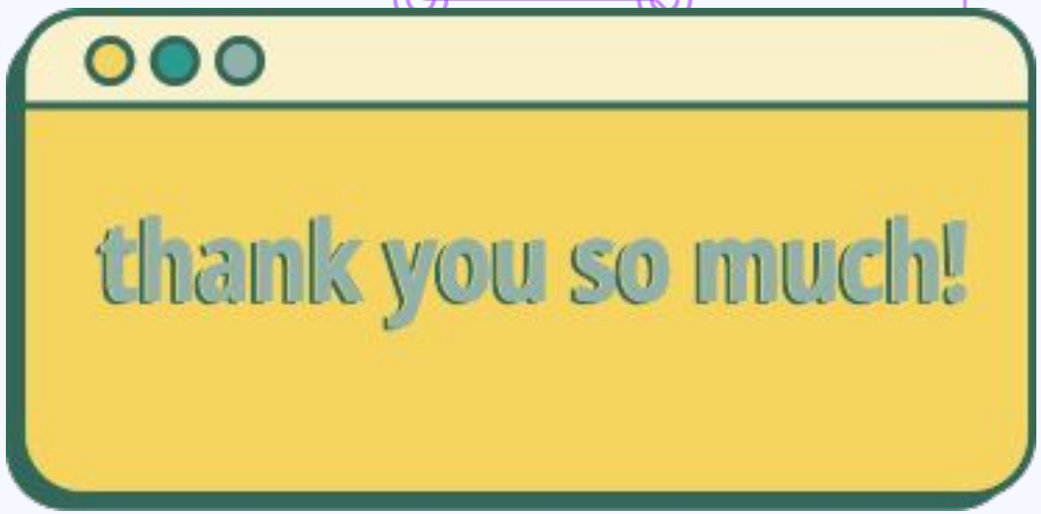
Get started and improve with no complicated hardware

CW Proficiency

Tailored to provide CW learning and improving

Scenario Building

Users can create and completely customize a “scenario” to play using AI bots.



Thank you to everyone for their attention, special thanks to all the collaborators, Dr. Stonedahl and Dr. Vincent for the help on this project