

Projekt Signalverarbeitung: Morsecode

Matthias Holzinger, 0256542 & Christian Humer 0957891,

Teil A & B)

Alle Sourcen liegen im Order „Morse“ als importierbares Eclipse-Projekt zur Verfügung.

Die Applikation kann auf zwei Arten gestartet werden. Mit Argumenten wird die Kommandozeilenversion gestartet ohne Argumente die GUI Version. Um die GUI Version zu starten kann man folgendend Befehl verwenden:

```
java -jar morse.jar
```

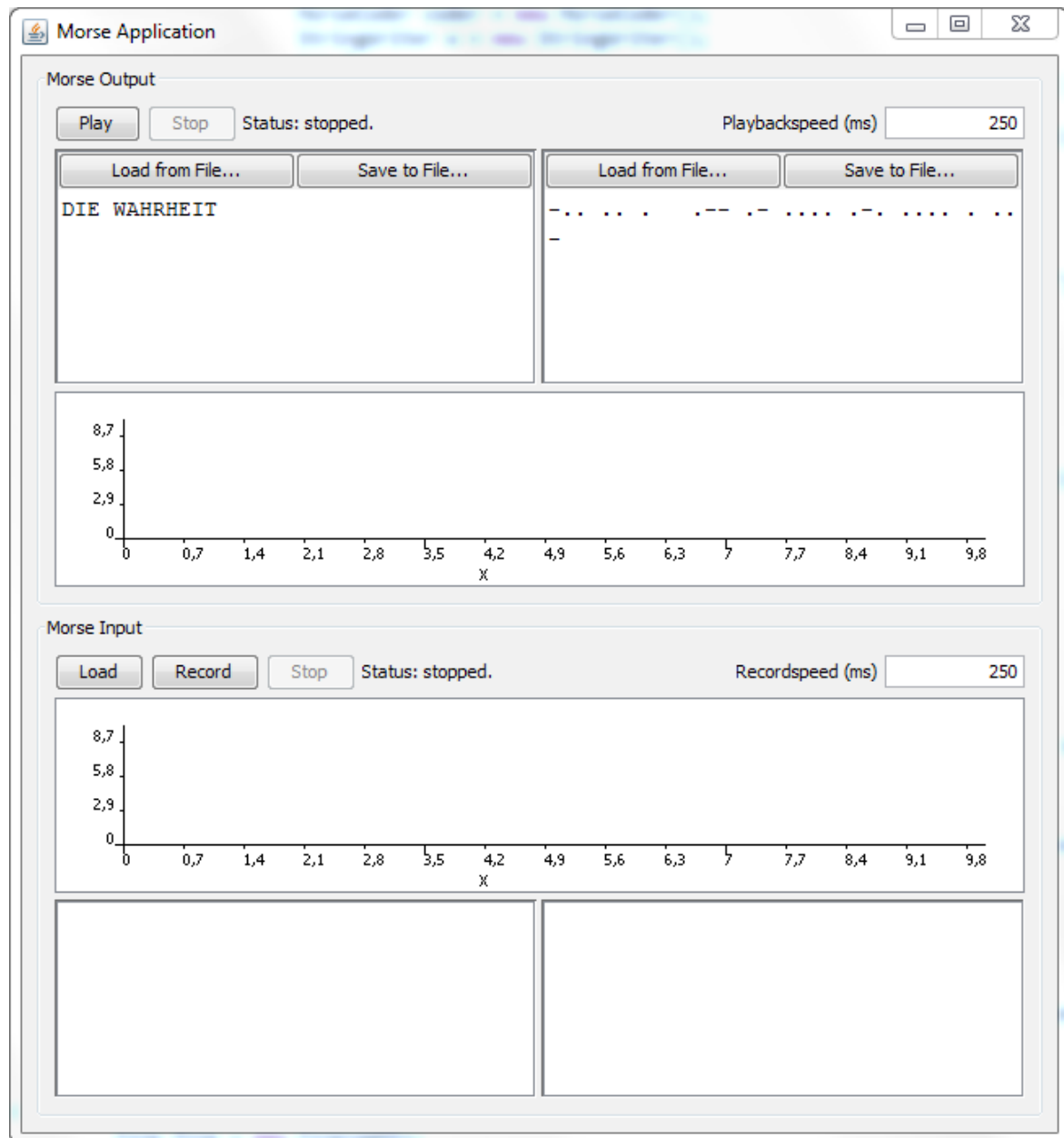
In der Kommandozeilenversion können folgende Argumente verwendet werden:

```
usage: CommandLineApplication -e|d <filename> [-s <filename>] [-l
                                <length>] [-h]
  -d <file>          decode morse code to text
  -e <file>          encode text as morse code
  -h                 print this message
  -l <length>        set the dot length
  -s <file>          store result in file
```

```
java -jar morse.jar -e input.txt -s output.txt
  -.. .. .   .-- .-  .... .-. .... .  -
```

```
java -jar morse.jar -e input.txt -s output.txt
  DIE WAHRHEIT
```

Über unsere GUI lassen sich alle Funktionen viel leichter bedienen und alle Funktionen verwenden.



En- & Decodierung mithilfe eines Binärbaumes in MorseCoder.java

Sample1.txt decodiert:
THE ANSWER IS 42

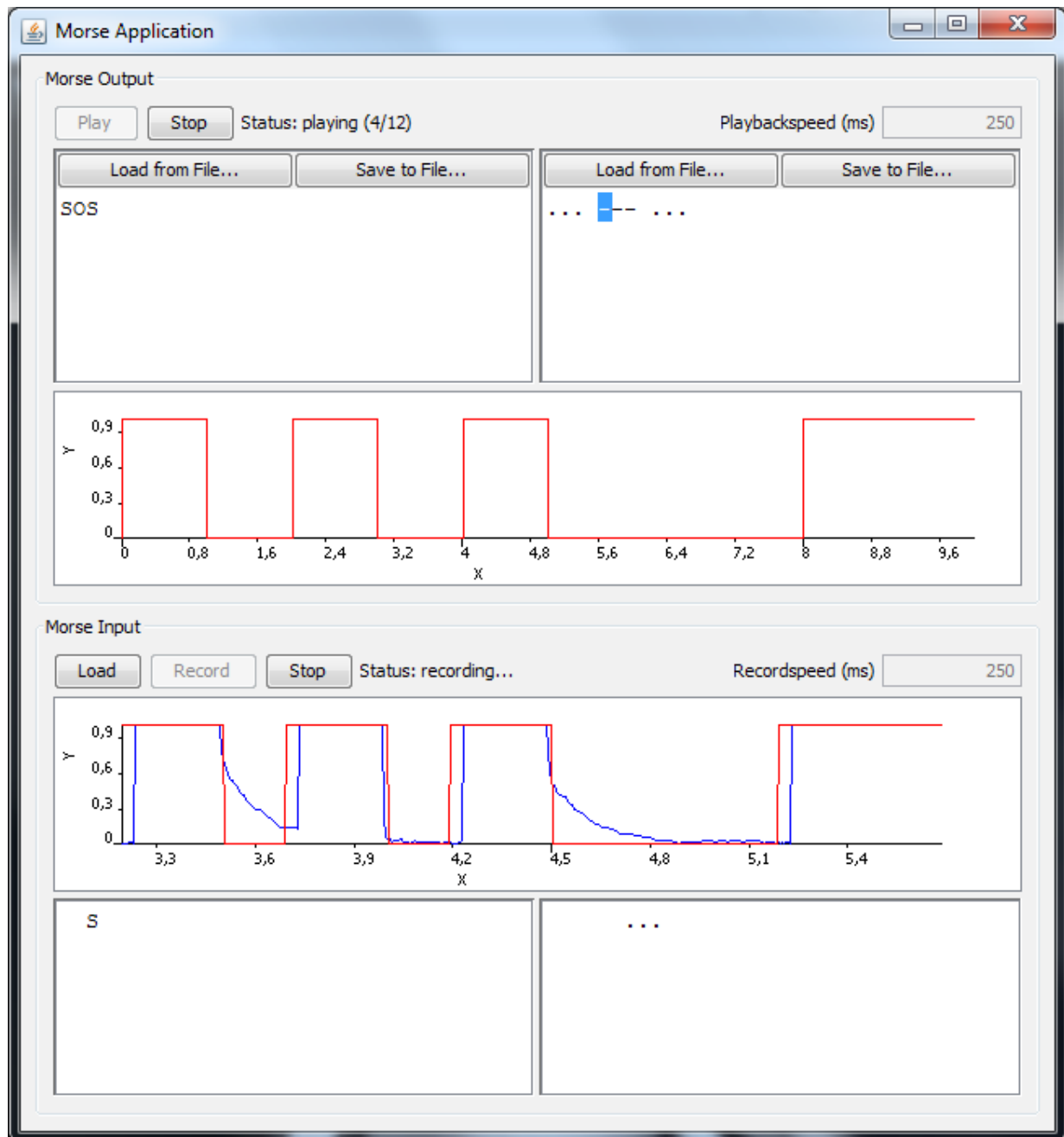
Sample2.txt decodiert:
THERE IS A THEORY WHICH STATES THAT IF EVER ANYONE DISCOVERS EXACTLY
WHAT THE UNIVERSE IS FOR AND WHY IT IS HERE IT WILL INSTANTLY DISAPPEAR
AND BE REPLACED BY SOMETHING EVEN MORE BIZARRE AND INEXPLICABLE STOP
THERE IS ANOTHER THEORY WHICH STATES THAT THIS HAS ALREADY HAPPENED
STOP THERE IS A THIRD THEORY WHICH SUGGESTS THAT BOTH OF THE FIRST TWO
THEORIES WERE CONCOCTED BY A WILY EDITOR OF THE HITCHHIKERS GUIDE TO
THE GALAXY IN ORDER TO INCREASE THE UNIVERSAL LEVEL OF UNCERTAINTY
AND PARANOIA AND SO BOOST THE SALES OF THE GUIDE STOP THIS LAST THEORY

IS OF COURSE THE MOST CONVINCING AS THE HITCHHIKERS GUIDE TO THE GALAXY IS THE ONLY BOOK IN THE WHOLE OF THE KNOWN UNIVERSE TO HAVE THE WORDS DONT PANIC INSCRIBED IN LARGE FRIENDLY LETTERS ON THE COVER

Teil C)

Wir haben uns Entschieden die Decodierung in Echtzeit mithilfe der `javax.sound` und der `TarsosDSP` Libraries direkt in Java zu implementieren.

Über das GUI kann entweder aufgenommen oder eine AudioDatei gelesen werden.



Die Bibliothek liefert nur die rohen Funktionswerte der Sinuskurven, die dann von uns mittels eines Schwellwertes zu 1 bzw. 0 konvertiert werden.

Als Signalübertragungsfrequenz haben wir uns nach einigen Tests für die Empfohlenen 2kHz entschieden, weil wir keine relevanten Unterschiede beim erkennen des Signals bemerkt haben.

Als dot-length haben wir für WAV-Dateien eine Erkennung bis zu 150ms geschafft, bei direkten Audio Aufnahmen nur bis 250 ms. Danach werden die Pausen nicht mehr richtig erkannt.