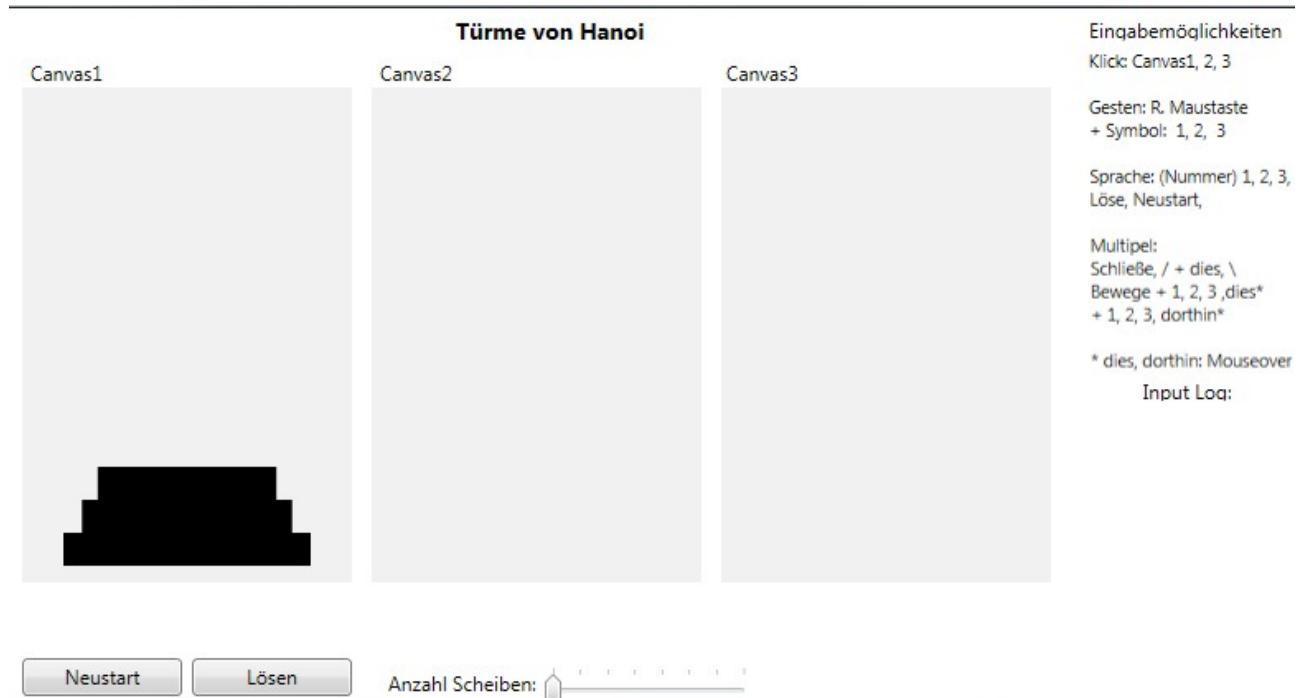


# Dokumentation Hanoi

Von Duc Anh Trinh und Stephanie Schöne



1. Ziel des Spiels
2. Aufbau der Benutzeroberfläche
3. Spielmodi
4. Eingabemöglichkeiten
5. Programmcode

# 1. Ziel des Spiels

Ziel des Spiels ist es alle Scheiben durch Interaktionen von der linken Fläche (Canvas1) zur rechten Fläche (Canvas3) zu bewegen. Dabei dürfen nie größere Scheiben auf kleinere gelegt werden.

Abbildung 1 stellt das Resultat dar.

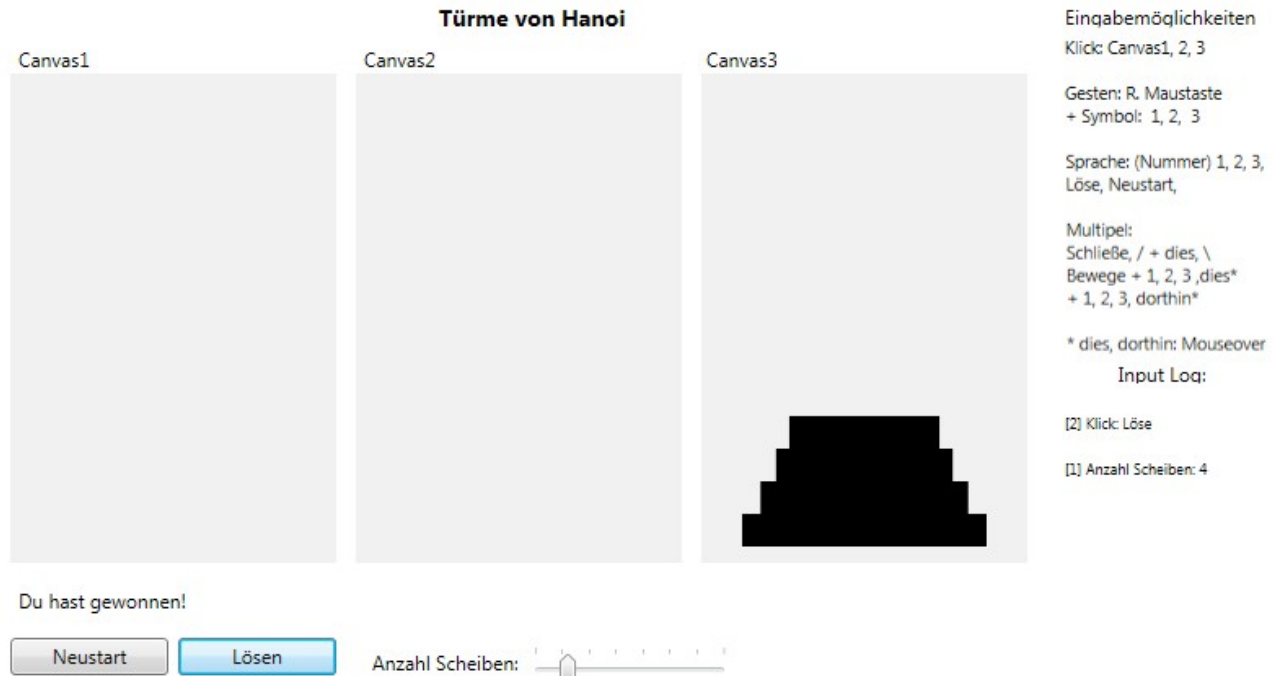


Abbildung 1: Erfolgreiches Absolvieren des Spiels

## 2. Aufbau der Benutzeroberfläche

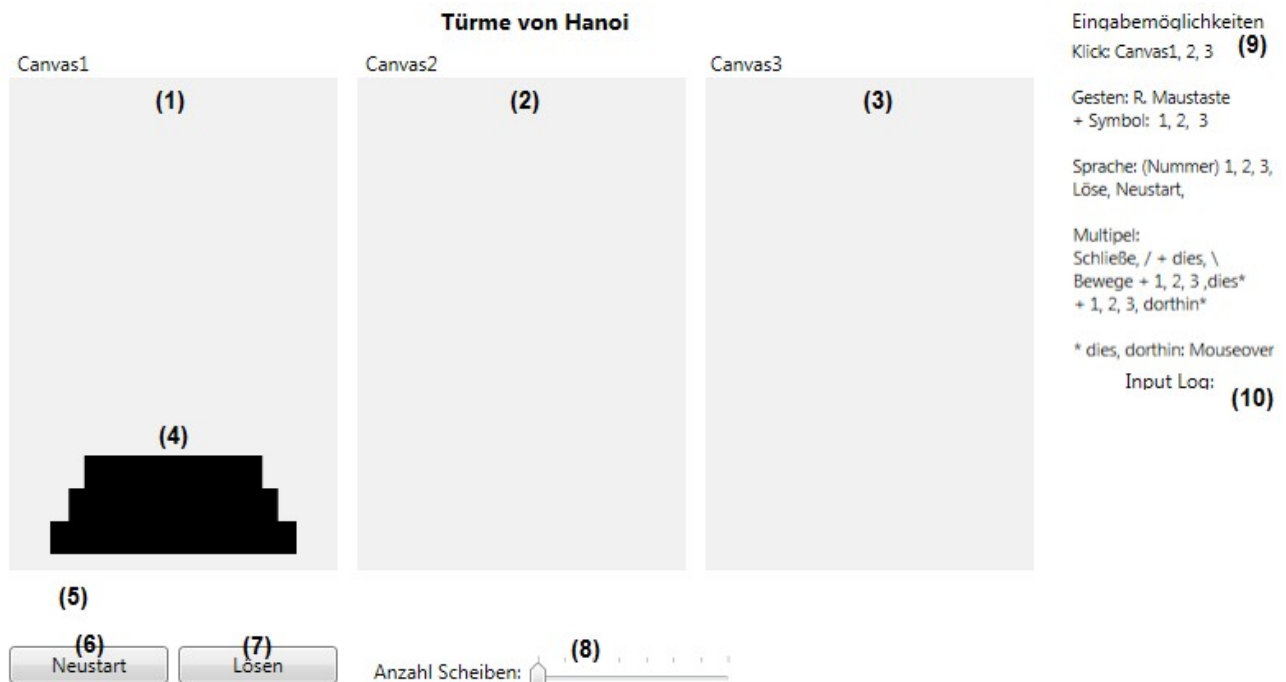


Abbildung 2: Benutzeroberfläche der Anwendung mit nummerierten Programmelementen

Folgend werdend die Elemente der Benutzeroberfläche, wie in Abbildung 2 nummeriert, beschrieben.

Nummer	Funktionsweise
(1)	Canvas1: Alle Scheiben werden dort zu Spielbeginn gestapelt.
(2)	Canvas2: Diese Fläche dient zur Zwischenlagerung von Scheiben.
(3)	Canvas2: Hierhin müssen die Scheiben der Größe nach (groß unten, klein oben) sortiert werden.
(4)	Scheiben: Die schwarzen, langen Rechtecke sind die zu verschiebenden Scheiben.
(5)	In dieser Zeile werden Informationen zu getätigten Eingaben ausgegeben.
(6)	Neustart Button: Über diesen Button kann das Spiel zurückgesetzt werden.
(7)	Lösen Button: Dieser Button erlaubt das automatische Lösen des Spieles ausgehend von der Ausgangsposition.
(8)	Schieberegler: Mit diesem Regler kann die Anzahl von Scheiben eingestellt werden. Das Minimum beträgt 3 Scheiben.
(9)	Die Legende gibt Hinweise auf mögliche Eingaben.
(10)	Unterhalb von „Input Log“ werden die zuletzt getätigten Eingaben mit Eingabemodalität sowie Funktion aufgelistet.

## 3. Spielmodi

### 3.1 Manuelles Spielen

Das Spiel kann durch verschiedene Eingabemodalitäten manuell gespielt werden. Dabei muss für jeden Zug ein Canvas, von dem die obere Scheibe genommen werden soll, gewählt werden. Nur Flächen mit Scheiben werden für diese Auswahl angenommen.

Wurde eine gültige Auswahl getroffen, wird der Canvas grün gefärbt. Für die Verschiebung muss anschließend ein weiterer Canvas gewählt werden. Hier gilt: Größere Scheiben dürfen nicht auf Kleinere platziert werden.

Wurden alle Scheiben der Größe nach (unten groß, oben klein) von **(1)** nach **(3)** sortiert, ist das Spiel gewonnen.

Durch den Neustart Button **(6)** kann das Spiel jederzeit von vorn begonnen werden. Das Verändern der Scheibenanzahl **(8)** bewirkt ein automatisches Zurücksetzen.

Das Automatische Lösen funktioniert nur im Ausgangszustand.

### 3.2 Automatisches Spielen

Befindet sich das Spiel im Ausgangszustand, kann das Spiel automatisch gelöst werden. Falls es nicht im Ausgangszustand ist, wird eine Fehlermeldung in der Informationszeile **(5)** dargestellt.

Da für diesen Modus ein paralleler Thread verwendet wird, ist die Benutzeroberfläche noch immer verwendbar. Durch den Neustart Button **(5)** kann das automatische Lösen jederzeit abgebrochen werden. Bei Veränderung der Scheibenanzahl **(8)** wird das Spiel ebenfalls zurückgesetzt.

Wird während des Vorgangs des automatischen Lösens das automatische Lösen erneut gewählt, erscheint eine Fehlermeldung. Der Vorgang wird nicht unterbrochen.

Nach dem Lösevorgang muss für ein erneutes Spielen das Spiel neugestartet werden.

## 4. Eingabemodalitäten

### 4.1 Einteilige Eingaben

Funktion	Eingabe	Eingabemodalität
Auswahl Canvas 1,2,3	Sprechen von „1“, „2“, „3“, „Nummer 1“, „Nummer 2“ oder „Nummer 3“	Sprache
	Rechte Maustaste gedrückt halten und die Ziffer des gewünschten Canvas zeichnen: 1, 2 (mit langem unteren Strich) oder 3	Geste
	Anklicken des gewünschten Canvas	Klick
Neustart des Spieles	Sprechen von „Neustart“	Sprache
	Anklicken des Neustart Button (6)	Klick
Automatisches Lösen	Sprechen von „Löse“	Sprache
	Anklicken des Lösen Button (7)	Klick
Verändern der Scheibenanzahl	Bedienen des Schiebereglers	Klick (Drag)

### 4.2 Mehrteilige Eingaben

Ab der 1. Eingabe, ist die nächste Eingabe innerhalb von 10 Sekunden zu tätigen. Andernfalls wird die multiple Eingabe zurückgesetzt.

Funktion	Eingabeteile		
	1. Eingabe	2. Eingabe	3. Eingabe
Bewegen einer Scheibe	Sprechen von „Bewege“	Klick auf Canvas (1, 2, 3)	Klick auf Canvas (1, 2, 3)
		Gestenauswahl Canvas (1, 2, 3)	Gestenauswahl Canvas (1, 2, 3)
		Sprachauswahl Canvas (1, 2, 3)	Sprachauswahl Canvas (1, 2, 3)
		Sprechen von „dies“ & Maus über Canvas	Sprechen von „dorthin“ & Maus über Canvas
Schließen der Anwendung	Sprechen von „Schließe“	Sprechen von „dies“	-
	Geste „/“	Geste „\“	-

## 4. Programmcode

Das Programm unterteilt sich in die Klassen:

- MainWindow.xaml.cs
- Speech.cs
- Gesture.cs
- MultiInput.cs
- Feedback.cs

### 4.1 MainWindow.xaml.cs

Diese Klasse wickelt alle Interaktionen mit und auf der Benutzeroberfläche ab. Sie übernimmt folgende Aufgabenbereiche:

- Initialisierung der Eingabeklassen
- Verarbeitung der Eingaben (für Geste & Sprache nach Interpretation durch die Eingabeklassen)
- Überprüfung von Bedingungen von Eingaben & Zielfunktionen
- Ausführung der Zielfunktionen

Für die weiteren Beschreibungen wird sie „Main“-Klasse genannt.

### 4.2 Speech.cs

Durch eine Spracherkennung werden Spracheingaben mit hinterlegten Befehlen verglichen. Bei erfolgreicher Spracheingabe, wird die Zielfunktion an die Main-Klasse gereicht.

### 4.3 Gesture.cs

Gestenvorlagen für 1, 2, 3 sowie die Schließgesten befinden sich im XML-Format im Projekt. Nach einer Gesteneingabe werden die Distanzen zwischen Vorlagen und Eingabe durch einen DTW-Algorithmus berechnet. Die Vorlage mit der geringsten Distanz wird zur Interpretation der Eingabe verwendet. Die Zielfunktion wird an die Main-Klasse weitergereicht.

### 4.4 MultiInput.cs

Diese Klasse dient als Container für mehrteilige Eingaben. Sie setzt sich aus einem Key-, einem Source- sowie einem Goal-Slot zusammen. Sie werden mit interpretierten Funktionstypen gefüllt. Die Interpretation erfolgt zuvor durch die jeweiligen Eingabeklassen.

Die Initialisierung erfolgt bei der „Bewege“ Spracheingabe oder dem ersten Schließbefehl. Die

jeweiligen Slots können nur bei Einhalten der Befehlssequenzen (Siehe obige Tabelle) befüllt werden. Bei einer falschen Eingabe oder dem Ablaufen des Zeitintervalls von 10 Sekunden, der nach jeder Eingabe für die nächste bereit steht, wird der MultiInput zurückgesetzt. So bald die Befehlssequenzen abgeschlossen sind, werden sie durch die Main-Klasse interpretiert.

## **4.5 Feedback.cs**

Diese Klasse bearbeitet die Ausgabe von Informationen in Labels.

- Eingabehistorie unter „Input Log“
- Statusinformationen und Fehlermeldungen unterhalb der Canvas