



Hochschule für Technik,
Wirtschaft und Kultur Leipzig

FAKULTÄT INGENIEURWISSENSCHAFTEN

E492.1 - EMBEDDED SYSTEM 1

Tetris - Beleg

Author Florian Wolf, Johannes Both
Betreuer Marco Braun

14. März 2024

Inhaltsverzeichnis

1	Aufgabenstellung	1
2	Grundlegende Überlegungen	2
2.1	Auswahl Display	2
2.1.1	Pico Display Pack von Pimoroni	2
2.1.2	Pico-Eval-Board von Waveshare	2
2.1.3	Entscheidung	3
2.2	C/C++ oder MicroPython?	3
2.2.1	C/C++	3
2.2.2	MicroPython	4
2.2.3	Entscheidung	4
3	Tetris	5
3.1	Geschichte	5
3.2	Logik	5
	Abbildungsverzeichnis	7
	Literatur	7

1 Aufgabenstellung

Das Projekt „Tetris“ wurde uns am 23.11.2023 von Herrn Marco Braun übergeben. Dieses soll während des Semesters in der Hochschule durchgeführt werden, in einem Zeitraum von 15 Wochen.

Unsere Aufgabe ist es, das Spiel "Tetris" mithilfe eines Raspberry Pi Pico und einem Display umzusetzen.

Die zentralen Aufgaben umfassen primär die Programmierung des Spiels auf dem Raspberry Pi Pico. Parallel dazu erfordert die Auswahl eines geeigneten Displays eine sorgfältige Abwägung, gefolgt von der Integration und Programmierung in Verbindung mit dem Raspberry Pi.

Bei der Realisierung des gesamten Projekts sind uns keine einschränkenden Rahmenbedingungen auferlegt. Wir genießen die Freiheit, die Programmiersprache sowie die Hardware nach eigenem Ermessen zu selektieren. Die einzige vorgegebene Komponente ist der Raspberry Pi Pico.

Das Projekt soll am 18.09 abgegeben werden.

2 Grundlegende Überlegungen

2.1 Auswahl Display

2.1.1 Pico Display Pack von Pimoroni

Wir haben zuerst mit dem Pico Display Pack von der Firma Pimoroni begonnen das Projekt umzusetzen. Dieses LCD-Display ist 1,14“ klein mit einer Auflösung von 240 x 135 Pixel und besitzt vier integrierte Buttons.

Nach der Inbetriebnahme des Displays und erfolgreicher Herstellung der Verbindung zum Raspberry Pi haben wir mit den ersten Programmierungen und Visualisierungen begonnen. Allerdings wurde uns nach einer kurzen Zeit sehr schnell klar, dass wir dieses Display nicht final benutzen werden. Zum einen ist die Größe des Displays zu klein für uns, weshalb die Blöcke, welche das Spiel ausmachen, ebenfalls sehr klein sein müssten, was zu keinem schönen Spielerlebnis führen würde. Die vorgegebene Position der Buttons lässt keinerlei Spielraum für eine flexible Integration gemäß unseren individuellen Gestaltungspräferenzen im späteren Design.

Aufgrund der dargelegten Gründe haben wir beschlossen, keine weiteren Ressourcen in die Fortsetzung der Programmierung dieses Displays zu investieren.



Abbildung 1: *Pico Display Pack von Pimoroni 2024*

2.1.2 Pico-Eval-Board von Waveshare

Anschließend griffen wir auf das Pico-Eval-Board der Firma Waveshare zurück. Dieses ist ein LCD-Touch-Display in 3,5“ mit einer Auflösung von 480 x 320 Pixel. Zudem besitzt es viele Features, wie ein 3,5mm Klinkenanschluss, Micro SD Kartenslot, RGB LED und viele weitere.

Durch die Größe des Displays, hatten wir nun genug Platz, um die Blöcke für das Spiel ausreichend groß zu gestalten. Zudem konnten wir durch das Touch-Display unsere Buttons individuell auf dem gesamten Display platzieren, wie wir es wollten.



Abbildung 2: *Pico-Eval-Board* von Waveshare 2024

2.1.3 Entscheidung

Nach sorgfältigen Überlegungen haben wir uns dafür entschieden, das Pico-Eval-Board der Firma Waveshare zu verwenden. Dieses Board beeindruckt mit einem 3,5-Zoll LCD-Touch-Display, das eine Auflösung von 480 x 320 Pixeln bietet. Da dieses Display ebenfalls eine Touch-Funktion besitzt, hat unsere Wahl nur begünstigt, da herkömmliche Knöpfe als veraltet gelten und die Touch-Technologie dem aktuellen Stand der Zeit entspricht. Zudem rechtfertigt die Fülle an Features den geringfügigen Mehrpreis im Vergleich zum Display von Pimoroni mehr als angemessen und ist daher akzeptabel.

Die großzügige Größe des Displays ermöglichte es uns, die Blöcke für das Spiel ausreichend groß zu gestalten. Darüber hinaus bot das Touch-Display die Möglichkeit, die Buttons individuell auf dem gesamten Bildschirm zu platzieren, ganz nach unseren Vorstellungen. Diese Entscheidung eröffnete uns neue kreative Freiheiten bei der Gestaltung und Optimierung des Spielerlebnisses.

2.2 C/C++ oder MicroPython?

C/C++ und MicroPython sind zwei Programmiersprachen, die sich in Bezug auf Anwendungsgebiete, Vor- und Nachteile unterscheiden. C/C++ sind traditionelle, leistungsstarke Sprachen, die oft für systemnahe Programmierung, hardwarenahe Softwareentwicklung und ressourcenintensive Anwendungen eingesetzt werden. MicroPython hingegen ist eine schlankere Variante von Python, die auf Mikrocontrollern und Embedded-Systemen verwendet wird.

2.2.1 C/C++

Vorteile:

1. **Leistung:** C/C++ bieten eine hohe Leistungsfähigkeit, da sie hardwarenah arbeiten und optimierten Maschinencode erzeugen können.

-
2. **Hardwarekontrolle:** Diese Sprachen ermöglichen eine präzise Kontrolle über Hardware-Ressourcen, was in eingebetteten Systemen und systemnahen Anwendungen wichtig ist.

Nachteile:

1. **Komplexität:** Die Syntax von C/C++ kann komplex sein, was die Entwicklung und Wartung erschweren kann.
2. **Längere Entwicklungszeiten:** Projekte in C/C++ können tendenziell länger dauern, da mehr Code geschrieben werden muss.

Einsatzgebiete:

1. **Embedded Systems:** C/C++ werden häufig in eingebetteten Systemen verwendet, wo direkte Hardwarekontrolle erforderlich ist.
2. **Betriebssysteme:** Viele Teile von Betriebssystemen werden in C/C++ geschrieben.

2.2.2 MicroPython

Vorteile:

1. **Einfachheit:** MicroPython bietet eine einfachere Syntax, die besonders für Entwickler ohne umfangreiche Erfahrung geeignet ist.
2. **Schnelle Entwicklung:** Dank der klaren Syntax und des schnellen Testzyklus ist die Entwicklung in MicroPython oft schneller als in C/C++.
3. **Community-Unterstützung:** MicroPython profitiert von der großen Python-Community, was den Zugang zu Bibliotheken und Ressourcen erleichtert.

Nachteile:

1. **Leistungseinschränkungen:** Im Vergleich zu C/C++ ist MicroPython in Bezug auf Leistung eingeschränkter, was in ressourcenintensiven Anwendungen relevant sein kann.
2. **Nicht für alle Anwendungen geeignet:** Aufgrund von Ressourcenbeschränkungen ist MicroPython nicht für alle Anwendungsfälle geeignet, insbesondere nicht für hochleistungsfähige oder hardwarenahe Anwendungen.

Einsatzgebiete:

1. **IoT-Geräte:** MicroPython wird häufig für die Entwicklung von IoT-Geräten eingesetzt, bei denen die Programmierung von Ressourcenbeschränkungen geprägt ist.
2. **Prototyping:** Aufgrund seiner Einfachheit eignet sich MicroPython gut für schnelle Prototypenentwicklung.

2.2.3 Entscheidung

Nach sorgfältiger Abwägung der Vor- und Nachteile haben wir uns für MicroPython entschieden. Die einfache Syntax, schnelle Entwicklungszeit und bessere Dokumentation im Internet sind in unserem speziellen Anwendungsfall entscheidende Faktoren, die zu einer effizienteren Programmierung beitragen. Mit Thonny steht ebenfalls eine einfache und leicht zu verstehende Entwicklungsumgebung für MicroPythonprogramme zur Verfügung.

3 Tetris

3.1 Geschichte

Tetris ist eines der ikonischsten Videospiele der Welt. Es wurde im Jahr 1984 von dem sowjetischen Informatiker Alexey Pajitnov entwickelt. Der Name Tetris leitet sich von den griechischen Wörtern "tetra" (Vier) und "tennisäb", was für die Grundidee des Spiels steht.

Ursprünglich auf Mainframe-Computern in der Sowjetunion gespielt, erregte Tetris schnell die Aufmerksamkeit der westlichen Welt. Die Veröffentlichung auf der Internationalen Elektronikmesse in Las Vegas im Jahr 1988 markierte den Beginn seiner globalen Bekanntheit.

Die Rechte an Tetris waren in den Händen des sowjetischen Staates, was zu einem komplexen Lizenzierungsprozess führte. Nintendo sicherte sich die Rechte für den Game Boy und veröffentlichte Tetris 1989 als Teil des Game-Boy-Bundles, was maßgeblich zur Popularität des tragbaren Systems beitrug und festigte Tetris als Klassiker.

3.2 Logik

Die Logik hinter dem Spiel Tetris ist elegant in ihrer Einfachheit und gleichzeitig herausfordernd in ihrer Umsetzung. Das grundlegende Ziel des Spiels besteht darin, die auf dem Bildschirm erscheinenden Tetrominos, eine geometrische Formen bestehend aus jeweils vier Blöcken, so zu platzieren, dass sie vollständige horizontale Linien bilden.

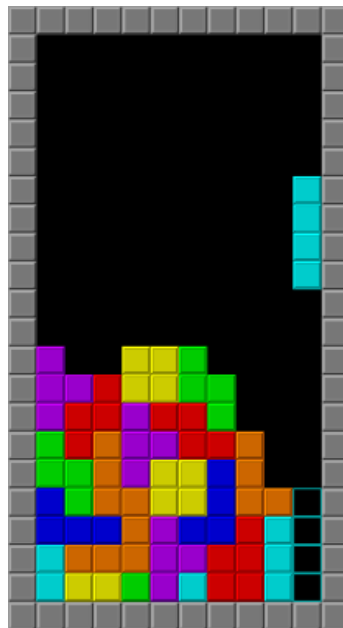


Abbildung 3: *Typisches Tetrisbild* 2024

Die Tetrominos fallen kontinuierlich von oben in das Spielfeld herab und die Spieler müssen rasch entscheiden, wie sie die Formen drehen und verschieben, um Lücken zu füllen und Linien zu vervollständigen. Sobald eine horizontale Linie vollständig ausgefüllt ist, verschwindet sie und die darüberliegenden Tetrominos rutschen nach unten, um Platz zu schaffen.

Die Herausforderung von Tetris besteht in der ständig steigenden Geschwindigkeit des Fallens der Tetrominos und der begrenzten Reaktionszeit des Spielers. Die Entscheidungen müssen schnell getroffen werden, um das Spielfeld nicht zu überfüllen. Ein überfülltes Spielfeld bedeutet das Ende des Spiels.

Die Logik des Spiels erfordert nicht nur schnelle Hand-Augen-Koordination, sondern auch strategisches Denken. Spieler müssen vorausschauend planen und Lücken im Spielfeld im Auge behalten, um Platz für kommende Tetrominos zu schaffen. Jede Entscheidung beeinflusst die Dynamik des Spiels und hat Auswirkungen auf die Möglichkeit, weitere Linien zu vervollständigen.

Tetris ist damit nicht nur ein Test der Geschicklichkeit, sondern auch ein mentales Puzzlespiel, das eine Balance zwischen Reaktionsfähigkeit und strategischem Denken erfordert. Die stetig steigende Intensität des Spiels macht es zu einer zeitlosen Herausforderung, die Spieler aller Altersgruppen fesselt.

Abbildungsverzeichnis

1	Pico Display Pack von Pimoroni	2
2	Pico-Eval-Board von Waveshare	3
3	Typisches Tetrisbild	5

Literatur

- Chat-GPT* (2024). URL: <https://chat.openai.com/?next=%2Fchat> (besucht am 7. Jan. 2024).
- NTNU, Department of Marine Technology (2020). *IMT Software Wiki - LaTeX*. URL: <https://www.ntnu.no/wiki/display/imtsoftware/LaTeX> (besucht am 15. Sep. 2020).
- Pico Display Pack von Pimoroni* (2024). URL: <https://shop.pimoroni.com/products/pico-display-pack?variant=32368664215635> (besucht am 1. Feb. 2024).
- Pico-Eval-Board von Waveshare* (2024). URL: <https://www.waveshare.com/pico-eval-board.htm> (besucht am 2. Feb. 2024).
- Typisches Tetrisbild* (2024). URL: <https://en.wikipedia.org/wiki/Tetris> (besucht am 10. Feb. 2024).