Wissensspiel in Python und Tkinter:

“Letzte Karte”

Klausurersatzleistung Q1.1

bei Gerrit Herrmann

von *Amos Peter* und *Thilo Butt*

Abgabe 10.01.2023

Aufgabe

Die Aufgabe war die Implementierung eines Spiels in Python. Gefragt waren Wissensspiele, nach Rücksprache mit dem Tutor war aber auch anderes möglich. Diese Spiele werden mit Hilfe einer grafischen Benutzeroberfläche mit Tkinter programmiert.
Es handelt sich um eine Partnerarbeit mit zwei Personen. Die Projektabgabe erfolgt am 08.01.2023 und beinhaltet neben dem Programm eine Präsentation und diese Projektdokumentation.
Weitere Anforderungen waren zum Beispiel Lesen und Schreiben von Dateien zum Laden und Speichern von Programmdaten und Spielständen und Unterschiedliche Modi.

Konzept

Idee

Unsere Idee war es, ein Kartenspiel in Python umzusetzen, aber hierbei nicht einfach ein Spiel zu kopieren. Deswegen haben wir nach einem simplen, bereits vorhandenen Spiel gesucht, dass wir modifizieren können und sind auf Mau Mau gestoßen, welches grundlegend für viele verschiedene Kartenspiele ist. Wir haben uns vorgenommen, dieses nach eigenem Ermessen umzusetzen. Außerdem wollten wir computergesteuerte Spieler programmieren und so die Vorteile eines Videospiels nutzen.

Spielregeln

Das Ziel ist es, als erster Spieler alle Karten loszuwerden. In jeder Runde kann man eine passende Karte aus seiner eigenenHand ablegen. Eine Karte ist passend, wenn sie die gleiche Farbe oder die gleiche Zahl wie die oberste Karte auf dem Ablagestapel hat. Wenn man keine passende Karte hat, muss man solange vom Ablagestapel ziehen, bis man eine passende Karte gezogen hat. Wenn man gelegt hat, sind die anderen Spieler an der Reihe.

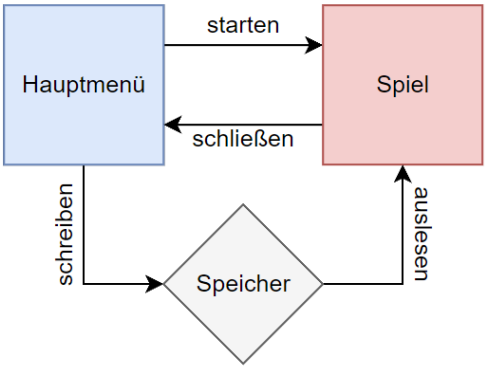
Entwicklung

Herangehensweise

Thilo hat damit angefangen, das GUI mit den zugehörigen Elementen zu entwickeln. Als nächstes wurde die Logik programmiert. Die dann implementierten computergesteuerten Spieler greifen auf dieses Grundgerüst zu.
Da Thilo bereits früher angefangen hat, haben wir beschlossen, dass Amos das Menü entwickelt. Dadurch, dass wir nicht beide am selben Code gearbeitet haben, was rückblickend eine sehr gute Entscheidung war, herrschte eine klare Arbeitstrennung. Weil wir zwei Dateien hatten, wurde ein effizienter Weg benötigt Information zwischen diesen beiden Dateien auszutauschen. Weil es sich auch zum Speichern der Einstellungen angeboten hat, haben wir uns für das Zwischenspeichern in einer Textdatei entschieden, welches Amos implementiert hat. Zuletzt hat Amos GUI und Funktionen für das Menü und die Einstellungen erstellt.

Struktur

Durch das Starten des Programmes, öffnet sich das Menü: Zum einen kann man hier Einstellungen vornehmen, die dann in einer Textdatei zwischengespeichert werden, zum anderen kann man von dort das Spiel starten und beenden. Dieses liest die Speicherdatei aus und kann so vorgenommen Änderungen anwenden.



Wichtige Programmbestandteile

Menü

Das Menü besteht aus drei simplen Buttons, die jeweils eine Funktion aufrufen, von denen eine sogar nur eine Python-interne Funktion ist.

Einstellungen

Die **“textdatei”-Funktion** ist vollständig für sämtliche Interaktionen mit der Textdatei zuständig und bildet mit 200+ Aufrufen pro Programmdurchlauf die Basis für unser Programm. Da sich die Möglichkeit angeboten hat, hat Amos auch einige grafische Konfigurationen über die Textdatei definiert, die man so nicht ändern können soll, damit wir sie nach Fertigstellung einfacher anpassen können.
Die **“play”-Funktion**, die **“close”-Funktion** und die **“back_to_menu”-Funktion** (leztere zwei in “game_function.py”) ermöglichen es nach dem Schließen des Spiels die “tk_menü”-Funktion aufzurufen, obwohl diese von “game_function.py” aus nicht aufrufbar ist und das Menüfenster beim Spielen vollständig geschlossen ist.

Spiel

Das GUI ist zweigeteilt: Die Anzeige der eigenen Hand; diese Karten sind alle Buttons, damit sie zu legen sind und die sonstigen Anzeigen, Buttons und Stapel: Darunter befindet sich der Button, um das Fenster zu schließen und der Aufnahmestapel, ebenfalls ein Button. Außerdem die Labels, die die Kartenanzahl der Bots darstellen und der Ablagestapel, der aus Vierecken besteht, welche auf einer Leinwand dargestellt werden. Die Schrift wird über die Vierecke gelegt. Wichtig war, dass das Layout “responsive” ist, da wir das Spiel im Vollbildmodus programmiert haben und es sich auf verschiedene Bildschirmgrößen anpassen soll. Als nächstes wurde die Logik implementiert, also wann was gelegt werden kann, welches Element wohin verschoben werden muss und wann Gewinnbedingungen ausgelöst werden. Außerdem wurden die Listen angelegt, die die verschiedenen Stapel darstellen. Die Karten sind dabei einzelne Elemente, die verschoben werden. Dabei waren die weiteren Spieler im Hinterkopf, so dass dann am Ende die computergesteuerten Spieler hinzugefügt werden konnten. Am Ende ensteht ein Kreislauf, der Spieler kommt dran, dann jeweils die computergesteuerten Spieler. Dies wird dann immer wiederholt, bis eine Gewinnbedingung ausgelöst wird.

Schwierigkeiten und Probleme bei der Entwicklung

Situation, in denen wir gar nicht weiter wussten, gab es keine, allerdings hatten wir ein paar Ideen, bei deren Umsetzung Probleme entstanden, für dessen Lösung einige Zeit in Google investiert werden musste. Wir beide hatten zum Beispiel die Situation, in der wir nahezu identische Zeilen brauchten, diese aber individuell abrufbar sein mussten. Nachdem wir eine Weile zum Thema “Variablen automatisch erzeugen” googelten, mussten wir uns für einen Kompromiss einigen. Amos konnte von fünf auf drei mal zehn Zeilen reduzieren und die anderen in einer Schleife automatisch erstellen. Ansonsten haben die Funktionen von Tkinter und insbesondere die des Grid-Managers häufig nicht zuverlässig funktioniert, weshalb wir gerade am Anfang immer wieder Probleme mit dem Layout hatten. Aber mit der Zeit haben wir die Funktionen zu nutzen gelernt.
Hinzugekommen ist noch ein Speicher Problem, bei dem Thilo alle aktuellen Versionen seines Codes überschrieben hat. Wir konnten in den Logs von Visual Studio Code aber einen wenige Minuten alten Backup wiederherrstellen. Danach haben wir hierdrauf besonders geachtet.

Fazit

Das Projekt hat uns großen Spaß bereitet, auch wenn wir an manchen Zeitpunkten mit vielen Problemen zukämpfen hatten. Die Zusammenarbeit im Team und die sehr gut funktionierende Kommunikation zwischen uns hat gegenseitige Hilfe und damit eine reibungslose Projektdurchführung ermöglicht. Mit dem Ergebnis sind wir sehr zufrieden. Das Spiel funktioniert gut und wir konnten einige Features umsetzen. Außerdem hat uns das Bespielen des Produktes viel Spaß gemacht. Aber wir haben immer noch viele Ideen, die wir auch gerne noch umgesetzt hätten.

Erwähnungen/Quellenverzeichnis

Wir möchten uns bei unseren Mitschülerinnen und Mitschülern bedanken, für die hilfreichen Tipps zu Tkinter und außerdem den ausgiebigen Tests unseres Spiels.

Abb. Seite1 Tkinter
https://i0.wp.com/iot4beginners.com/wp-content/uploads/2020/04/65dc5834-de21-4e2e-bd4d-5e0c3c6994dd.jpg?fit=375%2C422&ssl=1

Der Code zum Spiel ist zu 100% selbst geschrieben. Inspiration und Hilfe bei Problemen gab es von folgenden Webseiten:
python-forum.de
stackoverflow.com
inf-schule.de
delftstack.com