

Aufgabe 1 (2-3 Personen)

Grafische Benutzeroberfläche zur manuellen Segmentierung in Schichtbild-Datensätzen

Manuelle Segmentierungen von medizinischen Bildern sind notwendige Voraussetzungen, um den Erfolg von (lernbasierten bzw. überwachten) Bildverarbeitungsalgorithmen zur automatisierten Auswertung von medizinischen Bilddaten zu ermöglichen. Vor allem zur Bereitstellung gelabelter Daten durch medizinische Experten sind solche Segmentierungsarbeiten von entscheidender Bedeutung, um damit z.B. Verfahren der künstlichen Intelligenz (welche Lerndaten erfordern) zu entwickeln und zu testen.

In dieser Praktikumsaufgabe sollen mit einer GUI manuelle Segmentierungen für medizinische MRT-Schichtdatensätze durch einen Anwender realisiert werden. Dabei soll der Anwender die Umrisse von Organen schichtweise eingeben können und diese Ergebnisse über „binäre Masken“ (Binärbilder) von den Organbereichen schichtweise abspeichern können. Der Anwender soll weiterhin schichtweise Korrekturen an den bereits segmentierten Organbereichen vornehmen bzw. Zwischenergebnisse verfeinern können.

Hinweis 1: für diese Aufgabe benötigen Sie keine medizinischen bzw. anatomischen Kenntnisse über abdominale Organe. Zumindest fließt das Unwissen über die korrekten Organbereiche nicht in die Benotung ein! Die Beispielbilder zeigen die Organbereiche nur zur Illustration.

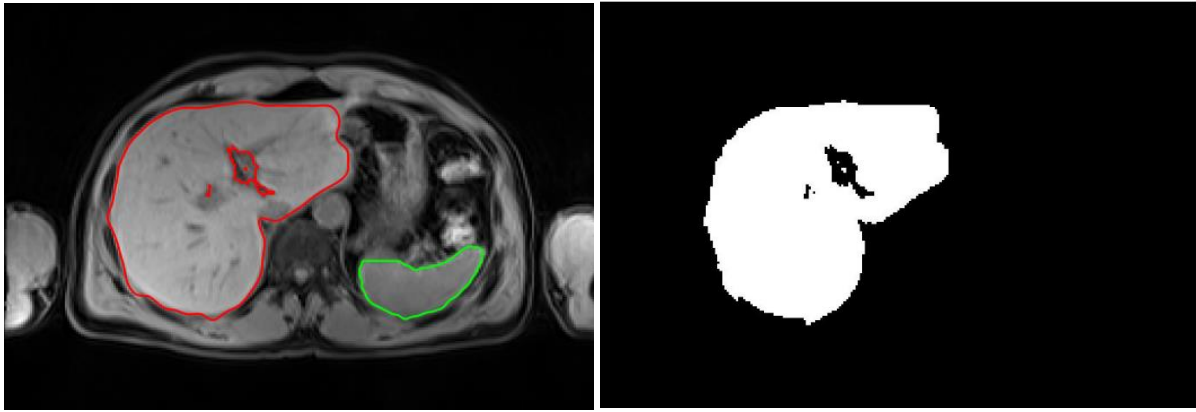
Hinweis 2: die Magnetresonanz-Bilder sind keine Originalbilder von Probanden, sondern wurden mit einem speziellen Entrauschungsverfahren geglättet. Somit wurde das Bildrauschen gegenüber dem Originalbild reduziert, was allerdings kleinere Bilddetails verwischen kann.

Die GUI soll folgende Funktionalitäten vorweisen und dabei möglichst fehlerfrei mit dem Nutzer interagieren:

- **Auswahl von Schichtbilddatensätzen** verschiedener Probanden, die jeweils in eigenen Ordnern abgespeichert sind
- **Einlesen aller Schichtbilder** eines Probanden: es existieren von jedem Probanden 64 Schichtbilder, die alle Grauwertbilder sind (keine Farbe!)
- Die **Schichtbilder** sollen **nacheinander** in einem Bereich der GUI **angezeigt** werden. Dabei soll der Benutzer die Möglichkeit haben, die Bilder **sowohl in aufsteigender als auch in absteigender Reihenfolge der Schichtnummern** durchzugehen.
- Die **Probandennummer** und die **aktuelle Schichtnummer** sollen in der GUI angezeigt werden.
- Der GUI-Nutzer soll eine **Kontur** auf dem angezeigten MR-Schichtbild **eingeben** können, damit er den Umriss eines Organs in der aktuell angezeigten Schicht manuell segmentieren kann.
- Die **Kontur soll geschlossen sein** und dabei das **Organsegment** in dieser Schicht **mit einer Farbkontur** darstellen.
- Das Organsegment kann Bereiche bzw. **Löcher** enthalten, die nicht zum Organ gehören.

- Der segmentierte Organbereich soll für jede Schicht in einem Binärbild abgespeichert werden. Das Binärbild entspricht einer **binären Maske**, die eine 1 an den Stellen des segmentierten Bereichs enthält und eine 0 für Bereiche, die nicht zum Organgewebe gehören (s.u.).
- Es sollte ein **Binärbild für jede Schicht und für jedes Organ** in speziellen Ordnern abgespeichert werden, der idealerweise als Name die entsprechende Probandennummer enthält.
- Die **Konturen** stellen somit die **Grenzen der 1er -Bereiche zu den 0er-Bereichen** in der binären Maske dar.
- Der Benutzer soll Möglichkeiten erhalten, Die **Organgrenzen zu verändern bzw. zu verfeinern**, indem **Organbereiche hinzugefügt und entfernt** werden können. Das Hinzufügen/Entfernen kann mit logischen Verknüpfungen auf den binären Masken erreicht werden:
 - o Beim Entfernen einer Region werden alle Pixel innerhalb der eingegebenen Region mit Nullen versehen und die Werte (0 oder 1) der anderen Bereiche in die Ergebnismaske übernommen.
 - o Beim Hinzufügen einer Region werden alle Pixel innerhalb der eingegebenen Region mit Einsen versehen und die Werte (0 oder 1) der anderen Bereiche in die Ergebnismaske übernommen.
- Der Anwender soll **4 verschiedene Organe** eingeben können, deren Organbereiche mit **unterschiedlichen Farben** angezeigt werden sollen.
- Die 4 Organe sind: **Leber, Milz, Galle, Nieren** (linke und rechte)
- Die Daten von 4 Probanden liegen in vier unterschiedlichen Magnetresonanz-Gewichtungen vor. Das Rauschen in den Bilddaten wurde mit speziellen Bildverarbeitungsverfahren reduziert (entrauscht). Die Bilddaten der 4 Gewichtungen liegen ordnerweise vor (Gewichtung_42..45).
- Der Anwender soll sich die Bilder in den **4 verschiedenen Gewichtungen anzeigen lassen können**, wobei er auch in jeder Schicht zwischen den **4 verschiedenen Gewichtungen beliebig wechseln** können soll.
- In **jeder Schicht soll für jedes Organ** die manuell **segmentierte Fläche** dem Benutzer übersichtlich angezeigt werden. Jedes Bildpixel hat eine reale Abmessung von 1.64 * 1.64 mm (DIXON-Sequenz gängiger SIEMENS MRT-Tomographen). Falls in einer Schicht kein Organbereich segmentiert wurde, soll für dieses Organ keine Fläche angezeigt werden.
- Der Benutzer soll die Möglichkeit erhalten, die manuell segmentierten Flächen **für jedes Organ und für jede Schicht in einer Datei** abzuspeichern. Das Speicherformat ist von den Programmierern frei wählbar. Beim Laden der gespeicherten Segmente für einen Probanden sollen auch die segmentierten Organflächen aus der Datei wieder eingelesen werden und in der GUI angezeigt werden.

Links: Beispielbild für die farblich unterschiedlichen Konturen 2 verschiedener Organe (Leber: rot, Milz: grün). Rechts: Beispiel für eine binäre Maske (hier nur Lebergewebe)



Für diese Aufgabe ist das Aneignen von Programmierwissen (bzgl. Python) aus den folgenden Links hilfreich:

findcontours:

<https://www.delftstack.com/de/howto/python/opencv-find-contours/>

<https://www.programcreek.com/python/example/70440/cv2.findContours>

<https://medium.com/analytics-vidhya/opencv-findcontours-detailed-guide-692ee19eeb18>

ginput:

<https://www.geeksforgeeks.org/matplotlib-pyplot-ginput-in-python/>

Finden von Punkten innerhalb von Polygonen:

`pip install Shapely`

https://autogis-site.readthedocs.io/en/latest/notebooks/L3/02_point-in-polygon.html

<https://automating-gis-processes.github.io/CSC18/lessons/L4/point-in-polygon.html>

Aufgabe 2 (2-3 Personen)

Grafische Benutzeroberfläche zur Auswahl von Schiffsreisen

Es soll eine grafische Benutzeroberfläche zur Abfrage und Auswahl bestehender Schiffsreisen erstellt werden, die im Anschluss für eine personenbezogene Buchung weiter genutzt werden können. Aktuell existieren 50 unterschiedliche Schiffsreisen, die in einer Excel-Datei (Schiffsreisen.xlsx) mit ihren wesentlichen Daten zusammengefasst sind. Die GUI soll die einzelnen Daten aus der Excel-Datei selektiv nach den Bedürfnissen des Anwenders abrufen können. Die GUI soll diejenigen Reisen heraussuchen, die den Abfragekombinationen des Anwenders (=Auswahlkategorien) entsprechen: Z.B. alle Schiffsreisen auf der **Ostsee**, mit mindestens **8 Übernachtungen**, wobei die Städte „**Stockholm, Helsinki, etc. ...**“ besucht werden und die einen bestimmten Schiffstyp (**A-F**) haben. Die Auswahl soll in einer Ergebnisliste für jede Abfragekombination auf der GUI angezeigt werden. Dabei sollen die wesentlichen Informationen in der Liste dargestellt werden. Der Benutzer kann zu den einzelnen Kategorien Auswahlen treffen, allerdings sich auch dafür entscheiden, keine weiteren Angaben oder Bedingungen zu einzelnen Auswahlkategorien machen zu müssen. Falls der Anwender keine Angaben zu einer speziellen Auswahlkategorie vorgibt, sollen alle restlichen Reisen, die von der Abfragekombination nicht ausgeschlossen wurden auch komplett angezeigt werden. Der Anwender soll folgende Abfragen einstellen können:

- Der **Ort des Meeres** („Meerart“: Ostsee, Nordsee, Mittelmeer, Nordpolarmeer, Nordpolarmeer mit Spezialflugtraining und Sonderbewachung) soll als oberstes Auswahlkriterium zunächst auswählbar sein.
- Die **Anzahl an Übernachtungen** +/-2 Übernachtungen. Wenn der Anwender 5 Übernachtungen auswählt, dann sollen Reisen mit 3-7 Übernachtungen angezeigt werden. Bei einer Nutzereingabe von bis zu 2 Übernachtungen werden negative Übernachtungszahlen nicht weiter berücksichtigt.
- Der Nutzer soll **keine, eine oder mehrere Städte**, die er als Schiffsreisender besuchen möchte, **auswählen** können. **Bei Bedarf soll der Anwender ein Bild für eine Stadt, die er auswählt, separat angezeigt bekommen.**
- Dabei ist zu beachten, dass **belegte Auswahlkategorien die Auswahl der anderen Auswahlkategorien limitieren** können. Z.B.: wenn ein spezielles Meer wie die Nordsee ausgewählt wurde, dann sollen keine Städte aus dem Mittelmeer mehr auswählbar sein.
- Der Schiffstyp (A-I, X) soll auswählbar sein, der für die aktuell ausgewählten Reiseparameter verfügbar ist. Dabei soll – bei Bedarf- der Schiffstyp als Bild angezeigt werden können.
- Für die meisten Auswahlkategorien sind Bilder hinterlegt, die auf Verlangen des GUI-Nutzers angezeigt werden können. Die Bilder sollen dabei möglichst in einheitlicher Größe angezeigt werden (**Googlen Sie am besten nach geeigneten `resize()`-Methoden für Bilder in Python**).
- **Bilder sollen dem Nutzer für folgende Auswahlkategorien angezeigt werden:**
 - o **Besuchte Städte:** es können mehrere Bilder von den besuchten Städten ausgewählt und entsprechend angezeigt werden.

- **Der Schiffstyp:** es sind Beispielbilder für die Schiffstypen A-I und X in den Daten bereitgestellt.
- **Die Kabinenart:** es sind Bilder für die unterschiedliche 6 Kabinenarten hinterlegt.

Nutzerkapital verwalten

In der GUI soll es eine Möglichkeit geben, neue Nutzer mit Vor- und Zunamen einzugeben. Ein Nutzer soll sich mit einem Passwort, welches zu Beginn abgefragt wird, am Reiseportal einloggen können. Der Name des eingeloggten Nutzers soll auf der GUI entsprechend zu sehen sein. Es kann immer nur ein Nutzer eingeloggt sein. Ein neu angelegter Nutzer hat zu Beginn kein Kapital (Kontostand = 0 €) zum Kauf einer Reise. Bei der Erstanmeldung sowie **bei jeder weiteren Anmeldung eines Nutzers** im Reiseportal (in der GUI) soll ein **Zufallsgenerator im ganzzahligen Zahlenbereich zwischen 1000 und 3000 € das aktuelle Nutzerkapital erhöhen**. Somit kann ein Nutzer sein Kapital bei jeder neuen Anmeldung immer weiter erhöhen. Das Nutzerkapital soll jedoch **maximal 20.000 €** betragen dürfen.

Für jeden Nutzer sollen folgende Informationen abgespeichert werden:

Nutzername, Passwort sowie **aktueller Kontostand**. Diese Informationen können wahlweise in einer Datei enthalten sein oder es kann für jeden Nutzer eine Datei zum Verwalten der Informationen angelegt werden.

Der Kontostand des aktuell angemeldeten Nutzers soll ebenfalls auf der GUI angezeigt werden.

Weitere GUI-Funktionalitäten

Die vom Nutzer **vorausgewählten Reisen** sollen ähnlich wie in der Excel-Datei **in einer Art Liste erscheinen**. Dabei soll die Reisennummer, Meerart, Anzahl Übernachtungen, etc....der Reihe nach angezeigt werden. Falls mehrere Reisen auf die Nutzerauswahl zutreffen, sollen mehrere Reisen untereinander erscheinen. Falls alle Kabinenpreise einer Reise unter dem Kontostand des Nutzers liegen, sollen diese nicht weiter anwählbar sein und ein Hinweis in der GUI erscheinen, dass diese Reisen aufgrund von zu niedrigem Kontostand nicht weiter auswählbar sind. Der Nutzer soll sich dann für eine Reise (bei der er sich zumindest eine Kabinenart finanziell leisten kann) aus der Liste entscheiden können. Danach sollen die anderen Reisen aus der Liste verschwinden.

Wenn der Nutzer sich für eine Reise entschieden hat, soll er auch hier bei einem Klick auf eine der 6 angezeigten Preiskategorien für die Kabinenarten ein **Bild von der Kabine angezeigt** bekommen. **Da die ausgewählte Kabinenart den Kaufpreis der Reise bestimmt**, muss der Nutzer eine Kabinenart für die ausgewiesene Reise mit einer **Preiskategorie** auswählen können. Dabei soll der Kontostand des Benutzers berücksichtigt werden: alle vorausgewählten Reisen (inklusive der zur Verfügung stehenden Kabinenpreise) sollen dem Benutzer zwar angezeigt werden, jedoch sollen die Reisen (inkl. Kabinenart), deren Preise über dem Kontostand des Benutzers liegen **ausgegraut bzw. nicht weiter auswählbar** sein. Dabei soll ein **Hinweis in der GUI erscheinen**, dass für diese Kabinenpreise der **Kontostand des Nutzers nicht ausreicht**.

Nachdem die Kabinenart vom Nutzer festgelegt wurde (die er sich laut seinem Kontostand auch leisten kann), **hat er sich für eine Reise entschieden**. Nun soll er noch die Möglichkeit bekommen sich **für eine Reisezeit zu entscheiden**. Gehen Sie dabei der Einfachheit halber von folgenden Bedingungen der Reisezeitauswahl aus:

- Alle Reisen können nur im Zeitraum vom Mai 2025 bis Oktober 2025 durchgeführt werden. Dabei muss nicht nur der Starttermin, sondern auch der Endtermin der Reise innerhalb dieses Halbjahres liegen.
- Um die Zeitplanung einfach zu halten, sollen nur für die Abfahrtstage der Schiffstypen folgende Bedingungen gelten:
 - Schiffstyp A: fährt nur Montags ab
 - Schiffstyp B: fährt nur Dienstags ab
 - Schiffstyp C: fährt nur Mittwochs ab
 - Schiffstyp D: fährt nur Donnerstags ab
 - Schiffstyp E: fährt nur Freitags ab
 - Schiffstyp F: fährt nur Samstags ab
 - Die Schiffstypen G, H und I: fahren nur Sonntags ab
 - Die Schiffstypen J und K: fahren nur am ersten Montag und Mittwoch eines Monats ab
 - Schiffstyp X: fährt nur am 1.Tag eines Monats ab

Der Nutzer soll sich (z.B. in einer Kalenderansicht oder einer anders gearteten Ansicht, die Sie als geeignet halten) für einen der möglichen Starttermine entscheiden können. Idealerweise wird dem Nutzer angezeigt, welche Starttermine für seine Auswahlkombinationen möglich sind (beachten Sie, dass der Rückreisezeitpunkt maximal am 31. Oktober 2025 sein darf). Falls dem Nutzer keine dieser Daten passen sollten, soll er sich entscheiden können, die Reiseplanung abzubrechen und wieder in den Startbereich der Reiseauswahl zurückzuspringen.

Nach der erfolgreichen Festlegung der Reisedaten soll dem Nutzer seine Reise als „gekauft Produkt“ angezeigt werden. Dabei sollen dem Nutzer **alle besuchten Städte der Reise, der Schiffstyp, die Kabinenart sowie die Reisedaten gemeinsam mit dem zu zahlenden Endbetrag entweder in einem Bereich innerhalb der GUI oder in einem Extra-Fenster** (z.B. mit dem Titel: „**gekauft Produkt**“) **angezeigt werden**. Die Bilder sollten übersichtlich angeordnet sein und die Aufteilung sich nach der Anzahl der anzuzeigenden Bilder orientieren. Diese finale Anzeige dient als Darstellung des gekauften Produkts durch den Nutzer.

Beim GUI-Bereich bzw. im Extra-Fenster („gekauft Produkt“) soll mit einem entsprechenden GUI-Steuerelement einstellbar sein, **ob die besuchten Städte auf einmal erscheinen oder nacheinander einzeln angezeigt werden sollen**. Wenn die besuchten Städte auf einmal erscheinen, dann sollen sie entsprechend ihrer Anzahl neben- bzw. untereinander

übersichtlich angezeigt werden. Falls die besuchten Städte nacheinander erscheinen sollen, sollen sie einzeln angezeigt werden und mit entsprechenden GUI-Steuerelementen (z.B. mit Vorwärts- und Rückwärts-Buttons) durchiteriert werden können.

Zum Schluss soll dem **Nutzer ein Dialog** angezeigt werden, in dem sein Name (Vor- und Zuname), die Reisennummer, die Kabinenart und der Reisepreis angezeigt wird. Der Nutzer soll in diesem Kaufdialog seine Adresse, Handynummer und seine **Bankdaten eingeben können**. Diese Daten sollen (quasi als Buchung für die ausgewählte Reise) in einer **Textdatei abgespeichert** werden. Nachdem die Reise vom Benutzer gekauft wurde, soll sein Kontostand um den Kaufbetrag erniedrigt werden, was in der entsprechenden Datei abgespeichert werden soll.

Beispielauswahl besuchte Städte

Bergen



Göteborg



Trondheim



Tromsø



Beispielauswahl Schiffstyp (links) und Kabinenart (rechts)



Aufgabe 3 (2-3 Personen)

Grafische Benutzeroberfläche für einen Wochenbehandlungsplan für Zahnärzte und deren Patienten

Es soll eine GUI zur Zusammenstellung von zahnärztlichen Behandlungszeiten und -kosten für jeden Patienten konzipiert werden. Die Zusammenstellung soll auf Zeitbasis der nächsten 3 Monate durchgeführt werden. Dabei soll der **Patient Möglichkeiten erhalten**, um u.a. die **Anzahl seiner Behandlungen selbst festzulegen**, so dass er weitere Behandlungen z.B. auf spätere Wochen zu verlegen. Es soll ein detaillierter Terminkalender erstellt werden, mit dem die Terminplanung für die nächsten 3 Monate realisiert werden kann. Der Terminkalender soll die Tageszeiten von 8-18 Uhr für jeden Tag abdecken und jede volle Stunde anzeigen.

Für diese grafische Benutzeroberfläche gibt es **2 verschiedene Rollen: Zahnarzt und Patient**, wobei sich jeder unter **seinem initialen Passwort (wie in der Excel-Datei gegeben) anmelden** kann und somit die Perspektive aus Zahnarztsicht bzw. Patientensicht auf die GUI automatisch eingestellt wird:

- Ein Patient erhält eine Übersicht über seine anstehenden Behandlungen und kann Anzahl der zu behandelnden Zähne und die Wertigkeit des Füllmaterials auswählen.
- Ein Zahnarzt kann sich die für jeden Wochentag eingetragenen Patienten, die er behandeln soll, anzeigen lassen.

In der Datei Patienten_Zahnärzte_Kosten.xlsx sind die aktuellen Stammpatienten (Untertabelle „Stamm-Patienten“) sowie die Zahnärzte der Zahnarztpraxis (Untertabelle „Zahnärzte“) abgelegt. Zudem sind dort die **Kosten und die Dauer von standardmäßigen Zahnbehandlungen** (Untertabelle „Kosten und Behandlungsdauer“) hinterlegt.

Jeder Patient und jeder Zahnarzt haben eine **eigenes Passwort**, das zu Beginn deren initiales Passwort ist. Idealerweise werden diese Daten aus der Excel-Datei eingelesen, können aber für die ersten Entwicklungsschritte per copy&paste in den Programmtext übernommen werden.

Rolle: Patient

Funktionalitäten/Programmabläufe als Patient: In der GUI soll es eine Zugangsmöglichkeit für jeden Patienten geben.

- Ein Patient meldet sich mit **seinem Passwort** am System an.
- Es soll eine Möglichkeit geben, dass der Patient ein **neues Passwort eingibt** und dieses im System hinterlegt wird. Der Benutzer kann, muss aber sein Passwort nicht ändern.
- Der aktuelle **Patient soll eine Kosten- und Zeitanzeige erhalten**, in der sowohl die **Kosten für seine Behandlungen als auch die Behandlungszeiten summiert** werden, für die er (bis jetzt) noch keinen Termin bei einem Zahnarzt gebucht hat. Dabei soll der Patient die Möglichkeit bekommen:

- a. die **Anzahl der zu operierenden Zähne selbst einzustellen** (um z.B. die restlichen Behandlungen in einem Folgetermin durchzuführen) und
- b. sich zwischen den 3 Möglichkeiten für ein **Füllmaterial** (normal, höherwertig, höchstwertig) zu entscheiden.

Je nach eingestellter Anzahl der Zähne, dem Füllmaterial und der Krankenkasse soll dem Patienten ein Endbetrag angezeigt werden. Die anteilmäßige Kostenübernahme der gesetzlichen und privaten Krankenkassen sind in der Excel-Datei für jede Behandlungsart und Wertigkeit des Füllmaterials aufgeführt.

- Die **Auswahlmöglichkeiten von Zahnanzahl und Füllmaterial** sollen für den aktuell angemeldeten Patienten in der GUI sichtbar sein und seine Auswahl mit einem **Bestätigungsbbutton** abgeschlossen werden können.
- Der Patient soll sich entscheiden können, ob er im Falle mehrerer dentaler Problematiken (z.B. mehrere betroffene Zähne), die Behandlungen auf mehrere Termine aufteilen möchte.
- Der/die Patient(in) soll eine Auswahl an Zahnärzten bekommen, die ihn/sie (bzgl. der Krankenkassenart) behandeln können. Der/die Patient/in soll sich im Falle von mehreren angezeigten Zahnärzten für einen Zahnarzt entscheiden können. Mit einer Bestätigung soll danach in die konkrete zeitliche Buchung in den Terminkalender des ausgewählten Zahnarztes übergegangen werden.
- Im angezeigten Terminkalender des ausgewählten Zahnarztes sollen nur die freien Termine auswählbar sein. Die bereits belegten Termine des Zahnarztes sollen ebenfalls (z.B. anders farblich markiert) im Terminkalender dargestellt werden.
- Der Patient soll sich einen Termin auswählen, den er dann offiziell für die Terminbuchung bestätigt. Die dentale Problematik bzw. die Anzahl der Zähne, die der Patient bei der Terminbuchung festgelegt hat, soll dann aus dem Kontingent seiner dentalen Problematiken entfernt werden (z.B. in einer Datei), so dass diese bereits gebuchte Problematik bei seinen dentalen Problematiken (z.B. für eine erneute bzw. doppelte Terminbuchung) nicht mehr auftaucht.
- Zum Beispiel: Frau Meyer hat 2 Zähne, die eine kleine Karies aufweisen. Sie wählt aber nur einen Zahn aus, den sie behandeln lassen möchte. Sie wählt als gesetzlich Versicherte das höchstwertigste Füllmaterial (250€) für diesen Zahn aus. Somit muss Frau Meyer $1 \times 250\text{€} \times 0.5$ (Anteil als gesetzlich Versicherte) = 125€ bezahlen. Diese Behandlungskosten sollen Frau Meyer in der GUI angezeigt werden. Frau Meyer wird im nächsten Schritt angezeigt, dass nur Herr Dr. Kraft und Herr Dr. Hausmann sie behandeln können. Sie entscheidet sich für Herrn Dr. Kraft. Anschließend klappt der Terminkalender von Herrn Dr. Kraft auf und sie kann sich für die nächsten 3 Monate einen Termin bei ihm (zu dem Herr Dr. Kraft auch wirklich behandelt:

Behandlungszeiten beachten) buchen. Damit ist der gebuchte Termin konkret bei Herrn Dr. Kraft besetzt und es dürfen keine anderen Patienten zu diesem Termin eine Behandlung bei Herrn Dr. Kraft mehr buchen können. Ein Zahn mit kleiner Karies wird aus der dentalen Problematik von Frau Meyer entfernt, so dass sie bei einer erneuten Buchung nur noch den 2.Zahn (noch nicht eingeplanten Zahn) für eine Terminplanung einsetzen kann.

- Für jeden eingeloggten Patienten soll zudem eine Terminübersicht auswählbar sein, die ihm Datum und Uhrzeit aller seiner Behandlungen innerhalb der kommenden 3 Monate anzeigt.
- **NEUE PATIENTEN:** Es soll eine **Eingabemöglichkeit für neue Patienten** in der GUI geben: dabei sollen Patientennamen, initiales Passwort, Krankenkassenart, dentale Problematik (aus der Menge wie in der Excel-Tabelle dargestellt: Karies klein/groß, Teilkrone, Krone, Wurzelbehandlung sowie die Anzahl der betroffenen Zähne) eingegeben werden können. Es sollen auch mehrere dentale Problematiken eingegeben werden können. (Hinweis: auch wenn es für den praktischen Ablauf weniger realistisch erscheint, gehen Sie der Einfachheit davon aus, dass ein neuer Patient seine dentale Problematik schon kennt.) Diese Informationen sollen entsprechend in Dateien gespeichert und verwaltet werden.
- Falls sich ein/e neue(r) Patient(in) einloggt, der/die denselben Namen haben, wie ein bereits in den Daten enthaltener Patient, soll der Nutzer gefragt werden, ob er sich schon vorher angemeldet hat. In diesem Fall, soll sein Passwort, um das neu angegebene Passwort ersetzt werden und die neuen dentalen Problematiken aufgenommen werden. Falls sich ein neuer Patient anmeldet, der zufällig den Namen eines bereits in den Daten enthaltenen Patienten hat, soll dieser unter einer zahlenmäßigen Erweiterung im Nachnamen (z.B. Frau Meyer_2) im System angelegt werden. Die Zahl in der Namens Erweiterung soll um eins höher sein wie die Anzahl der Nachnamen, die im System gefunden wurden (Wenn es bereits 4 verschiedene Frau Meyer gibt, soll die neue Patientin den Namen Frau Meyer_5 erhalten).

Rolle: Zahnarzt

Wenn sich ein Zahnarzt in die GUI einloggt, dann:

- soll er ebenfalls die Möglichkeit haben, sein **Passwort zu ändern**.
- sich seinen Terminkalender für die kommenden 3 Monate ansehen können. Dabei sollen die Patientennamen mit Ihren Behandlungen und der gebuchten Zeit für den Zahnarzt in der GUI visualisiert werden.
- **Anlegen neuer Zahnärzte:** ähnlich wie beim Anlegen neuer Patienten, sollen sich neue Zahnärzte mit Namen und initialen Passwort am System anmelden können. Zudem soll hier auch die Krankenkassenart eingegeben werden können, die ein neuer Zahnarzt behandeln möchte. Es muss auch eine Möglichkeit gefunden werden, wie man die wöchentlichen Behandlungszeiten eines neuen Zahnarztes eingibt.

Aufgabe 4 (2-4 Personen)

Grafische Benutzeroberfläche für den Onlinekauf von landwirtschaftlichen Fahrzeugen und mobilen Nutzgeräten

Es sollen Käufe von Traktoren mit einer grafischen Benutzeroberfläche durchgeführt werden. Die zugrundeliegenden Informationen stehen in der Excel-Datei „mobile Arbeitsmaschinen Landwirtschaft.xlsx“. Ein Nutzer hat ein bestimmtes Budget zur Verfügung und soll anhand von Preisbereich, Km/h-Bereich, PS-Bereich und Baujahr-Bereich Traktoren auswählen können. Der Online-Verkäufer (der hier der Einfachheit halber den Namen „Klaus“ haben soll) kann die Traktoren, deren Lagerbestand auf 0 geschrumpft ist von den einzelnen Herstellerfirmen (z.B. John Deere, Claas, Kubota, etc...) nachbestellen. Nutzer und der online-Verkäufer sollen sich mit ihren Namen und Passwörtern in der GUI anmelden können, dazu sollen entsprechende Dialoge für die Passwordeingabe in der GUI realisiert werden. Vergeben Sie dazu **initiale Passwörter für jeden Nutzer**. Die Passwörter können der Einfachheit halber in einer Textdatei abgelegt werden.

Es sollen **zunächst 5 Nutzer** und der online-Verkäufer angelegt werden. Jeder Nutzer (jede Nutzerin) hat einen Namen und ein Startbudget:

Nutzer 1: Oskar, Budget: 270.000 €

Nutzer 2: Benni, Budget: 250.000 €

Nutzerin 3: Daniela: Budget: 200.000 €

Nutzer 4: Horst: Budget: 550.000 €

Nutzerin 5 Sieglinde: Budget: 800.000 €

Die Nutzer sollen sich unter ihrem Namen einloggen können und ihr Name und Budget sollen in der GUI angezeigt werden. **Es soll auch ein Dialog für die Aufnahme neuer Nutzer in die GUI integriert werden, bei dem Namen, Budget und initiales Passwort abgefragt werden.**

Zudem soll der Online-Verkäufer „Klaus“ sich mit seinem Namen und einem Passwort einloggen können. „Klaus“ hat ein Startbudget von 3.000.000 €.

Ein Nutzer muss sich aus dem Online-Handel ausloggen (z.B. durch einen Ausloggen-Button), bevor ein anderer Nutzer oder der online-Verkäufer sich anmelden kann.

Ein Nutzer erhält die **Perspektive des Käufers** und kann seine Traktorensuche nach **4 verschiedenen Kriterien** regeln:

- **PS, Km/h, Preisbereich und Baujahr.** Für diese Suchkriterien sollen **geeignete Eingabemöglichkeiten** mit geeigneten Auswahlbereichen (z.B. Preisbereich in 10.000€ Abstufungen, km/h-Bereich +/- 5, ...) zur Verfügung gestellt werden.
- Es soll eine Liste von ausgewählten Traktoren angezeigt werden, wobei der erste Eintrag ein kleines Bild des Traktors ist. Die anderen Kennzahlen (PS, km/h, Baujahr) sollen folgen sowie der Preis deutlich angezeigt werden.

- **Das angezeigte Bild soll bei Bedarf größer werden und der Nutzer soll in das Bild hineinzoomen können, um sich das Produkt genau ansehen zu können.**
- Der eingeloggte Nutzer soll einen der **Traktoren aus der Liste zum Kauf auswählen** können. Der ausgewählte Traktor soll dann in einen **Warenkorb** abgelegt werden können.
- Abhängig von der Firma des ausgewählten Traktors soll der Nutzer sich **kompatible Zusatzgeräte anzeigen lassen** können. Aus der Liste der kompatiblen Zusatzgeräte soll der Nutzer sich wiederum einen oder mehrere Zusatzgeräte zum Kauf aussuchen können.
- Auch bei den **Zusatzgeräten sollen kleine Bilder** am Anfang der Liste angezeigt werden.
- Nachdem der Nutzer seine Auswahl getroffen hat und quasi „zum Kauf“ geht, soll der **Endpreis aller Warenkorbprodukte** (ausgewählte Traktoren und deren Zusatzprodukte) **angezeigt** werden.
- **Überschreitet der errechnete Endpreis das Budget des Nutzers** soll der **Kauf nicht möglich** werden und der Nutzer auf sein zu geringes Budget für den gewählten Kauf hingewiesen werden.
- Kann sich der Nutzer die Produkte im Warenkorb leisten, wird **der Endpreis von seinem Budget abgezogen** und dieses **aktualisiert**. Zudem werden die **Anzahlen der eingekauften Produkte im Lager des Onlineanbieters entsprechend reduziert**.

Perspektive des Online-Verkäufers

Der Online-Verkäufer kann, nachdem er sich eingeloggt hat, ebenfalls Traktoren und Zusatzgeräte, der im Bestand aufgeführten Marken einkaufen. Dabei muss er die **Anzahl des Zukaufs** bestimmen können und die eingekauften Produkte werden um ihre neue Anzahl im **Lager aktualisiert**. Auch der Verkäufer verfügt über ein **Budget**, das bei einem Verkauf an einen Nutzer erhöht bzw. einem Neukauf zur Lagerauffüllung reduziert wird. Gehen Sie der Einfachheit davon aus, dass der **Verkäufer einen Einkaufspreis von 65% des Verkaufspreises hat**. Wenn der Verkäufer (Klaus) kein Budget mehr hat, muss er warten bis wieder Nutzer bei ihm Waren gekauft haben, so dass er wieder Geld für neue Einkäufe bei den Herstellern hat.

Zeitliche Wertminderung der Traktoren und Zusatzgeräte

Gehen Sie davon aus, dass das Baujahr der Traktoren bereits im Kaufpreis (bei einem Kauf im online-Handel von Klaus) enthalten ist. Die Zusatzgeräte werden immer als Neuware verkauft. Zusätzlich sollen Sie einen zeitbasierten Wertminderungsprozess für alle Geräte (Traktoren und Zusatzgeräte) simulieren. Sowohl der Kaufpreis als auch der zeitabhängige geminderte Wert eines Gerätes (**aktueller Gebrauchswert**) sollte für jeden Nutzer mitabgespeichert und - falls dieser eingeloggt ist- auch angezeigt werden.

Die geschätzte Wertminderung soll aufgrund der Zeitdifferenz zwischen Kauf und aktuellen Zeitpunkt für jedes erworbene Gerät eines Nutzers simuliert werden. Sie können die Rahmenbedingungen für die Zeitdifferenz z.B. in Abhängigkeit von der Häufigkeit des

Einloggens der Nutzer festlegen: bei den hohen Kaufpreisen und damit verbundenen selteneren Käufen könnte man z.B. der Einfachheit halber davon ausgehen, dass nach jedem Einloggen wieder ein halbes Jahr vergangen ist. Sie können den zeitlichen Vorgang auch mit anderen Zeitdifferenzen testen (z.B. 0,25 Jahre oder 0,75 Jahre vergehen bis zu einem nächsten Login). Sie können sich auch eine eigene Idee zur zeitabhängigen Simulation der Wertminderung überlegen und z.B. eine fiktive Uhr im Hintergrund mitlaufen lassen, die die Echtzeit in eine fiktive Zeit umrechnet, um solch einen Prozess für einen kurzen Vortrag von 10-20 Minuten zu simulieren. (In diesem Punkt haben Sie also eine gewisse Gestaltungsfreiheit!)

Im Tabellenblatt „Wertminderungsraten“ der Excel-Tabelle sind jährliche Raten zur Wertminderung angegeben, die Sie zur Bestimmung der aktuellen Gerätewerte verwenden sollten. Für die Traktoren sind diese Wertminderungsraten herstellerspezifisch. Für die Zusatzgeräte sind die Wertminderungsraten (der Einfachheit halber) herstellerspezifisch und besitzen somit alle dieselbe Wertminderungsrate.

Verkauf von Gebrauchtgerten durch den privaten Nutzer

Es soll dem privaten Nutzer zusätzlich ermöglicht werden, dass er seine erworbenen agrartechnischen Geräte nach einiger Zeit auch wieder auf einem Gebrauchtmrkt verkaufen kann. Dazu könnte er sich theoretisch selber um Kontakte oder Kaufinteressenten bemühen, jedoch soll der private Nutzer in dieser Aufgabe seine Geräte über die GUI des online-Handels („Klaus“) verkaufen können.

Klaus hat schon aus seinem Agrartechnik-Studium gute und vertrauensvolle Kontakte zu Kommilitonen/innen aufgebaut, die heute potentielle Käufer von gebrauchten agrartechnischen Geräten sind. Zudem hat er gute Netzwerke und Kontakte aus seiner Branche während seiner Arbeitszeit aufgebaut. Klaus „verkauft“ diese in vielen Jahren aufgebauten Vertrauenskontakte über seine Webseite (die GUI), da er diese potentiellen Gebrauchtgertekäufer sowie ihre Neigungen gut kennt und so auch Vertrauen zwischen privaten Nutzer und Gebrauchtgertekäufer schafft. Aus diesem Grunde soll in der GUI auch der Verkauf von gebrauchten Geräten des privaten Nutzers realisiert werden, wobei Klaus für seine Mühe und „Vertrauensgarantie“ bei jedem Verkauf von gebrauchten Geräten eines privaten Nutzers eine Provision von 1 % vom Verkaufswert erhält, die seinem Budget gutgeschrieben werden soll. (Klaus selber kauft keine gebrauchten Geräte!)

Realisieren Sie in der GUI eine Möglichkeit für so einen Verkauf von gebrauchten Geräten, der nur stattfinden kann, wenn ein privater Nutzer auch in der GUI eingeloggt ist. Der eingeloggte Nutzer kann nur etwas verkaufen bzw. Verkaufsangebote erteilen, wenn er über mindestens ein agrartechnisches Gerät verfügt.

Klaus Vertrauensnetzwerk besteht aus 8 Personen, die ebenfalls nur ein Start-Budget für den Gebrauchtgertekauf besitzen. Diese Personen kommen teilweise auch aus anderen europäischen Ländern (wobei Sie mögliche Währungsunterschiede zu nicht EU-Ländern vernachlässigen können) und sind:

Alejandro: 160.000 €

Francois: 400.000 €

Chris: 90.000 €

Francesco: 190.000 €

Mette: 275.000 €

Birgit: 700.000 €

Lukas: 110.000 €

Sandra: 75.000 €

Nach erfolgten Gebrauchtgeräteverkäufen werden die Budgets dieser Gebrauchtgeräte Käufer entsprechend erniedrigt und die der privaten Nutzer (die in dieser Rolle als Privatverkäufer auftreten) entsprechend erhöht, was auch korrekt in der GUI aktualisiert und angezeigt werden soll.

Der Verkauf von Gebrauchtgeräten sollte zumindest prinzipiell folgendermaßen ablaufen:

- Der eingeloggte private Nutzer hat durch ein GUI-Steuerelement Zugang zu einer Verkaufsplattform für den Gebrauchtgeräteverkauf und bekommt dadurch eine Übersicht über die 8 Gebrauchtgeräte Käufer (Name sowie – zu leichteren Kontrollzwecken- auch deren Budgets) aus Klaus Netzwerk. Er kann ein Gerät aus seinem Besitz auswählen und dieses auf eine Verkaufsplattform stellen.
- Alle 8 Gebrauchtgeräte Käufer können nun ein Angebot abgeben. Bestimmen Sie für jeden Gebrauchtgeräte Käufer 2 zufällige Zahlen:
 - die 1. Zufallszahl aus dem Intervall $[0;1]$ gibt an, ob der Gebrauchtgeräte Käufer überhaupt an dem Gerät interessiert ist => ist die 1. Zufallszahl < 0.6 , dann ist dieser nicht an dem aktuell auf der Verkaufsplattform veröffentlichten Agrargerät interessiert und gibt kein Angebot ab.
 - Die 2. Zufallszahl ist für das Zustandekommen des Kaufpreises verantwortlich: da sich die 8 Gebrauchtgeräte Käufer gut auskennen, schlagen Sie immer Kaufangebote im Bereich von -30% bis +10% des **aktuellen Gebrauchswertes** vor. Verwenden Sie die 2. Zufallszahl, um ein Kaufangebot aus diesem Bereich auszuwählen.
- Alle zufällig ermittelte Angebote sollen angezeigt werden. Falls ein Gebrauchtgeräte Käufer kein Angebot abgegeben hat, soll dies ebenfalls angezeigt werden. Falls ein Gebrauchtgeräte Käufer einen **Höchstwert** geboten hat, soll dieser **automatisch ausgewählt** werden und der private Nutzer kann sich (z.B. durch Klick auf einen Verkaufsbutton) für den Verkauf entscheiden. Die Budgets bzw. Gerätebestände des privaten Nutzers und des Gebrauchtgeräte Käufer (mit dem höchsten Angebot) werden automatisch aktualisiert und angezeigt.

Auffüllung der Budgets

Mit der simulierten Zeit in Ihrer GUI sollen auch die Budgets der privaten Nutzer und der Gebrauchtgerätekäufer aufgefüllt werden können. Diese sollen nach dem Zufallsprinzip um einen Betrag zwischen 0 € und 50.000 € pro Jahr ansteigen können. Benutzen Sie dazu eine vom Computer generierte weitere Zufallszahl (hier wäre das die 3.Zufallszahl) um diesen Anstiegswert zu bestimmen. Das Budget von Klaus soll allerdings nicht jährlich mit einem zufälligen Wert ansteigen, da sein Budget sich nur nach seinen Käufen und Verkäufen bzw. Verkaufsprovisionen orientieren soll.

Schlussbemerkung

Mit dieser interessanten GUI setzen Sie also ein realistisches Szenario im Agrartechnikbereich programmtechnisch um, das durch einem einen Warenstrom von Hersteller => Großhandelsverkäufer => private Käufer von ungebrauchten Neugeräten => Gebrauchtgerätekäufer gekennzeichnet ist.



Aufgabe 5 (2-3 Personen)

Grafische Benutzeroberfläche für ein „Schiffe versenken“-Spiel

Das bekannte Spiel „Schiffe versenken“ soll mit einer grafischen Benutzeroberfläche simuliert werden. Dabei wird das altbekannte Spiel aus:

https://de.wikipedia.org/wiki/Schiffe_versenken

leicht verändert und auf mehr als 2 Spieler ausgeweitet.

Vor Spielbeginn

- Vor Spielbeginn soll die Anzahl an Spielern ausgewählt werden können, jedoch soll die **Spieranzahl zunächst im Bereich 2-4 begrenzt werden**.
- **Die Spielfelder**, auf denen die Schiffe der Spieler positioniert werden können, sollen **für jeden Spieler auf der GUI angezeigt werden**.
- Die **Spielfelder** sollen eine **quadratische Größe von 10x10** haben und können als eine Tabelle oder sonstige Matrixform in der GUI erscheinen. **(Add-on: optional soll die Matrixgröße der Felder in sinnvollen Unter- und Obergrenzen auswählbar sein)**
- Die **Felder** sollen von **1-10** auf der vertikalen Achse sowie A-J auf der horizontalen Achse nummeriert und angesteuert werden (bzw. *entsprechend bei optionalen Variante mit variablen Matrixgrößen*). Die beiden Achsen sollen am linken bzw. oberen Spielfeldrand angezeigt werden.
- **Farbauswahl für jeden Spieler**: jeder Spieler soll sich seine Spielfarbe aus einem Menü oder Farbmenü auswählen können.
- Der **Spieler soll die Positionen seiner Schiffe geeignet eingeben** können. (Im echten Spielverlauf müssen dann die anderen Spieler wegsehen, da das Programm nur so die Schiffspositionen aller Spieler erhalten kann).
- Die Schiffe dürfen nur **vertikal oder horizontal** positioniert werden.
- Das GUI-Programm soll warnen und **keine Schiffspositionen zulassen, die über den Spielfeldrand hinausgehen bzw. nicht vertikal oder horizontal ausgerichtet sind**.
- **Jeder Spieler hat sein eigenes Spielfeld, so dass sich Schiffspositionen von anderen Spielern an denselben Spielfeldkoordinaten nicht überlagern**.
- **Die Anzahl der Schiffe soll für alle Spieler gleich sein aber generell eingestellt werden können**. Der Auswahlbereich soll zwischen **1 und 5 Schiffen** liegen.
- Für jedes Schiff soll zudem die Schiffsgröße (minimal 1, maximal 5 Felder) eingestellt werden können. Z.B. angenommen alle Spieler verfügen über 2 Schiffe (Schiff A und B). Dann soll sowohl für Schiff A als auch für Schiff B deren **Schiffsgröße über eine Nutzerinteraktion auswählbar sein**. Die **ausgewählte Schiffskonstellation (Schiffsanzahl und Anzahl belegter Felder)** soll dann für jeden Spieler gleich sein.

Spielphase

In einer Runde kommen die Spieler nacheinander dran und sollen in jedem Spielzug 2 Aktionen ausführen können:

1. **Sich einen Spieler auswählen, auf den sie einen Schuss abfeuern möchten.** Bei 2 Spielern soll nur auf den direkten Gegenspieler abgefeuert werden können und die **Abfrage zur Spielerauswahl (auf den ein Schuss abgefeuert werden soll) in der GUI schon automatisch nicht erscheinen (oder zumindest ausgeblendet sein).**
2. Die Koordinaten für den Schuss sollen über eine geeignete Nutzerinteraktion ausgewählt werden können. Idealerweise soll das Koordinatensystem des ausgewählten Gegners in einem Extra-Dialog erscheinen und die Beschusskoordinaten mit der Maus auswählbar sein.

Nachdem die Beschusskoordinaten ausgewählt wurden, wird eine Rakete auf den ausgewählten Spieler an der eingestellten Koordinate abgefeuert. **Erfolgt ein Treffer, dann soll das Feld entsprechend markiert werden (z.B. mit einem X). Erfolgte kein Treffer, dann wird dies ebenfalls in der Matrix kenntlich gemacht (z.B. mit einer 0).**

Zudem **sollen 4 Geräuscharten als Audiodaten** in den Spielablauf eingebunden werden:

- Abschussraketenstart
- Wassereinschlag (kein Treffer)
- Teiltreffer
- Finaler Versenkungsschuss eines Schiffes

Nach jedem abgegebenen Schuss sollen einer der entsprechenden Geräusche ertönen. Vorschläge für verschiedenen Geräusche werden als mp3-Format auf der Moodle-Seite mitgeliefert. **Es können auch eigene Geräusche recherchiert oder konstruiert werden.**

Nachdem der letzte Spieler an der Reihe war, beginnt wieder Spieler Nr.1 mit seinen Aktionen. **Die Bedingungen für das Spielende sollen vom Benutzer eingegeben werden können.** Das Spiel kann grundsätzlich solange durchgeführt werden bis:

1. Nur noch ein Spieler Schiffe oder Schiffsteile besitzt. Dieser Spieler ist dann der Spielsieger.
2. Ein Spieler keine Schiffe mehr hat
3. Eine Anzahl an Spielrunden erreicht wurde.

Im Falle von 2. Und 3. muss die erreichte Punktzahl über den Sieger entscheiden:

- Ein Treffer auf ein Einerschiff ist 5 Punkte Wert
- Ein Treffer auf ein Zweierschiff ist 4 Punkte Wert
- Ein Treffer auf ein Dreierschiff ist 3 Punkte Wert
- Ein Treffer auf ein Viererschiff ist 2 Punkte Wert
- Ein Treffer auf ein Fünferschiff ist 1 Punkt Wert

Der Sieger soll mit seiner erreichten Trefferpunktzahl angezeigt werden.

Das Spiel Schiffe versenken existiert in verschiedenen Ausführungen auch im Internet. Einen Eindruck vom Ablauf bietet z.B.:

<https://www.spiele-kostenlos-online.de/brettspiele/brett-spiele/3d-schiffe-versenken/>

Beispielübersicht für ein Schiffe-versenken-Spiel mit 4 Spielern:

Spieler 1

	A	B	C	D	E	F	G	H	I	J
1										
2		0								
3		X	0		0			0		
4		X								
5		0								
6						0				
7										
8							0		0	
9										
10	0									

Spieler 2

	A	B	C	D	E	F	G	H	I	J
1										
2		0							0	
3						0				
4				0						
5										
6							0			
7		0								
8								0		X
9										X
10									0	

Spieler 3

	A	B	C	D	E	F	G	H	I	J
1										
2	0		0							
3			0							
4				0						
5						0				
6										
7										
8						0		0		
9										
10	0		0				0			

Spieler 4

	A	B	C	D	E	F	G	H	I	J
1		0								
2		X								
3	0	X	0							
4		X						0		
5		X								
6		X								
7										
8									0	
9										
10		0								

Aufgabe 6 (2-3 Personen)

Grafische Benutzeroberfläche zur Verwaltung von Medikamentenkäufen zwischen Patienten, Apotheken und Medikamentengroßhandel

Es soll eine graphische Benutzeroberfläche für verschiedene Akteure erstellt werden, die bei der Versorgung mit Medikamenten von Bedeutung sind. Diese Akteure sind: **Patienten, Ärzte, Apotheken und Medikamentengroßhandel**. Dabei soll die GUI ermöglichen, die verschiedenen Handlungen dieser Akteure in ihren Rollen anzuzeigen bzw. deren Daten zu verarbeiten.

In der Datei Apotheke_Patienten_Ärzte.xlsx sind die Daten für diese GUI-Programmieraufgabe enthalten, die in verschiedene Untertabellen (Karteireiter) unterteilt sind. In der Tabelle Patienten sind mehrere Patienten enthalten, die mit ihrer ID, Art der Krankenversicherung, Symptomatik sowie einem individuellen Maximalbudget Medikamentenkauf aufgelistet sind.

Rolle: Patient

Ein Patient soll sich mit der GUI anmelden können und sein initiales Passwort (ID) bei Bedarf verändern können. Die Passwörter der einzelnen Akteure können in einer Textdatei abgespeichert werden. Wenn ein Name eines unbekannten Patienten eingegeben wird, soll die GUI mit einem entsprechenden Hinweis reagieren. Der/die Patient(in) kann sich für seine/ihre Beschwerden einen der 4 Ärzte aussuchen oder ohne Arztbesuch bei einer Apotheke einkaufen.

Wenn er/sie ohne Arztbesuch ein Medikament kauft, kann er/sie nur nicht verschreibungspflichtige Medikamente erwerben. Entscheidet sich ein Patient für einen Arzt, dann soll der dauerhaft an diesen Hausarzt gebunden sein und keinen anderen Hausarzt aufsuchen können(!) Bei der Arztbehandlung wird mit einem Zufallsgenerator unter Einbeziehung der Verschreibungswahrscheinlichkeit des Hausarztes entschieden, ob der Hausarzt dem Patienten ein Medikament im konkreten Fall verschreibt. Wenn der Patient für seine Symptome kein Rezept erhält, kann dieser nur versuchen, ein Medikament für seine Symptome ohne Verschreibungspflicht bei einer Apotheke zu kaufen.

Der Patient soll sich die Preise aller zur Verfügung stehenden Medikamente für seine Symptome anzeigen lassen können. Dabei soll er eine Übersicht erhalten, bei welcher Apotheke welcher Preis aktuell veranschlagt wird (falls diese Apotheke das Medikament auf Lager hat). Er kann sich dann für ein Produkt entscheiden: dabei wird die Anzahl im Lager der Apotheke entsprechend um eine Mengeneinheit erniedrigt. Zudem wird dem Patienten ein Geldbetrag abgebogen. Dieser Geldbetrag wird folgendermaßen ermittelt:

- Im Falle einer privaten Krankenversicherung werden 70% des Preises und bei einer gesetzlichen (oder freiwillig gesetzlichen) Krankenversicherung werden 25% des Preises ermäßigt.

- Wenn das Rezept nicht verschreibungspflichtig ist, werden zusätzlich noch 20% auf den bereits durch den Krankenkassensatz erniedrigten Preis erlassen, wenn dem Patienten vom Arzt eine Verschreibung gewährt wurde.

Wenn das Rezept verschreibungspflichtig ist, kann der Patient es generell nur kaufen, wenn er eine Verschreibung durch seinen Hausarzt erhalten hat. Die GUI soll darauf hinweisen, wenn das Budget des Patienten für den ausgewählten Medikamentenkauf nicht ausreicht.

Wenn 10 Käufe durch einen oder mehrere Patienten durchgeführt wurden, soll mit einem Zufallsgenerator eine Betrag zwischen 1 und 50 € ermittelt werden. Dieser Zufallsgenerator soll für jeden Patienten angewendet werden, um die Patientenbudgets zufällig zu erhöhen.

Rolle: Apotheke

Eine Apotheke (bzw. ein Apothekenbesitzer) soll sich ebenfalls mit ihrer ID einloggen können und diese ID bei Bedarf verändern. Eine Apotheke kann aus dem Großhandel neue Medikamente hinzukaufen, um den Lagerbestand zu erhöhen. Zudem soll eine Apotheke entscheiden können, ob sie neue Medikamente in einer bestimmten Anzahl hinzukauft. Eine Apotheke hat ebenfalls ein Budget. Wenn dieses Budget aufgebraucht ist, kann die Apotheke erst wieder neue Medikamente aus dem Großhandel kaufen, wenn sie durch neue Verkäufe an Patienten ihre Budget aufgefüllt hat. Die Apotheke kauft zu Großhandelspreisen beim Großhandel ein. Wenn im Großhandel das von der Apotheke bestellte Medikament nicht mehr zur Verfügung steht, dann kann es von der Apotheke nicht gekauft werden.

Zudem soll die GUI eine Funktionalität bereitstellen, so dass bei Bedarf der **gesamte Bestand der Apotheke sowie das aktuelle Budget angezeigt** wird.

Rolle: neue Apotheke

Es soll in der GUI die Möglichkeit geben auch neue Apotheken aufzunehmen. Dabei soll der Bestand an bereits vorhandenen Medikamenten über geeignete GUI-Steuererelemente abgefragt werden und somit die neue Apotheke mit seinem Medikamentenbestand ins GUI-System aufgenommen werden können. Dabei sollen auch Name der Apotheke und das aktuelle Budget eingegeben werden können, so dass die neue Apotheke auch wieder Medikamente beim Großhandel kaufen kann oder an einen Patienten verkaufen kann.

Aufgabe 7 (2-3 Personen)

Grafische Benutzeroberfläche zur Aktienanalyse für Kunden eines Aktienfonds

Sie sind ein Startup-Unternehmen im Finanzdienstleistungsbereich und möchten ihren potentiellen Kunden eine grafische Benutzeroberfläche zum Zusammenstellen für ein Aktienportfolio anbieten. Ein Teil Ihrer Kunden schwört auf den Index deutscher Mittelstandfirmen (MDAX) und möchte sich nur aus diesen Aktien ein Portfolio zusammenstellen können.

Die 5 in der Excel-Datei aufgelisteten Kunden sind Interessenten ausschließlich für den MDAX und sollen sich mit Passwort und ID in die Benutzeroberfläche einloggen können. Passwort und ID sollen auch vom Kunden bei Bedarf wieder geändert werden können.

In der Excel-Datei MDAX_60 sind 60 Unternehmen mit einigen generellen Informationen (wie z.B. Branche, Firmensitz) sowie finanziellen bzw. anlagerelevanten Informationen (wie z.B. KGV, ISIN, Dividendenrendite).

Die folgenden Teilaufgaben stellen die wesentlichen Programmierleistungen für diese Aufgabe dar:

Teilaufgabe 1: Erstellung einer Auswahlmöglichkeiten zur Anlageinformation des Kunden

Der/die eingeloggte Kunde/Kundin soll mehrere Auswahlmöglichkeiten haben, um sich eine für ihn/sie interessante Unterauswahl in der GUI (oder in einem extra Fenster) anzeigen zu lassen. Zu Beginn sollen die in der Excel-Datei enthaltenen Informationen zur Übersicht entsprechend in einer Gesamtanzeige aufgelistet werden.

Der Kunde soll sich bei Bedarf beliebige Unterlisten einer Aktienausswahl nach den Spaltenbezeichnungen der Tabelle anzeigen lassen können, damit er sich auf bestimmte Aktien fokussieren kann, die seinem Anlegerverhalten entsprechen. Dabei sollen Unterlisten angezeigt werden können, die den Auswahlprinzipien des Nutzers entsprechen. Interaktiv auswählbar sollen folgende Eigenschaften (Spaltenbezeichnungen) sein:

- Die Branche: der/die Kunde/Kundin soll sich alle Unternehmen einer Branche anzeigen lassen. Die Branche soll sich idealerweise mit einem geeigneten GUI-Steuerelement zur Auswahl (z.B. eine Combobox) auswählen lassen.
- Für die Eigenschaften: Index-Gewichtung, Anzahl an Aktien, Marktkapitalisierung, KGV und Dividendenrendite sollen geeignete Einstellungen zur Auswahl realisiert sein (z.B. ein Schieberegler oder ein ähnliches Steuerelement), der diese Eigenschaften in geeigneten Zahlenschritten auswählbar gestaltet.
- Die restlichen Eigenschaften (Firmenname, Firmensitz und ISIN=internationale Wertpapierkennnummer) sollen nicht auswählbar sein. Sie sollen aber in den ausgewählten Unterlisten als zusätzliche Informationen erscheinen.
- Die Spalte „geheimer Trendwert“ soll nicht angezeigt werden können, da sie dem Kunden nicht zur Verfügung steht!

- In der 1.Spalte für die Firmennamen sind URL-Links angegeben. Der GUI-Benutzer soll sich bei einem Klick auf einen Firmennamen in der Unterliste weitere Informationen zum Unternehmen anzeigen lassen können (entweder in einem Webbrowser oder in der GUI selber).

Der Kunde soll sich aus der Gesamtanzeige oder den ausgewählten Unterlisten sein Aktienportfolio zusammenstellen können. Er soll sich dann entscheiden können, wie er sein Anlagekapital auf die einzelnen Portfolio-Positionen verteilt. Dabei muss er nicht sein gesamtes Anlagekapital ausnutzen müssen. Das Anlagekapital von 5 Beispielkunden ist ebenfalls in der Tabelle unten aufgeführt.

Teilaufgabe 2: Realisierung eines zufälligen Aktienverlaufs

Das ausgewählte Portfolio eines Kunden soll sich nach einer zufälligen Änderung der Aktienkurse über 12 Monate verändern. Dabei soll für jedes Monatsende eine Veränderung berechnet werden. Es soll eine Zufallszahl im Bereich -0.2 bis + 0.2 erzeugt werden (damit also zunächst eine monatliche zufällige Änderung von maximal 20% erzeugt werden können). Um den Kurs nicht rein zufällig entwickeln zu lassen, sollen zudem für jede Aktie die „geheimen Trendwerte“ einbezogen werden, die in der letzten Spalte der Excel-Tabelle stehen. Die monatliche Kurswertänderung einer Aktie (KW) soll nach folgender Formel berechnet werden:

Neuer KW für den Folgemonat = KW vom Vormonat + KW vom Vormonat * Zufallszahl + KW vom Vormonat * geheimer Trendwert

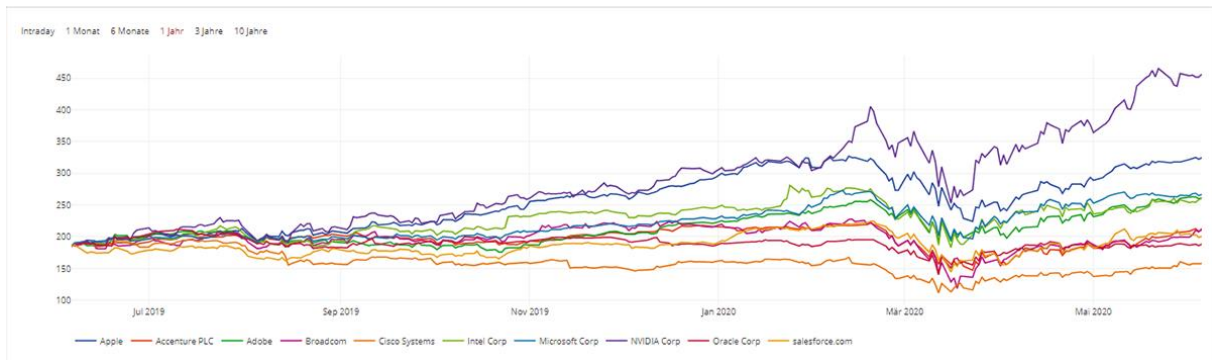
Die monatliche Kurswertänderung soll monatlich zum Aktienkurs hinzuaddiert (bzw. abgezogen) werden, um daraus den Aktienkurs nach einem Jahr zu ermitteln. Der Ausgangswert für den Aktienkurs entspricht dem in der Tabelle aufgelisteten Kurs (vom 07.10.22). Die monatliche Veränderung soll also 12 mal ermittelt werden so dass sich 12 monatliche Kurswerte ergeben.

Das Portfolio soll nach 12 Monaten ebenfalls angezeigt werden können, um einen Gesamtgewinn oder Verlust für jeden Kunden anzeigen zu können.

Teilaufgabe 3: Anzeige und Gegenüberstellung von Aktiencharts

Jeder Kunde soll sich die 12 monatigen Aktienverläufe sollen für jede Aktie seines Aktiendepots anzeigen lassen können. Diese Aktienverläufe bzw. Charts können z.B. mit der Bibliothek Matplotlib in einem Linienplot angezeigt werden. Dabei soll der Anleger die Möglichkeit erhalten, sich die Charts von mehreren Aktien seines Aktiendepots (z.B. in verschiedenen ausgewählten Farben) anzeigen zu lassen. Damit soll der Kunde die Aktienverläufe seiner Einzelpositionen visuell vergleichen können.

Aktiencharts-Vergleich Beispiel 1



Aktiencharts-Vergleich Beispiel 2



Aufgabe 8 (2-4 Personen)

Grafische Benutzeroberfläche für ein erweitertes Memorykarten-Spiel

Es soll ein Memoryspiel in einer Benutzeroberfläche realisiert werden, das die grundsätzliche Spielidee von Memory um mehrere Varianten und Spielfeldanordnungen erweitert. Dabei werden zufällige Veränderungen wie das Vermischen der Karten im gesamten Spielfeld oder in ausgewählten Spielfeldbereichen an bestimmten Punkten des Spielablaufs integriert, um das Memoryspiel zufälliger und interessanter zu gestalten. Aufgrund der begrenzten Darstellungsfläche auf dem Bildschirm soll eine Extra-Anzeige zur Darstellung der aktuell von einem Spieler ausgewählten Karten implementiert werden. Optional können Zoom- und Verschiebefunktionalitäten programmiert werden, um einzelne Spielfeldbereiche vergrößert anzusteuern (die aber zur Erfüllung der Aufgaben nicht zwingend notwendig sind).

Vorbereitung der Spielphase

Für die Memorybilder bzw. Spielkarten liegen Bilder (hauptsächlich in jpg-Format) aus 3 Kategorien vor (Comicfiguren, lustige Tieraufnahmen, atemberaubende Stunts). Die Benutzer der GUI sollen **vor Spielbeginn auswählen können, für welche Bilderkategorie sie sich entscheiden**. Dafür soll es eine interaktive Auswahlmöglichkeit geben. Zudem müssen die Bilder für die Anzeige verdoppelt werden, da für jedes Bild jeweils 2 Bilder auf dem Spielfeld angezeigt werden müssen. Ihnen ist freigestellt, ob Sie von jedem Bild eine Kopie erstellen möchten oder dasselbe Bild einfach an 2 Stellen im Memoryspielfeld anzeigen lassen möchten.

Gleiche Größe der Bilder: Da die Bilder unterschiedliche Größen und Auflösungen haben können, müssen sie in der Spielfeldansicht immer auf eine Einheitsgröße skaliert (verkleinert bzw. vergrößert) werden, damit kein Spieler anhand der Größe auf den Karteninhalt schließen kann. In der Startansicht sollen der Übersichtlichkeit wegen zunächst alle verwendeten Memorybilder angezeigt werden (auch wenn sie z.B. bei 80 Karten kaum erkennbar sind).

Des Weiteren soll auswählbar sein, wie viele Spieler am Memoryspiel teilnehmen möchten: es sollen **2 bis 4 Spieler auswählbar** sein.

Von jeder Bilderkategorie existieren aktuell 40 Bilder, die bei Verdopplung maximal 80 Memorykarten ergeben. Die GUI-Nutzer sollen jedoch **3 Wahlmöglichkeiten für die Menge der Memorykarten** haben. Zunächst soll auswählbar sein, ob das Memoryspiel mit 40 (2x20), 60 (2x30) oder 80 (2x40) Karten (inklusive doppelte) gespielt werden soll. Nachdem sich die Spieler auf eine Kartenanzahl geeinigt haben, werden die Karten aus der gewählten Kartenkategorie verwendet. Falls nur 20 oder 30 Karten gewählt werden, sollen die Bilder aus der gewählten Bilderkategorie (mit jeweils 40 Bildern) zufällig zusammengestellt werden.

Jokerkarten ergeben im Spielverlauf 3mal so viele Punkte wie die anderen Karten. Deshalb soll die Anzahl an Jokerkarten von den Benutzern vor Spielbeginn bestimmt werden können. Dafür soll eine Auswahlmöglichkeit von 0-5 Jokerkarten existieren. Die Nutzer einigen sich durch ihre interaktive Auswahl auf eine Anzahl an Jokerkarten. Erst im aufgedeckten Zustand sollen die Spieler darüber informiert werden, ob es sich bei der aufgedeckten Karte um eine Jokerkarte handelt. Die Jokerkarten sollen dann entsprechend markiert werden: z.B.: mit

einem **farbigen Rand** um die Karte herum. Welche der Spielkartenpaare Jokerkarten sind, soll vom Programm zufällig am Spielbeginn bestimmt werden.

Variable Spielfeldanordnungen

Häufige Anzahlen an Spielkarten in kommerziellen Memoryspielen liegen zwischen 64 und 72 Karten. Trotz der Platzbeschränkung auf einem Bildschirm, soll in dieser Aufgabe versucht werden, mit den 3 Spielkartenanzahlen (40, 60 oder 80) diese große Kartenmenge unter Zuhilfenahme einer Extra-Anzeige (aktuell aufgedeckter Karten) oder von optionalen, interaktiven Zoomfunktionen zu beherrschen. Für die 3 Spielkartenanzahlen (40, 60 oder 80) soll es mehrere rechteckige Anordnungen der einzelnen Memorykarten geben, die z.B. in einer Art „Array“ vorliegen können:

Bei 80 Spielkarten soll eine der folgende matrixartigen Anordnungsmöglichkeiten zu Spielbeginn von den GUI-Nutzern auswählbar sein:

4x20, 5x16, 8x10, 10x8, 16x5, 20x4

Bei 60 Spielkarten:

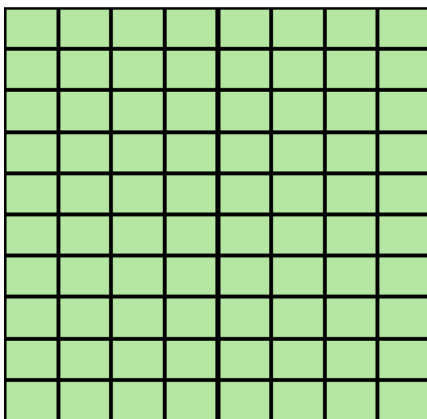
4x15, 5x12, 6x10, 10x6, 12x5, 15x4

Bei 40 Spielkarten:

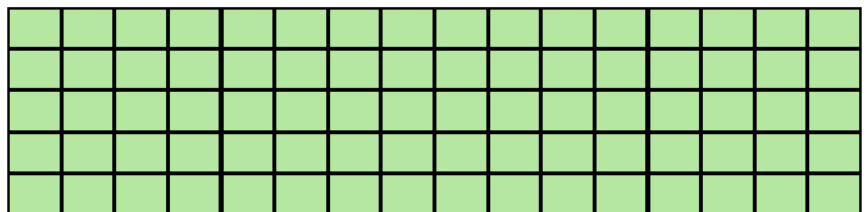
2x20, 4x10, 5x8, 8x5, 10x4, 20x2

Beispielhafte Matrix-Anordnungen bei 80 Memorykarten

10x8



5x16



Zusammenfassend sollen also 5 Einstellungen kurz vor Spielbeginn abgefragt werden:

- Anzahl der Spieler: 2, 3 oder 4
- Bildkategorie (es können auch eigene Bilder verwendet werden. Die bereits zusammengestellten Bildkategorien sollen nur der Arbeitserleichterung dienen)
- Spielkartenanzahl: 40, 60 oder 80

- Anzahl der Jokerkarten
- Die Matrixform der rechteckigen Spielfeldanordnung

Spielphase

Nachdem diese Voreinstellungen durchgeführt wurden, sollen die Memorykarten (Bilder) nach einem Zufallsprinzip den einzelnen Matrixfeldern zugeordnet werden. Alle Karten werden dabei im verdeckten Zustand ausgelegt. Danach läuft das Spiel prinzipiell wie in einem gewöhnlichen Memoryspiel ab:

Die Spieler sind nacheinander an der Reihe. Wenn ein Spieler an der Reihe ist, dann deckt er 2 Karten per Mausklick nacheinander auf, die in einer vergrößerten Extra-Anzeige dargestellt werden (s.u.). Wenn diese Karten gleich sind, dann bekommt der Spieler 2 Punkte auf seinem Punktekonto gutgeschrieben. Die beiden Karten werden danach aus dem Spiel genommen und die Stellen, an denen sich die beiden aufgedeckten Karten befunden haben, werden als „nicht mehr im Spiel vorhanden“ markiert. Diese Markierung kann eigenständig gewählt werden (z.B. durch ein Kreuz). Es sollte nur damit deutlich gemacht werden, dass an diesen Stellen keine Karten mehr aufgedeckt werden können. Wenn ein Spieler an diesen „bereits aufgedeckten“ Stellen eine Karte per Mausklick aufdecken will, soll dieser mit einer entsprechenden Meldung darauf hingewiesen werden, dass das nicht mehr möglich ist.

Deckt ein Spieler 2 zueinander passende Jokerkarten auf, dann bekommt er dafür $3 \times 2 = 6$ Punkte auf seinem Punktekonto gutgeschrieben. Wie bereits oben beschrieben sollen die Jokerkarten nur im aufgedeckten Zustand (in der Extra-Anzeige) mit einem farbigen Rand gekennzeichnet werden, damit sie für alle Spieler erkennbar sind.

Ein Spieler ist solange an der Reihe, wie er gleiche Kartenpaare aufdeckt. Wenn ein Spieler unterschiedliche Karten aufdeckt, ist der nächste Spieler an der Reihe. Das Spiel ist zu Ende, wenn es keine Karten mehr auf dem Spielfeld gibt, die man aufdecken kann. Der Spielsieger wird anhand des höchsten Stands der einzelnen Punktekonto ermittelt.

Zufälliges Mischen der Positionen (=> Ärgern des gewinnenden Spielers)

Generell soll in dieser erweiterten Form des Memoryspiels mehr Zufälligkeit integriert werden, um das Spiel spannender zu gestalten. Dabei sollen die Positionen der Memoryspielkarten zufällig vermischt werden, was aber nur im begrenzten Ausmaß während des Spiels ermöglicht werden soll. Ein Spieler soll die Möglichkeit erhalten, das Spielfeld bzw. nur Bereiche des Spielfelds zufällig neu anordnen zu lassen. Es sind 2 Formen von zufälligen Kartenvermischungen möglich sein:

- 1.) Alle Spielkarten neu mischen:** Jeder Spieler soll **2x während des gesamten Spiels** die Möglichkeit haben, alle Spielkarten zufällig mischen (bzw. ihre Positionen austauschen) zu lassen. Dabei werden auch die bereits aufgedeckten (und nicht mehr ansteuerbaren) Karten mit vermischt. In diesem Zusammenhang kann er auch bestimmen, ob das gesamte Spielfeld neu angeordnet werden soll. Wie im Kapitel

„*Variable Spielfeldanordnungen*“ beschrieben, gibt es für jede Spielkartenanzahl eine Auswahl an möglichen Spielfeldanordnungen, die ebenfalls vom Spieler in diesem Schritt verändert werden kann (aber nicht muss). **Das Mischen aller Memorykarten kostet den Spieler allerdings 8 Punkte**, die ihm von seinem Punktekonto abgezogen werden.

- 2.) Nur einen Quadranten des Spielfelds mischen:** Jeder Spieler soll **während des gesamten Spiels 2x die Möglichkeit** bekommen, nur in einem Viertel (Quadranten) des Spielfelds die Karten neu zu mischen. Der Quadrant soll für den Spieler entsprechend auswählbar sein: z.B. „links oben“, „rechts oben“, „links unten“, „rechts unten“. Die zu vermischenden Kartenpositionen liegen dann nur im angegebenen Spielfeldquadranten, dessen Kartenpositionen bezüglich der aktuell eingestellten matrixartigen Spielkartenanordnung ermittelt werden sollen. Bei dieser Vermischungsart der Karten soll der Spieler nicht die Möglichkeit wie unter 1) haben, die matrixartige Spielfeldanordnung zu verändern. **Die Kartenvermischung in einem Quadranten kostet den Spieler 4 Punkte**, die ihm von seinem Punktekonto abgezogen werden.

Extra-Anzeige aktuell aufgedeckter Karten

Vermutlich werden bei der großen Kartenanzahl (vor allem bei 80 Spielkarten) die Erkennung einzelner Karten auf dem begrenzten Platz des Bildschirms bzw. innerhalb einer GUI schwierig sein. Aus diesem Grund können zusätzliche Zoom- und Verschiebefunktionen für das Spielfeld implementiert werden, um bestimmte Spielfeldbereiche zu vergrößern. Da dies ein besonderer Aufwand sein kann, ist für diese Aufgabe zunächst eine extra-Anzeige ausreichend, welche die im aktuellen Spielzug aufgedeckten Karten anzeigt. Der Spieler, der aktuell an der Reihe ist, soll per Mausklick nacheinander 2 Spielkarten auf dem Spielkartenfeld anklicken können, die in einer extra-Anzeige (entweder innerhalb der GUI oder in einem extra-Fenster) angezeigt werden. In der vergrößerten Extra-Anzeige sollen die Spielkarten ausreichend groß zu sehen sein, damit die Spieler sehen können, ob übereinstimmende Karten aufgedeckt wurden. Das Programm soll automatisch benachrichtigen, ob ein Kartenpaar richtig aufgedeckt wurde oder nicht. Zudem sollen die Punkte im Falle der Übereinstimmung der beiden aufgedeckten Spielkarten (für normales Kartenpaar oder Jokerkartenpaar) automatisch vom Programm zum Punktekonto des Spielers hinzuaddiert werden.

Timerfunktion pro Spielzug

Da in vielen Memoryspielen die Karten sehr zufällig (und oft nicht matrixartig) auf einem Tisch ausgelegt werden, ist durch die matrixartige Anordnung das Auffinden der Karten eventuell einfacher. Aus diesem Grund sollen auch keine Koordinatenangaben für die einzelnen Karten angezeigt werden, da sich somit die Spieler die Karten einfacher merken können. Dennoch kann die matrixartige Spielfeldanordnung das Auffinden der Kartenpaare vereinfachen, so dass zur Erschwerung eine Timerfunktion mit zeitlicher Begrenzung des Spielzuges

implementiert werden soll. Ein Spieler soll höchstens 20 Sekunden Zeit für das Aufdecken eines Kartenpaares haben. Braucht er mehr Zeit (> 20 Sekunden aber < 30 Sekunden) werden ihm 2 Punkte abgezogen. Nach mehr als 30 Sekunden kommt der nächste Spieler an der Reihe, für den der Timer wieder neu die Zeit zählt. Aus diesem Grund muss die Spielzeit des aktuellen Spielers immer auf der GUI angezeigt werden.

Sichtbarkeit des Punktekontostands

Das Punktekonto aller Spieler soll zu jeder Zeit in der GUI sichtbar sein und bei Punktänderungen aktualisiert werden. Aktualisierungen ergeben sich nur durch Addition von normalen Punkten oder Jokerpunkten übereinstimmender Kartenpaare oder durch Abzüge beim zufälligen Vermischen der Karten bzw. auch beim Überschreiten der 20 Sekunden Spielzeit für einen Spielzug.

Hinweise zur Gruppenbildung

Aufgrund der Komplexität der Aufgaben soll in 2er oder 3er Gruppen gearbeitet werden. Die Aufgaben 4 und 8 wurden erschwert, so dass hier 2 bis 4 Teilnehmer/innen möglich sind. Einzelgruppen (bzw. die Abgabe der Programmierlösungen durch nur eine Person) ist nicht mehr zulässig! Ein Ziel dieses Praktikums ist auch, dass Sie sich durch eine gute Verständigung und Kooperation im Team (Teamfähigkeit bzw. soft-skills sind heutzutage im Berufsleben vieler Bereiche unerlässlich und manchmal sogar wichtiger als fachliche Kenntnisse) gemeinsam und schrittweise an eine gute und Sie selber zufriedenstellende Lösung heranarbeiten.

Nehmen Sie zur Gruppenbildung bitte Kontakt zu ihren Kommilitonen/Kommilitoninnen auf, z.B.:

- in den Lehrveranstaltungen oder per E-Mail.
- Oder am besten durch unser „Diskussionsforum zur Gruppenfindung für die Praktikumsaufgaben“ auf Moodle-Seite von diesem Kurs (direkt unterhalb der Praktikumsaufgaben). Dort kann jeder/jede, der/die noch eine Gruppe sucht eine Nachricht schreiben.

Generelle Hinweise für alle Praktikumsaufgaben

Um die Kriterien für eine gute Usability zu garantieren sollen optional auch undo-Operationen implementiert werden. Das Einbinden von Undo-Operationen bei den einzelnen GUIs ist ein Add-on und soll nur optional (bei Interesse und genügend Restzeit) behandelt werden. Geeignete Quellen finden sich für Python unter:

Einfache und/redon Möglichkeiten für Text Buttons in TkInter:

https://www.youtube.com/watch?v=yd2a_olJ4WM

Undo/redon mit dem Command Design Pattern in Python:

https://medium.com/@sean_bradley/command-design-pattern-in-python-2f15b09f3774

<https://medium.com/design-patterns-in-python/undo-redo-pattern-in-python-70ade29644b3>

https://www.youtube.com/watch?v=FM71_a3txTo&t=1036s

Generelle Hilfen zum Arbeiten mit Verzeichnissen und Dateien sowie Arbeiten mit Bildern in Python:

Bilder Laden und Speichern:

<https://towardsdatascience.com/loading-and-saving-images-in-python-ba5a1f5058fb>

<https://www.delftstack.com/de/howto/numpy/save-numpy-array-as-image/>

Öffnen von Dateien und Iterieren über Verzeichnisse:

<https://www.geeksforgeeks.org/how-to-iterate-over-files-in-directory-using-python/>

<https://www.codegrepper.com/code-examples/python/iterate+over+folders+python>

Das Speichern von Passwörtern kann für alle Praktikumsaufgaben in einfachen Textdateien geschehen, wo die entsprechenden Zugangsdaten abgelegt sind.

Mp3-Dateien in Python abspielen:

<https://www.delftstack.com/de/howto/python/python-play-mp3/>

Erzeugen von Tabellen mit Tkinter bzw. Zeichnen mit Python-Canvas:

<https://pythonguides.com/python-tkinter-table-tutorial/>

<https://www.geeksforgeeks.org/create-table-using-tkinter/>

<https://anzelg.github.io/rin2/book2/2405/docs/tkinter/canvas.html>