

Seminararbeit

Interactive Games

Thema:

Entwerfen Sie ein Spiel mit Tablett und grafischem Peneinsatz.
Prototypische Realisierung.

Autoren:

Sylvia Berndt (706758)
Tim Friese (702960)
Katja Warzecha (703280)

Universität Potsdam, Institut für Informatik,
Seminar Nonlinear Media „Showcase Nichtlineare Medien“

Wintersemester 2005/2006

Prüfer: Prof. Dr. Klaus Rebenburg
Dipl.-Inform. Alexander Schulze

I. Inhaltsverzeichnis

I.	Inhaltsverzeichnis.....	III
II.	Abbildungsverzeichnis	IV
III.	Vorwort	V
A	Tablett-PCs	1
1	Arten von Tablett-PCs.....	1
2	Anwendungen für den Tablett-PC.....	2
3	Digitizer.....	4
3.1	Funktionsweise.....	4
3.2	Probleme des Digitizers	5
4	Semantik des Stiftes	7
5	Fazit zum Tablett -PC	9
B	Entwicklung des Spiels.....	11
1	Entwicklungsumgebung	11
2	Realisierung.....	12
3	Testprogramme.....	12
3.1	InkAnwendung	12
3.2	TablettAPI	13
4	Mystical Ink.....	14
4.1	Spielkonzept	14
4.2	Spielablauf.....	15
4.3	Handschriftenerkennung	17
4.4	Steuerung des Rades.....	19
4.5	Datei auslesen.....	20
4.6	Probleme bei der Entwicklung	22
4.7	Spielanleitung	22
5	Fazit zum Prototypen des Spiels	25
6	Ausblick	25
IV.	Literaturverzeichnis.....	VII
1	Bücher	VII
2	Links.....	VII

II. Abbildungsverzeichnis

Abbildung 1: LIFEBOOK P1510	1
Abbildung 2: Acer TravelMate C312XMi.....	1
Abbildung 3: LS800 "Centrino"	1
Abbildung 4: CF 18 Toughbook	1
Abbildung 5: Elektrostatischer Digitizer	4
Abbildung 6: Elektrostatischer Digitizer	4
Abbildung 7: Elektromagnetischer Digitizer	5
Abbildung 8: Elektromagnetisches Gitter und Hardwarekomponenten	5
Abbildung 9: Parallaxe - Problem.....	6
Abbildung 10: Aufbau der Klassenbibliothek Microsoft.Ink	11
Abbildung 11: Pen -, Ink - und Reco API.....	11
Abbildung 12: InkAnwendung	12
Abbildung 13: TablettAPI.....	13
Abbildung 14: Initialisieren und aktivieren des InkCollektors.....	17
Abbildung 15: Interpretation der Handschrifteneingabe	18
Abbildung 16: UpdateOutput - Sammeln der Daten.....	19
Abbildung 17: XML Struktur	20
Abbildung 18: Klassenobjekt zum XML-Datei auslesen	20
Abbildung 19: Schleife zum XML-Datei auslesen.....	20
Abbildung 20: Erzeugung der Zufallszahl.....	21

III. Vorwort

Dieses Dokument stellt die Ausarbeitung des Projektes „Interactive Games“ im Rahmen des Seminars Nonlinear Media „Showcase Nichtlineare Medien“ im WS 05/06 des Lehrstuhls Netzwerktechnologien und multimediale Teledienste der Universität Potsdam dar. Die zum Dokument zugehörige Webseite, auf der auch Videos, Folien und Programme zum Projekt heruntergeladen werden können, ist unter folgenden Adressen zu finden:

- <http://www.semanticmedia-showcase.de/WerkstattSM/WS0506/interactiveGames.htm>
- <http://fara.cs.uni-potsdam.de/~sberndt/V1/uni/semantic2/website/interactiveGames.htm>

In diesem Projekt wurde prototypisch ein Spiel für einen Tablett-PC mit grafischem Stifteinsatz entwickelt. Dafür sollten die verschiedenen Möglichkeiten, die der Einsatz eines Stiftes bietet, ausgenutzt werden.

Grundsätzlich werden in diesem Dokument die zwei Kapitel „Tablett-PCs“ und „Entwicklung des Spiels“ abgehandelt. Zunächst werden im ersten Kapitel die unterschiedlichen Tablett-PC - Arten näher erläutert, um die Unterschiede zu herkömmlichen Notebooks hervorzuheben. Auch die Software eines Tablett-PCs, ist speziell für diesen entwickelt bzw. modifiziert worden. Im Abschnitt *Anwendungen für den Tablett-PC*, werden auszugsweise einige dieser Programme vorgestellt.

Eine Besonderheit stellt der *Digitizer* des Tablett-PCs dar, weshalb im gleichnamigen Abschnitt dessen Arten, Funktionsweise und Probleme näher erläutert werden. Des Weiteren wird im Unterabschnitt *Semantik des Stiftes* auf die Eigenheiten des Umgangs mit dem Stift hingewiesen. Abgeschlossen wird dieses Kapitel mit einem Fazit zum Tablett-PC.

Im zweiten Kapitel wird zunächst erläutert, auf welche Besonderheiten bei der Programmierung, gerade im Bezug auf die Entwicklungsumgebung, geachtet werden muss. Dem schließt sich die Vorstellung der zwei Testprogramme an, welche die spätere Grundlage für das prototypisch entwickelte Spiel „Mystical Ink“ bildeten.

Der Abschnitt *Mystical Ink* befasst sich mit dem Prototypen des Spiels. Zuerst wird das Spielkonzept erläutert und danach der Spiellauf selbst, das Auslesen der Datei, die Handschriftenerkennung und das Steuern des Rades inklusive der Gewinnberechnung, vorgestellt. Hierbei wird auf einige technische Details eingegangen, die den wesentlichen Kernelementen des Spieles entsprechen und einen kurzen Blick auf die Realisierung des Prototyps ermöglichen sollen, ohne den gesamten Quelltext lesen zu müssen. Die Probleme, die sich während der Programmierung ergaben, werden im Abschnitt *Probleme* bei der Entwicklung dargestellt. Um das Spiel für den Benutzer verständlicher zu gestalten, werden im Abschnitt *Spielanleitung* Hilfestellungen zu Handhabung des Spiels gegeben. Schließlich bilden die Abschnitte *Fazit zum Prototyp des Spiels* und *Ausblick* den letzten Teil des Zweiten Kapitels.

Beim Lesen des Dokumentes sind folgende Formatierungsarten zu beachten:

- Hervorhebungen
- Quelltextauszüge

Potsdam, den 21.02.2006

A Tablett-PCs

1 Arten von Tablett-PCs

Tablett-PCs sind Notebooks, deren Bauform einem Tablett ähneln. Man unterscheidet zwei Arten von Tablett-PCs, *Convertibles* und *Slates*. Ihnen voran gegangen ist das T100X-Dynapad von Toshiba (1993). Es besaß ein berührungsempfindliches LCD mit 16 Graustufen und wurde mit dem Betriebssystem MS-DOS 6.2 und der Software Windows 3.1, inklusive dem *Microsoft Pen Services for Windows* Treiber, ausgeliefert.



Abbildung 2: Acer TravelMate C312XMi

Trennung von Tastatur und Bildschirm mittels Docking-Station. Die Eingabe erfolgt in der abgekoppelten bzw. in der gedrehten Variante über einen Stift. Ein Beispiel für ein *Convertible* ist das TravelMate C312XMi von Acer. Manche Vertreter der *Convertibles*, wie das LIFEBOOK P1510 von Fujitsu, besitzen einen berührungsempfindlichen Bildschirm (Touchscreen), welcher sich mit jedem beliebigen Stift oder auch mit den Fingern selbst, bedienen lässt.



Abbildung 3: LS800 "Centrino"

Jeder *Slate* lässt sich durch Anschließen externer Geräte, wie z.B. Maus und Tastatur, zu einem stationären Arbeitsplatz erweitern. Ein Vertreter der *Slates* ist z.B. das LS800 "Centrino" Base von Motioncomputing.

Outdoor Tablett-PCs können sowohl *Slates* als auch *Convertibles* sein. Ihre Besonderheiten liegen in der extremen Robustheit, der höheren Akkulaufzeit und der angepassten Hardwarekomponenten, wie z.B. stoßfeste Festplatten und kontrastreichere Displays, die gerade für den Außeneinsatz von größerer Bedeutung sind. Panasonic liefert mit dem CF 18 Toughbook einen Tablett-PC, der speziell für den Outdoorbereich entwickelt wurde.



Abbildung 4: CF 18 Toughbook

Convertibles ähneln in Bezug auf ihre Funktionalität fast vollkommen einem Notebook. Sie können jedoch nicht nur über integrierte Tastatur und Maus (Touchpad od. Touchstick) bedient werden, sondern auch zusätzlich über den Digitizer mit dem zugehörigen Tablett-Stift. Ein wesentlicher Unterschied zum herkömmlichen Notebook ist die Möglichkeit, das Display mittels Kardangelenks um 180° zu drehen und so auf die Tastatur zu legen. Einige Arten von *Convertibles* ermöglichen die völlige



Abbildung 1: LIFEBOOK P1510

Slates, auch *Pure Tablets*, sind Bildschirmgeräte ohne integrierte Tastatur oder Maus. Da sie nur über Stifteingaben oder Funktionstasten bedient werden können, stellen *Slates* die

2 Anwendungen für den Tablett-PC

In diesem Bereich, werden Anwendungen vorgestellt, welche speziell für den Tablett-PC konstruiert wurden. Sie lassen sich auf normalen PCs oder Notebooks nur eingeschränkt oder im schlechtesten Fall gar nicht nutzen. Ein großer Unterschied zu "normalen" Anwendungen stellt die Handschriftenerkennung dar, welche von vielen dieser Programme verwendet wird. Diese Art der Texteingabe erspart dem Benutzer die Verwendung der Tastatur.

Im Folgenden werden einige dieser Anwendungen kurz erläutert:

- **MS Education Pack**, bestehend aus:
 - **Equation Writer:** Dieses Programm ermöglicht per Stifteingabe mathematische Ausdrücke und Gleichungen zu schreiben. Für Weiterverwendung in z.B. wissenschaftlichen Arbeiten können diese dann automatisch erkannt und umgewandelt werden.
 - **GoBinder Lite:** Mit dieser Anwendung lassen sich persönliche Termine über einen Kalender verwalten. Gleichzeitig dient es zur Verwaltung von Notizzetteln.
 - **Hexic Deluxe:** Hexic Deluxe ist ein Denksport Spiel, bei dem Elemente nach bestimmten Regeln gedreht werden müssen, um Punkte zu erzielen.
 - **Ink Flash Cards:** Zum Lernen oder zur Lehrstoffaufbereitung eignet sich dieses Programm besonders gut, denn es stellt virtuelle Notizkarteikarten zur Verfügung, die handschriftlich beschrieben und mit Grafiken versehen werden können. So kann z.B. eine Frage auf die Vorderseite einer Karte und die dazugehörige Antwort auf die Rückseite geschrieben werden. Wie mit „normalen“ Karteikarten, kann sich der Anwender zum Bsp. selbst prüfen.
- **MS Experience Pack**, bestehend aus:
 - **Ink Art:** Ein Zeichenprogramm mit dem man, mittels Stifteingabe, Zeichnungen erstellen kann.
 - **Media Transfer:** Mit dieser Anwendung ist es möglich Mediendateien von einem anderem PC zum Tablett-PC zu kopieren oder zu streamen. Als Mediendateien sind hierbei Musik- und Video-Dateien bzw. Fotos (Bilddateien) zu verstehen. Diese können mittels Tablett-PC überall angesehen werden, unabhängig davon wo sich der Benutzer befindet.
 - **Ink Crossword:** Ein Kreuzworträtselprogramm, das durch handschriftliches Eingeben der einzelnen Buchstaben mit dem Stift gelöst werden kann.
- **Tablett Enhancements for Outlook 2.0:** Eine Erweiterung für Outlook, die es ermöglicht, das Eingeben von Kontakten, das Eintragen von Daten in das Outlook Journal und die Terminplanung über den Stift zu realisieren.
- **PDF Annotator:** Ein nettes Werkzeug, mit dem handschriftliche Notizen in ein PDF - Dokument eingetragen werden können. Zusätzlich können Textpassagen oder wichtige Worte in unterschiedlichen Farben markiert werden.

- **Webplanet Concept X7:** Mit diesem Programm können Office Dokumente verwaltet werden. Es enthält Meeting-Vorlagen, ermöglicht das Erstellen von Business Konzepten per Drag & Drop und das Verwalten von Notizen. Somit werden Brainstorming, Meetings, Ideen & Notizen, Klick Konzepte und Projektplanung durch handschriftliche Eingabe realisiert.
- **xThink Calculator:** Ein kompletter Taschenrechner der vollständig über den Tablett-Stift bedient wird.
- **nTcMedic:** Ein Programm im Bereich der Medizintechnik. Es dient zur Erfassung und Verarbeitung der unterschiedlichsten Patientendaten – „elektronische Patientenakte“. Die Dokumentation von Daten im klinischen Alltag wird unterstützt und dient der Entlastung des klinischen Personals. Abrechnungsrelevante Patientendaten, z.B. Diagnosen oder Prozeduren werden erfasst und an vorhandene administrative Systeme elektronisch weitergegeben

3 Digitizer

3.1 Funktionsweise

Man unterscheidet drei verschiedene Arten von Digitizern: berührungsempfindliche, passive und aktive Digitizer.

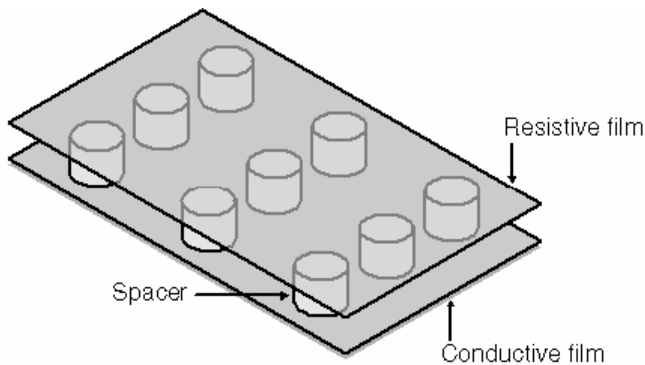


Abbildung 5: Elektrostatischer Digitizer

die beiden Schichten und an der Druckstelle entsteht ein elektrischer Kontakt. Der nun entstehende Widerstand des Kontaktes führt dazu, dass an jedem Punkt auf dem Digitizer eine andere Spannung anliegt, welche separat gemessen werden kann. Somit lässt sich durch diese Spannungsänderung die Position der Berührung auf dem Digitizer bestimmen.

Berührungsempfindliche Digitizer (Touchscreen) benötigen keine speziellen Stifte und können auch durch den Finger bedient werden. Diese Digitizer bestehen aus zwei Indiumzinnoxidschichten, welche mit einer konstanten Gleichspannung orthogonal zueinander angesteuert werden. Abstandshalter, so genannte Spacer, trennen die Schichten voneinander. Wird Druck auf die obere Schicht ausgeübt, berühren sich

Elektrostatische Digitizer (passive Digitizer) werden durch ein elektromagnetisches Feld gespeist. Dieses Feld induziert in den Stift, welcher mit einer Spule ausgestattet ist, eine Spannung, die dazu führt, dass der Stift ebenfalls ein elektromagnetisches Feld erzeugt. Mit dem elektromagnetischen Feld des Stiftes kann das elektromagnetische Feld des Digitizers manipuliert werden. Durch diese Manipulation ändern sich Spannung und Widerstand im Digitizer und die Position des Stiftes lässt sich anhand dieser Änderungen bestimmen. Auch Neigungswinkel, Rotation und Druck des Stiftes lassen sich durch diese Änderungen im elektromagnetischen Feld des Digitizers bestimmen. Bewegungen des Stiftes, über dem Digitizer, werden ebenfalls registriert, auch wenn der Digitizer dabei nicht berührt wird.

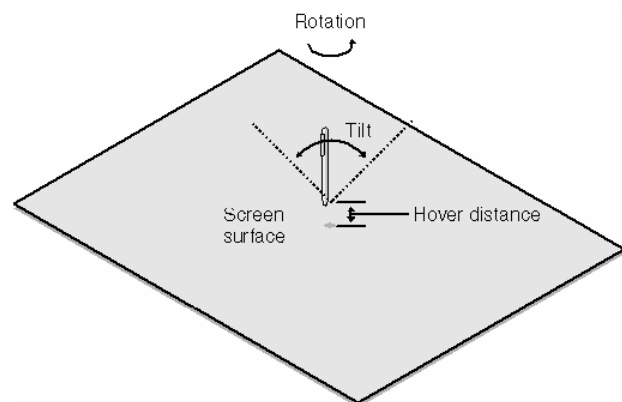


Abbildung 6: Elektrostatischer Digitizer

Der elektromagnetische Digitizer (aktive Digitizer) ähnelt in seinem Funktionsumfang dem passiven Digitizer. Unterschiedlich sind jedoch der benutzte Stift und der Aufbau des Digitizers. Stifte für aktive Digitizer besitzen eine eigene Stromversorgung, welche die im Stift enthaltene Spule induziert. Auch hier wird mit dem elektromagnetischen Feld des Stiftes der Digitizer manipuliert. Der aktive Digitizer befindet sich, im Gegensatz zum passiven Digitizer, hinter dem Display, da dieser aus vielen Spulen besteht und somit die

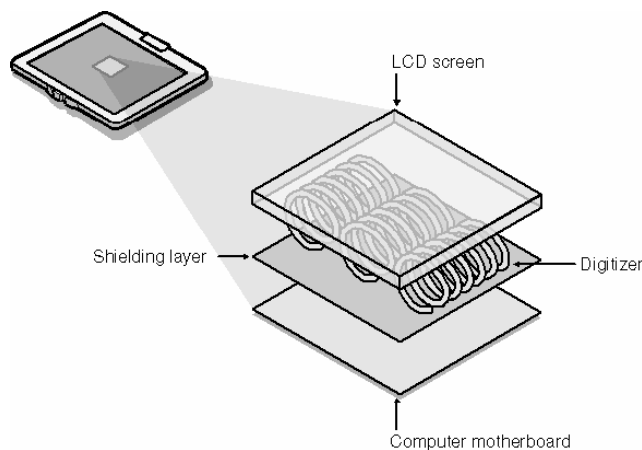


Abbildung 7: Elektromagnetischer Digitizer

Sicht auf das Display negativ beeinflussen würde. Die Auflösung des Digitizers beträgt mehr als 600 dpi und wird mit einer Rate von über 100 Hertz abgetastet.

Sowohl elektrostatische Digitizer als auch elektromagnetische Digitizer reagieren nur auf spezielle Stifte. Dies hat zum einen den Vorteil, dass nicht jede Berührung vom Bildschirm interpretiert wird, zum anderen

den Nachteil, dass sich der Tablett-PC ohne den Stift nicht mehr über den Bildschirm bedienen lässt.

3.2 Probleme des Digitizers

Viele Hardwarekomponenten in einem Tablett-PC erzeugen ihre eigenen elektromagnetischen Felder. Damit diese Felder das elektromagnetische Feld des Digitizers später nicht stören, wird gemessen welche Veränderungen die einzelnen Hardwarekomponenten hervorrufen. Anhand dieser Werte wird das elektromagnetische Feld eines Digitizerprototyps für die vorliegende Hardware kalibriert. Wird ein Tablett-PC in einer Produktionsreihe hergestellt, erhält er immer die gleiche Hardware und einen Digitizer, der mit Hilfe der Werte des Prototyps genau auf diese Hardware abgestimmt ist. Wird später eine Hardwarekomponente ausgetauscht, weil diese z.B. nicht mehr funktionstüchtig ist, und durch eine ähnliche aber nicht identische Hardwarekomponente ausgetauscht, verändert sich auch das elektromagnetische Feld des Digitizers. In Folge dessen müsste der Digitizer neu kalibriert bzw. implementiert werden. Da dieser Aufwand zu kostenintensiv ist, müssen die Anwender die möglichen Abweichungen akzeptieren oder defekte Hardwarekomponenten nur durch identische neue Hardwarekomponenten ersetzen.

In der folgenden Grafik sind das elektromagnetische Gitter eines Digitizers und die Wirkung der umgebenden Hardware auf dieses Gitter dargestellt:

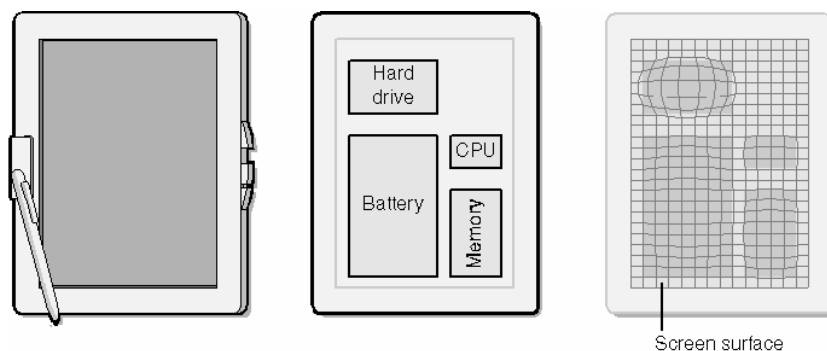


Abbildung 8: Elektromagnetisches Gitter und Hardwarekomponenten eines Digitizers

Ein weiteres Problem ist das Parallaxe - Problem. Durch die Schutzschicht, welche sich über dem Display befindet, erscheint der Cursor des Stiftes für den Benutzer an anderer Stelle, als dieser vom System interpretiert wird. Meist beträgt diese Abweichung nur ein paar Punkte. Dennoch wird diese Abweichung vom Benutzer als störend empfunden, weil dieser die gewollten Ziele nicht genau anvisieren kann. Da sich die Betrachtungswinkel der Benutzer unterscheiden, kann es passieren dass diese Abweichungen durch eine erneute Kalibrierung des Stiftes vermindert werden müssen.

Im Folgenden wird dieses Phänomen noch einmal graphisch veranschaulicht:

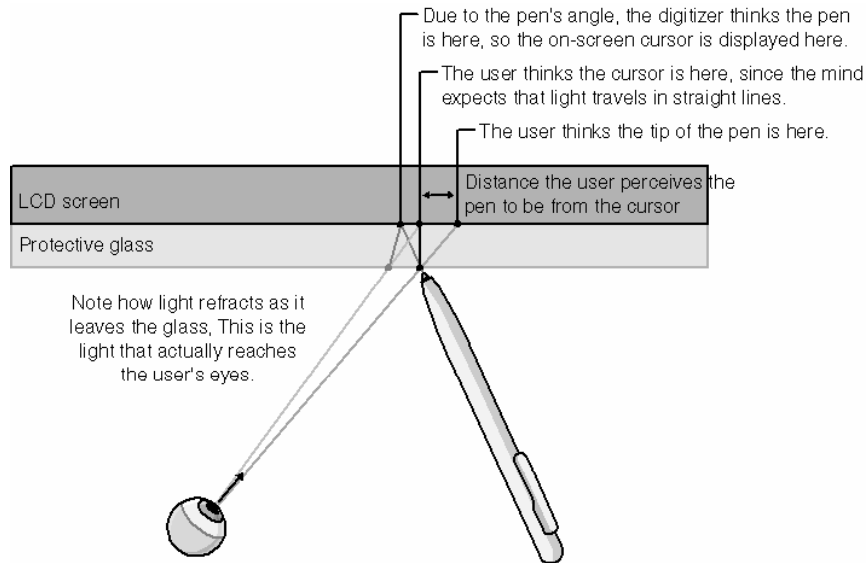


Abbildung 9: Parallaxe - Problem

4 Semantik des Stiftes

Der Stift eines Tablett-PCs dient als universelles Eingabegerät und kann diesen nahezu vollständig steuern. Dazu übernimmt der Stift die Funktionen von Maus und Tastatur, d.h. sowohl die Texteingabe als auch die Navigation erfolgen über ihn. Grundsätzlich wird der Benutzer den Tablett-Stift intuitiv wie einen normalen Kugelschreiber, Bleistift bzw. Marker benutzen wollen, um damit zu schreiben, zu zeichnen oder vorhandenen Text zu markieren. Das Navigieren mit dem Stift wird für Benutzer im Gegensatz dazu eher unnatürlich wirken. Im Folgenden wird zunächst die intuitive Benutzung des Tablett-Stiftes betrachtet und anschließend die Verwendung des Stiftes für die Navigation.

Sicherlich werden die meisten Benutzer glauben, dass die Benutzung des Tablett-Stiftes, der eines normalen Schreibgerätes vollkommen entspricht. Schnell wird sich jedoch herausstellen, dass dies ein großer Irrglaube ist, da der Tablett-PC mittels programmierter Routinen, die zur Interpretation dessen dienen, was mit dem Stift vollbracht wird, kein „Gefühl“ dafür entwickeln kann, was wir eigentlich mit dem Stift erreichen wollen. So steht die begrenzte Nutzbarkeit des Stiftes unserer ursprünglichen Denkweise, den Tablett-Stift z.B. als Kugelschreiber zu benutzen, im Wege, weil die Handhabung von verschiedenen Faktoren abhängt, welche beim Kugelschreiber in unserer Hand nicht vorhanden sind. So wird z.B. der Stift des Tablets als Mauszeiger, Punkt und bedingt als Stift auf dem Monitor angezeigt, dadurch wird dem Benutzer zwar visualisiert, wo er sich auf dem Bildschirm befindet, aber nicht was er genau tun soll. Auf einem Blatt Papier kann man mit einem Kugelschreiber an beliebiger Stelle schreiben, egal in welche Richtung, wie schräg oder wie unleserlich dies erfolgt. Bei einem Bildschirm ist dies nicht möglich. Der Benutzer muss in bestimmte Textfelder schreiben oder spezielle Anwendungen wie das *Tablet PC Input Panel*, welche im Softwareumfang des Betriebssystem *Windows XP for Tablet PCs* enthalten ist, nutzen um Text zu erzeugen, somit ist die beschreibbare Fläche stark begrenzt. Erkennen kann der Benutzer diese Flächen durch die Veränderung des Mauszeigers in einen kleinen Punkt, auch dies ist gewöhnungsbedürftig. Für eine optimale Handschriftenerkennung sollte möglichst deutlich, quasi in Schönschrift, geschrieben werden. Zwar werden auch weniger lesbare Schriftzüge erkannt, wobei jedoch die Fehlerquote deutlich höher ist. Wird etwas falsch erkannt, ist die Korrektur des schon geschriebenen Textes etwas aufwendiger. So werden zum Beispiel in der Anwendung *Tablet PC Input Panel* durch „Übermalen“ bis zur Unkenntlichkeit, fehlerhafte Buchstaben oder ganze Worte gelöscht. Auf normalem Papier wäre dies nicht möglich und nur ansatzweise mit dem Wegradieren von Bleistift oder Löschen von Tinte zu vergleichen, weil bei diesen Tätigkeiten kein „Übermalen“ stattfindet.

Die Erkennung des geschriebenen Textes erfolgt online, d.h. während des Schreibens. Hoher Speicher – und Zeitverbrauch, sowie die starke CPU – Last zeigen, dass die Erkennung der geschriebenen Handschrift alles andere als trivial ist. Dies führt dazu, dass die Übersetzung der geschriebenen Schrift verzögert stattfindet. Besonders anstrengend für den Benutzer ist das ständige Korrigieren eines Zeichens, wenn dieses nicht richtig erkannt wird. Meist ist es dann notwendig eine virtuelle Tastatur zu nutzen. Selbst das Bedienen der virtuellen Tastatur per Stift ist wesentlich zeitraubender als das so genannte „Ein-Finger-Suchsystem“, welches zur Texteingabe über die Tastatur oft angewendet wird, jedoch den Vorteil hat, dass der Benutzer schon aus Gewohnheit die Lage der Tasten kennt. Beim Bedienen der virtuellen Tastatur neigt der Benutzer dazu länger nach den Zeichen zu suchen. Gerade Benutzer, welche das „Schreibmaschinenschreiben“ beherrschen, benötigen deutlich mehr Zeit, wenn sie die virtuelle Tastatur über den Tablett-Stift bedienen müssen. Letztendlich lohnt sich das Schreiben per Stift mittels Handschrift oder virtueller Tastatur für begrenzte Texteingaben, wie Notizen,

Internetadressen, das Ausfüllen von Formulardaten u. ä. Für das Schreiben von längeren Texten oder z.B. das Programmieren von Anwendungen sind beide Verfahren eher ungeeignet, aber nicht unmöglich.

Dennoch hat der Tablett-Stift in Bezug auf das Markieren von Text bzw. Zeichnen einige Vorteile. So kann in einigen Anwendungen der Stift als effektiver Markerersatz dienen, indem der Text durch berühren des Stiftes hervorgehoben wird. Diese Funktion muss jedoch meist erst über ein Kontextmenü aufgerufen werden. Gleiches gilt für das Unter- oder Durchstreichen. Notizen deren Text nicht interpretiert wird, sondern einfach als Bild angesehen wird, bewirken ein natürlicheres Gefühl, als solche, welche den Text interpretieren. Auch Zeichnungen lassen sich wesentlich genauer erstellen als mit einer Maus. Grund dafür ist vor allem, dass sich der Stift präziser vom Benutzer führen lässt und ergonomischer in der Hand liegt. Dass dieser Vorteil schon erkannt wurde, zeigen die oft für CAD - Anwendungen verwendeten externen Grafik-Tabletts.

Die Navigation mit dem Tablett Stift ist ebenfalls nicht gerade einfach. Teilweise muss ein Benutzer ganz neu lernen mit einem Stift umzugehen, gerade dann, wenn er sehr an eine Maus gewöhnt ist. Auffällig ist dies vor allem dann, wenn der Benutzer Dinge nur markieren möchte, aber er durch den Druck auf das Display ungewollt einen Mausklick auslöst, z.B. wenn Dateien durch einmaliges antippen sofort geöffnet werden, obwohl sie nur markiert werden sollten.

Bewegt der Benutzer sich durch größere Dokumente, welche mehrere Seiten lang sind, ist es oftmals Notwendig mittels scrollen zu verschieben Seiten zu navigieren. Gerade hier zeigt sich ein großer Unterschied zu Tastatur und normaler Maus. Während mit Hilfe von Bild-Auf- und Bild-Ab-Taste der Tastatur bzw. dem Scrollrad der Maus einfach gescrollt werden kann, ist per Stift nur das Scrollen per Scrollleiste möglich. Diese Möglichkeit des Scrollens führt jedoch oft dazu, dass schneller durch das Dokument gescrollt wird, als dies vom Benutzer erwünscht ist. Gerade Personen, die oft das Scrollrad der Maus benutzen müssen hier stark umdenken. Versierte Benutzer verwenden oft Funktionstasten der Tastatur oder so genannte Shortcuts, um das Aufrufen von Befehlen über aufwendige Menüstrukturen zu vermeiden. Ein Tablett-Stift unterstützen diese Funktionen nicht. Passive Stifte (keine eigene Stromversorgung) besitzen nur bedingt Zusatzfunktionen, wie z.B. den rechten Mausklick, welche nur über Kontextmenüs benutzt werden können. Aktive Stifte unterstützen diese Funktion meistens durch eine Zusatztaste am Stift, welche dann während des Klicks bedient werden muss. Erleichternd wirkt sich diese Zusatztaste jedoch nur dann aus, wenn man ihre Handhabung beherrscht, andernfalls wird die Bedienung durch sie sogar noch erschwert.

Zusammenfassend lässt sich sagen, dass der Tablett-Stift eine positive Neuerung zu den klassischen Eingabegeräten Maus und Tastatur darstellt. Zurzeit muss jedoch das Denken der Benutzer, wie mit dem Stift umgegangen wird, an die Funktionsweise des Stiftes angepasst werden. Idealerweise könnte der Stift in seiner Handhabung in nicht so ferner Zukunft so weit optimiert werden, dass es irgendwann keinen Unterschied macht, ob man einen „normalen“ Stift oder den eines Tablett-PCs in der Hand hält und sich die Handhabung des Tablett-Stiftes somit an den Benutzer anpasst und nicht umgekehrt.

5 Fazit zum Tablett -PC

Ein Tablett-PC eignet sich sehr gut, wenn oft Notizen gemacht, Dokumente korrigiert oder wichtige Textpassagen markiert werden müssen. Wird der Tablett-PC als Terminplaner genutzt zeigt sich, wie komfortabel und schnell Termine verschoben werden können. Auch im medizinischen Bereich kann sich ein Tablett-PC positiv auswirken. So können z.B. Patientenakten leichter verwaltet und dadurch Zeit gespart werden, denn die Daten müssen nicht nachträglich per Hand in einen Rechner eingespeist werden, sondern können, wenn dies erforderlich ist, auf andere PCs per Netzwerk übertragen werden.

Interessant ist ein Tablett-PC auch für Benutzer, welche die speziell für Tablett-PCs entwickelten Spiele testen und spielen wollen. Gerade hier zeigt sich wie gut sich die neuen Steuerungsmöglichkeiten auswirken können. Beispielhaft ist hier das Spiel „Hexic Deluxe“ von MSN zu nennen. In der Version für den normalen PC, werden die Bausteine durch anklicken mit der Maus gedreht, ein zusätzlicher Schalter gibt die Richtung der Drehung an. Deutlich komfortabler geht dies in der Tablett-Version, in dieser wird per Stift einfach die Drehung auf die Bausteine gemalt, die sich dann entsprechend drehen. Ein weiterer Vorteil stellt das Schreiben mit dem Stift für Spiele dar, in denen Buchstaben eingetragen werden, wie z.B. in dem für dieses Seminar entwickelte Spiel „Mystical Ink“.

Tablett-PCs ohne integrierte Tastatur und Maus eignen sich nicht für jeden Benutzer. So gestaltet sich das handschriftliche Schreiben längerer Texte für geübte Tastaturschreiber als wesentlich langsamere und aufwendigere Eingabemöglichkeit. Auch das Umsteigen auf das Schreiben über die virtuelle Tastatur, stellt eine eher unhandliche Alternative dar. Ähnlich verhält sich dies beim Programmieren.

Ein weiteres Problem ist, dass nicht so leistungsstarke Tablett-PCs schnell an ihre Grenzen stoßen, infolge dessen ist bei solchen Endgeräten mit langen Ladezeiten bei speicherintensiven Anwendungen zu rechnen.

Abschließend kann gesagt werden, dass der Tablett-PC für viele Bereiche geeignet ist. In einigen Bereichen kann ein Tablett-PC die Arbeit wesentlich erleichtern und es gibt viele gute Anwendungen, die den Tablett-Einsatz unterstützen. Das der Tablett-PC nicht für jeden Bereich geeignet ist, wurde bereits herausgestellt. Wie schon erwähnt, gehören dazu sehr schreibintensive Aufgaben, z.B. das Programmieren. Auch das Scrollen ist beim Tablett-PC eher aufwendig. Die Handschrifterkennung ist nicht immer optimal und führt deshalb öfter zu Problemen, wie z.B. beim Erkennen von Sonderzeichen oder Klammern.

B Entwicklung des Spiels

1 Entwicklungsumgebung

Als Entwicklungsumgebung für das Projekt wurde das *MS Visual Studio .NET 2003* in Verbindung mit der *MS Tablet PC SDK 1.7* verwendet. Um die SDK in Visual Studio .NET nutzen zu können, muss diese in das jeweilige Projekt, mittels eines Verweises und einer `using`-Anweisung, eingebunden werden. Bei der Installation von Visual Studio .NET muss darauf geachtet werden, dass das Framework von .NET vorher installiert wurde. Die implementierte Software, welche die Tablet SDK nutzt, lässt sich nur auf einem Tablet-PC ausführen bzw. testen. Für dieses Projekt stand ein HP Compaq TC1000 HP zur Verfügung, auf welchem der Prototyp des Spiels *Mystical Ink* entwickelt und getestet wurde.

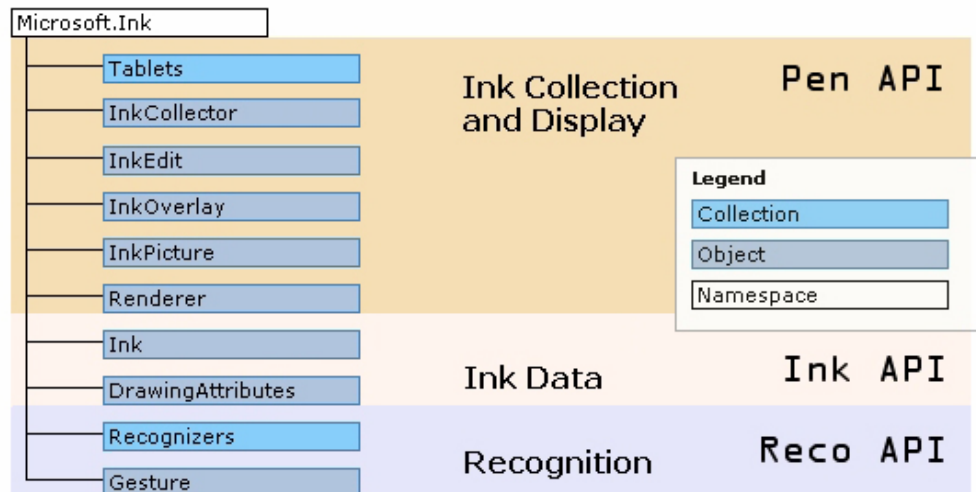


Abbildung 10: Aufbau der Klassenbibliothek Microsoft.Ink

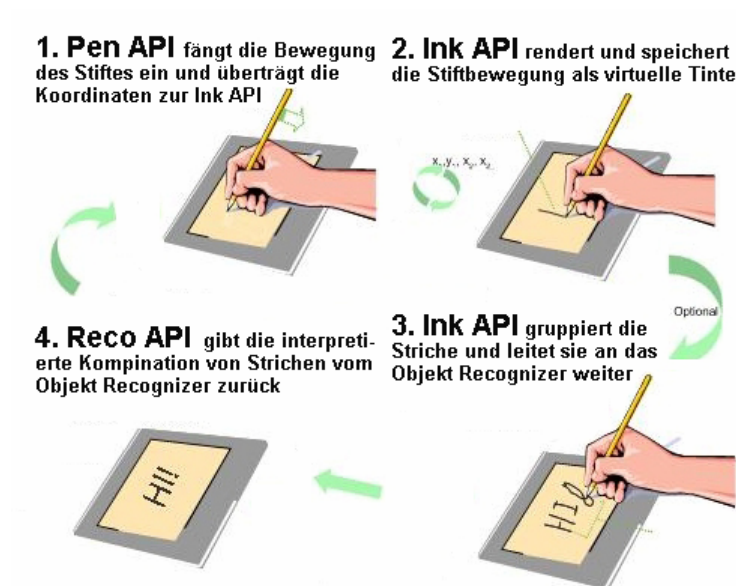


Abbildung 11: Pen -, Ink - und Reco API

noch einmal verdeutlichen

Abbildung 10 zeigt einen reduzierten Aufbau der Klassenbibliothek *Microsoft.Ink* des SDKs für .NET. im Folgenden werden die drei APIs Pen, Ink und Reco der Klassenbibliothek *Microsoft.Ink* etwas näher erläutert. Die Pen API fängt die Bewegungen des Stiftes ab und gibt die Koordinaten weiter an die Ink API. Mit Hilfe der Ink API werden diese Bewegungen in virtuelle Tinte (Ink-Objekt) umgewandelt und im Eingabebereich sichtbar. Die Reco API interpretiert die von der Ink API gelieferten Kombinationen von Tintenstrichen und wandelt diese in Text um. Abbildung 11 soll dies

2 Realisierung

Ziel des Projektes sollte der Prototyp eines Spiels, für Tablett-PCs mit graphischem Stifteinsatz, sein. Dazu wurde zunächst einmal überlegt, welche Grundanforderungen das Spiel erfüllen sollte. Zur Auswahl standen z.B. die Handschriftenerkennung, das Übernehmen der Maussteuerung bzw. der Steuerung der Pfeiltasten durch den Stift oder die Steuerung durch Zeichnen von Objekten, wie z.B. im Spiel InkBall. Außerdem war es wichtig das Genre für das Spiel festzulegen. Etappenziel war zunächst die Funktionen des Stiftes zu erforschen und zu testen. Deshalb wurde die o.g. Entwicklungsumgebung genutzt um zwei Testprogramme zu schreiben. Verwendet wurde dazu die Programmiersprache C# (CSharp).

3 Testprogramme

3.1 InkAnwendung

Mit dem Programm InkAnwendung soll die Funktionsweise, der drei APIs, der Klassenbibliothek *Microsoft.Ink*, etwas näher erläutert werden. Dazu wurde ein Formular erstellt, welches einen weißen Eingabebereich für die Handschrift beinhaltet und nur in diesem kann mit dem Stift geschrieben bzw. gezeichnet werden. Der Löschen-Button dient dazu diesen Eingabebereich zu löschen, da das "Übermalen" der falsch geschriebenen Zeichen um sie zu löschen, nicht im Funktionsumfang enthalten ist, sondern erst implementiert werden müsste. Weiterhin sollte getestet werden, ob man die Eingabe von Zeichen auf ein Zeichen reduzieren bzw. begrenzen kann. Weder Zeichen noch Zahlen sind für die Eingabe zugelassen. Eine Schwierigkeit stellte das Pipe-Zeichen (|) dar. Jeder gerade, senkrechte Strich wurde als Pipe-Zeichen und nicht als I interpretiert. Da Zeichen nicht zugelassen werden sollten, wurde eine kleine Anweisung geschrieben, welche dieses Zeichen in den Buchstaben I umwandelt. Wie die Handschriftenerkennung umgesetzt wurde, wird im Abschnitt B4.3 näher erläutert, da dieser Quelltext fast vollständig für den Prototypen des Spiels verwendet wurde. Ein Video, das die Funktionen dieses Testprogramms veranschaulicht und der Quelltext des Programms stehen auf der im Vorwort genannten Webseite zur Verfügung.

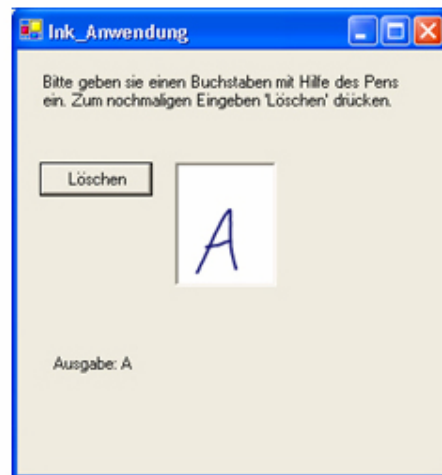


Abbildung 12: InkAnwendung

3.2 TablettAPI

Das Programm TablettApi versucht die Information über die Position des Cursors und die Richtung der Bewegung abzufangen. Somit fungiert der Stift als Mausersatz. Beim Antippen des Bildschirms (klicken) mit dem Stift und Entfernen des Stiftes vom Bildschirm werden Daten ausgewertet. Diese Daten sind die Grundlage zur Berechnung der Richtung der Bewegung, die mit dem Stift erzeugt wurde. Benutzt werden in diesem Programm die Pen und Ink API, sowie die Windows Forms API. Auch hier sind bei der Implementation der Anwendung Probleme aufgetreten. So wird z.B. noch nicht der richtige Winkel der Richtung erkannt, sobald die Richtung der Bewegung nicht eindeutig verläuft. Wird ein Kreis gezeichnet, dann werden für die Berechnung nur die Endpunkte und nicht der reale Weg verwendet, dies führt jedoch zu verfälschten Ergebnissen. Auch schnelle Bewegungen führen zu verfälschten Ergebnissen. Wichtig ist hierbei, dass der Pfeil im Video nicht den Himmelsrichtungen entspricht, sondern schon eine Vorbereitung zur Rotation visualisiert.

Auch dieses Programm steht als Videoaufzeichnung zu Verfügung und verdeutlicht so noch einmal Funktionsweise und Probleme.

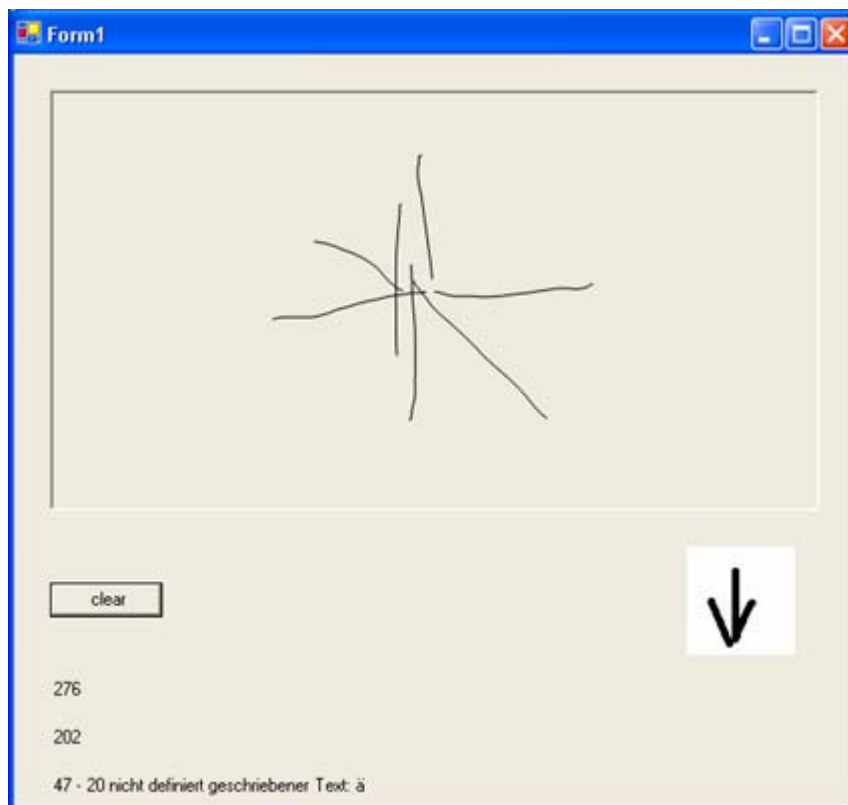


Abbildung 13: TablettAPI

4 Mystical Ink

4.1 Spielkonzept

Unser Spiel, welches sich in das Genre Denksport einordnen lässt, soll in Anlehnung an das Spiel Glücksrad entworfen werden und dennoch keine eins zu eins Umsetzung dieses Spiels darstellen. Entstanden ist diese Idee während der Entwicklung der Testprogramme. Es wurde eine Spielidee gesucht, die sowohl das Schreiben von Handschriftlichen Buchstaben als auch die Steuerung durch den Tablett-Stift miteinander vereint. Die handschriftlichen Komponenten des Spiels sollen durch die Eingabe der zu erratenden Buchstaben und die Eingabe der Lösung selbst, realisiert werden. Ein Weiteres Feature soll die Drehung des Rades durch das Zeichnen einer Linie auf dieses, darstellen. Dazu soll die Rotation des Rades abhängig von Druckstärke, Länge der Line, Rotation des Stiftes um seine eigene Achse, Neigungswinkel beim Zeichnen sowie die Zeit, welche für die Zeichnung der Linie benötigt wird, berechnet werden. Weiterhin wurde festgelegt die Daten für das Spiel in eine XML-Datei auszulagern. Dies hat den Vorteil, dass diese Daten den Quelltext nicht zusätzlich unübersichtlich gestalten und auch einfacher zu verwalten sind.

Ein Weiteres Highlight sollte die Spielereinstellung darstellen. Abhängig von der Anzahl der Spieler sollen entsprechend viele Felder für die Spieler eingeschaltet werden. Besonders daran, soll die Eingabe des Namens per Handschrift sein, der dann auch im weiteren Spiel dazu benutzt werden soll, um den Spieler direkt anzusprechen.

Um zu verhindern, dass das Spiel langweilig wird oder zu schnell vergeht, wurde beschlossen, das drei Runden pro Spiel gespielt werden und die entsprechende Kategorie sowie der dazugehörige Begriff zufällig pro Runde ausgewählt werden sollen, ohne das ein Begriff dabei doppelt vorkommt. Wie beim ursprünglichen Glücksrad soll es Felder geben, welche den Gewinn auf Null setzen (NULL), das Aussetzen (Niete) in einer Runde zur Folge haben und den Gewinn eines Extradrehs (Extra) beinhalten, welcher in Folge eines Aussetzers eingesetzt werden kann. Natürlich sollen auch Buchstaben erraten werden können, dies soll, wie schon erwähnt durch handschriftliche Eingabe, geschehen. Wird ein Konsonant angegeben und dieser gefunden, soll der erdrehte Wert mit der Anzahl der Vorkommen dieses Buchstaben multipliziert und gutgeschrieben werden. Im Unterschied zum richtigen Glücksrad sollen Vokale mit dem erdrehten Wert gekauft und dieser Wert einmalig abgezogen werden, egal ob der Vokal einmal, mehrmals oder gar nicht vorkommt. Dies hat zur Folge, dass sich die Spieler die Vokale nicht immer kaufen können. Ein Spieler soll so lange Spielen können, bis er entweder einen falschen Buchstaben rät, eine Niete erdreht und kein Extra einsetzen will bzw. kann oder einen falschen Lösungsversuch abgibt. Wird eine Lösung richtig abgegeben oder alle gesuchten Buchstaben gefunden, so dass der Begriff vollständig zu lesen ist, erhält der aktuelle Spieler einen Bonusbetrag. Gewinnen soll der Spieler, der nach diesen drei Runden die höchste Punktzahl erreicht hat.

4.2 Spielablauf

Im Folgenden soll der Spielablauf des Spiels näher erläutert werden. Beim Starten des Programms werden zunächst, alle Elemente des Formulars unsichtbar geschaltet und der Titel des Spiels, sowie der Start Button eingeblendet. Wird der Startbutton betätigt, wird der Dialog `Spielerabfrage` gestartet, welcher den Benutzer auffordert die Anzahl der Spieler anzugeben. Bricht der Benutzer diesen Dialog ab, wird das Programm beendet. Wurde die Spieleranzahl ausgewählt, werden zunächst die Komponenten des Startbildschirms entfernt. Danach wird die Funktion `label_spieler_sichtbar(...)` aufgerufen, der die Anzahl der Spieler übergeben wird. Anhand der Anzahl der Spieler werden die entsprechenden Label für Namen und Nummer der Spieler sichtbar, und die dazugehörigen `InkCollectoren` eingeschaltet und aktiviert. Nun haben die Spieler die Möglichkeit ihre Namen mit dem Stift einzugeben. Beim Betreten des Feldes wird der Text des Labels gelöscht. Nach Verlassen des Feldes, wird zunächst die Funktion `namen_eingeben(...)`, welcher der entsprechende `InkCollector` übergeben wird, gestartet. Diese Funktion überprüft die Eingabe des Benutzers. Wenn die Eingabe fehlerhaft oder leer ist, wird eine entsprechende Fehlermeldung im Hinweisfeld ausgegeben und die eingegebene „Tinte“ gelöscht. Ist die Eingabe gültig, dann wird eine `MessageBox` aufgerufen, die den interpretierten Text anzeigt und den Benutzer bittet diesen als richtigen Namen zu bestätigen oder abzulehnen. Die hier hinter liegende Handschriftenerkennung, wird im Abschnitt B4.3 *Handschriftenerkennung* näher erläutert. Wird der Text als Name abgelehnt, wird die „Tinte“ gelöscht und der Anwender hat die Möglichkeit den Namen erneut einzugeben. Ist der Name jedoch bestätigt worden, werden das Feld für den Namen sowie der entsprechende `InkCollector` deaktiviert und die Funktionen, welche ausgelöst werden, wenn das Textfeld betreten oder verlassen wird, ausgeschaltet. Anschließend daran, wird die Funktion `starte_spiel()` aufgerufen. Dort wird zuerst überprüft, ob die Anzahl der Spieler mit der Anzahl der eingegebenen Spieler übereinstimmt. Ist dies der Fall wird die Funktion weiter ausgeführt, wenn nicht, dann bricht sie ab und die Prozedur des Eingebens der Spielernamen wird solange wiederholt, bis alle Spielernamen eingegeben wurden. Tritt Letzteres ein, werden die restlichen Elemente, welche für das Spiel benötigt werden, geladen. Ausgenommen davon sind lediglich, die Elemente für Begriff und Kategorie. Im Anschluss daran wird die Funktion `runde_starten()` aufgerufen. Diese beinhaltet den Aufruf der Funktion `label_einschalten()`, welche die vorher ausgeschalteten restlichen Elemente für Begriff und Kategorie lädt und initialisiert. Dazu werden zunächst durch Aufrufen der Funktion `ermittle_zeileninhalt()` ein Begriff und die zugehörige Kategorie zufällig aus einer XML-Datei ausgelesen. Im Abschnitt B4.5 *Datei auslesen* wird dies näher beschrieben. Kategorie und Rundenzahl werden in diesem Zusammenhang in das Label im oberen Spielbereich eingetragen. Entsprechend des Begriffs, werden die Felder für die Buchstaben ohne Inhalt angezeigt. Nun wird die Funktion `spieler_starten()` aufgerufen, sie sorgt unter anderem durch Aufrufen der Funktion `rad_einschalten()` (s. Abschnitt B4.4 *Steuerung des Rades*) und aktivieren der Methode die beim Klicken des Button „Lösungsversuch“ aufgerufen wird, dass der erste Spieler seinen Zug beginnen kann. In diesem Zustand kann der Spieler nur versuchen die Lösung abzugeben oder das Rad zu drehen. Wird das Rad gedreht, kommen folgende Möglichkeiten in Betracht: Der Pfeil zeigt auf einen Zahlenwert, Null, Extra oder Niete. Null führt dazu, dass das Punktekonto des Spielers gelöscht wird und die Funktion `naechster_spieler()` abgearbeitet wird. Beim Erdrehen von Niete, muss der Spieler aussetzen und der nächste Spieler ist dran, es sei denn er kann ein Extra einsetzen, dass er erhält, wenn der Pfeil auf das entsprechende

Feld zeigt. Zeigt der Pfeil auf einen Zahlenwert, wird die Funktion `eingabe_einschalten()` aufgerufen. Sie aktiviert den `InkCollector` und das `MouseUp-Handle` für das Eingabefeld, sowie die Buttons Löschen und Senden. Das Rad wird durch Ausführen von `rad_ausschalten()` wieder deaktiviert. In diesem Zustand ist es nicht möglich einen Lösungsversuch zu tätigen, es kann nur ein Buchstabe gewählt werden. Die Handschriftenerkennung selbst, wird im Abschnitt B4.3 *Handschriftenerkennung* näher erläutert. Wird ein Buchstabe erkannt und vom Benutzer abgesendet, wird die Eingabe der Buchstaben deaktiviert und der Buchstabe der Funktion `finde_buchstaben(...)` übergeben und diese ausgeführt. Hierbei wird zunächst überprüft, ob der Buchstabe schon einmal gesendet wurde. Ist dies der Fall, bricht die Funktion ab, der Spieler bekommt keine Punkte und `naechster_spieler()` wird aufgerufen. Wurde der Buchstabe noch nicht verwendet, wird geprüft, ob der Buchstabe ein Vokal oder Konsonant ist. Wurde ein Vokal gewählt, wird geprüft, ob der Spieler den Vokal „kaufen“ kann. Ist der Punktestand zu niedrig, wird die Funktion abgebrochen und der nächste Spieler ist am Zug. Kann der Vokal bezahlt werden, wird unabhängig davon, ob der Vokal einmal, mehrfach oder gar nicht vorkommt, einmalig der erdrehete Zahlenwert vom Punktekonto des Spielers abgezogen. Wurde ein Konsonant gewählt und kommt dieser nicht vor, wird die Funktion ebenfalls abgebrochen und der nächste Spieler ist am Zug, andernfalls wird der Zahlenwert mit der Anzahl der Vorkommen multipliziert und dem Spieler gutgeschrieben. Wird ein Buchstabe gefunden, dann wird er in einem temporären String für die Lösung eingefügt und das entsprechende Label bekommt den Buchstaben zugewiesen, zeigt ihn an und wird für Benutzereingaben abgeschaltet. Bevor die Funktion `finde_buchstaben()` verlassen wird, werden die temporären Lösungsstrings mit den Strings verglichen, welche den Begriff enthalten, stimmen beide überein, d.h. alle Buchstaben des Begriffs wurden gefunden, dann erhält der Spieler Bonuspunkte für die Lösung und eine neue Runde wird durch Aufrufen von `runde_starten()` begonnen, andernfalls ist der aktuelle Spieler, wieder am Zug und das Rad bzw. der Button „Lösungsversuch“ werden wieder aktiviert. Wird der Button Lösungsversuch betätigt, werden für alle Buchstabenfelder (64 Stück), die Funktionen die beim Verlassen und Betreten des Feldes ausgelöst werden, mittels der Funktion `label_aktivieren()`, aktiviert. Beschreibbar sind dann jedoch nur die Label, welche noch keinen Buchstaben enthalten, sichtbar und nicht für Benutzereingaben gesperrt sind. Der Spieler kann nun die fehlenden Buchstaben handschriftlich eintragen. Betritt der Spieler ein Feld, wird dessen Inhalt, sofern vorhanden, gelöscht und ermöglicht somit Korrekturen. Wird das zuvor betretene Feld verlassen, wird der Buchstabe interpretiert und eingefügt. Hat der Spieler die Eingabe beendet, kann er durch Klicken des modifizierten Buttons „Lösungsversuch“, seine Lösung überprüfen. Diese wird auf Richtigkeit getestet und der Spieler erhält Bonuspunkte, wenn die Lösung richtig ist und es wird eine neue Runde gestartet. Ist die Lösung falsch, ist der nächste Spieler dran. Wurden alle drei Runden durchlaufen, wird eine `MessageBox` aufgerufen, mit der das Spiel entweder neu gestartet oder beendet werden kann. Außerdem enthält sie eine Zusammenfassung der erspielten Punkte.

4.3 Handschriftenerkennung

Wie schon erwähnt, wird die Handschriftenerkennung mehrfach eingesetzt. Da sich die Handschriftenerkennung der Eingabe von Spielernamen, Buchstaben im Eingabefeld bzw. Lösungsbuchstaben sich kaum unterscheiden, wird hier beispielhaft nur die Handschriftenerkennung für das Eingabefeld zwischen Rad und Spielernamen, erläutert.

Nach dem Erstellen der Variable vom Typ `InkCollector`, kann der `InkCollector` durch Zuweisen des Steuerelementes initialisiert werden. Dieser muss dann nur noch aktiviert werden. Nur wenn der `InkCollector` aktiviert ist, kann auf dem Steuerelement geschrieben werden. Die Farbe der „Tinte“ lässt sich frei auswählen. Folgender Quelltextauszug, veranschaulicht dies noch einmal:

```
//Initialisieren eines neuen InkCollectors und diesem das Panel
//für die Eingabe der einzelnen Buchstaben zuweisen
InkCollector_eingabe = new InkCollector(this.panel_eingabe.Handle);
//Farbe der "Tinte" für das Buchstabeneingabefeld auf Blau setzen
InkCollector_eingabe.DefaultDrawingAttributes.Color = Color.MidnightBlue;
//InkCollector_eingabe aktivieren, schreiben möglich
InkCollector_eingabe.Enabled = true;
```

Abbildung 14: Initialisieren und aktivieren des InkCollectors

Die Interpretation der eingegebenen Tinte in maschinellen Text erfolgt durch die Anweisung `InkCollector_eingabe.Ink.Strokes.ToString()`. Ist die Eingabe des Textes durch den Benutzer zu schnell erfolgt oder kann das eingegebene Zeichen nicht interpretiert werden, dann werden Fehlermeldungen ausgelöst, diese sollten abgefangen werden. Durch Ausführen der Methode `InkCollector_eingabe.Ink.DeleteStrokes()` werden die gezeichneten Tintenstriche gelöscht.

Da oft das Pipe-Zeichen statt des Buchstaben I, die Null statt des Buchstaben O und eine zwei statt des Buchstaben Z erkannt wurden, sind diese Zeichen abgefangen und dem entsprechenden Buchstaben zugewiesen worden. Ursache für die falsche Erkennung ist entweder unsauberes Schreiben bzw. das Anheben des Stiftes. `MouseUp` bedeutet, dass beim Loslassen des Klicks die Funktion ausgeführt wird. Schreibt man jedoch z.B. ein H, wird dieses in drei einzelnen Strichen gezeichnet. Jedes Mal, wenn einer dieser Striche gesetzt wurde, wurde die Funktion `panel_eingabe_MouseUp()` ausgelöst. Somit wurde die Interpretation dreimal durchgeführt. Nachteil dieser Art des Startens der Interpretation ist, dass diese mehrmals gestartet wird. Dieser Nachteil steht aber dem Vorteil gegenüber, dass der Anwender überblicken kann, wie seine Eingaben umgewandelt werden.

Der nachfolgende Quelltextauszug, veranschaulicht das eben geschriebene noch einmal:

```

private void panel_eingabe_MouseUp(object sender,
                                   System.Windows.Forms.MouseEventArgs e)
{
    //Testen ob Eingabe erfolgt oder leere Eingabe
    if (InkCollector_eingabe.Ink.Strokes.ToString() == null |
        InkCollector_eingabe.Ink.Strokes.ToString()=="")
    {
        //Löscht Eingabe der Schrift via Pen
        InkCollector_eingabe.Ink.DeleteStrokes();
        //Ausgabe Error
        ...
    }
    else
    {
        //String mit umgewandelter Eingabe initialisieren und in
        //Großbuchstaben umwandeln
        string helpText=
            InkCollector_eingabe.Ink.Strokes.ToString().ToUpper();
        //Testen ob mehr als ein Zeichen erkannt wird
        if (helpText.Length>1)
        {
            //Ausgabe Error
            ...
        }
        else
        {
            if(Regex.IsMatch(helpText, "[A-Z]"))
            {
                //Ausgabe der Eingabe
                label_ausgabe.Text = "Ausgabe: " + helpText;
                ...
            }
            else
            {
                if(Regex.IsMatch(helpText, "[|]") |
                    Regex.IsMatch(helpText, "[0]")
                    | Regex.IsMatch(helpText, "[2]"))
                {
                    if (Regex.IsMatch(helpText, "[|]"))
                    {
                        //"helpText" auf "I" setzen
                        helpText="I";
                    }
                    ...
                    //Ausgabe der Eingabe
                    label_ausgabe.Text = "Ausgabe: "+ helpText; ...
                }
                else
                {
                    //Ausgabe Error
                    ...
                }
            }
        }
    }
}

```

Abbildung 15: Interpretation der Handschrifteneingabe

4.4 Steuerung des Rades

Für die Rotation des Rades und die Gewinnberechnung sind fünf Funktionen und eine weitere zusätzliche Klasse verantwortlich. Diese Klasse, mit dem Namen `Utilities`, realisiert die eigentliche Rotation des Rades. Ihr werden das Bild und der Rotationswinkel für die Drehung des Bildes übergeben. Zurückgeliefert wird das gedrehte Bild, um den angegebenen Winkel. Entnommen wurde der Quellcode der Klasse `Utilities`, dem Artikel *Image Rotation* in .Net von der Code-Projekt-Webseite¹. Auf diese Klasse wird hier nicht weiter eingegangen, denn die genauere Informationen zu allen Funktionen der Klasse sind auf der Webseite selbst und in dem dort zur Verfügung gestellten Quelltext zu finden.

Beim Aufrufen der Funktion `rad_einschalten()` werden zwei Funktionen aktiviert die für das sammeln und starten der Rotation benötigt werden. Die Funktion `myInkCollector_wheel_NewPackets(...)` wird immer dann aufgerufen, wenn sich der Stift auf dem Display befindet und ein Klick ausgeführt wird. Solange der Klick anhält, wird diese Funktion immer wieder aufgerufen und dabei jedes Mal die Funktion `UpdateOutput(...)` ausgelöst, welche die Daten ausliest und verarbeitet. In dieser Funktion werden die Klick-Zeit und die „zurückgelegte“ Strecke berechnet, welche dann in der Rotationswinkelberechnung weiterverwendet werden.

```
private void UpdateOutput(int count, int [] data)
{
    int length = data.GetLength(0);
    int offset = length * (count-1);
    int i =0;
    int [] arraycount = new int[4];
    for (int n = 0; n < length; n++)
    {
        arraycount[i] = data[offset+n];
        i++;
    }
    ...
}
```

Abbildung 16: UpdateOutput - Sammeln der Daten

Wenn sich der Stift über dem Bildschirm befindet, wird kontinuierlich die Funktion `myInkCollector_wheel_NewInAirPackets(...)` aufgerufen. Wurden Daten gesammelt, löst diese Funktion die Rotation des Rades aus und der Rotationswinkel wird berechnet. Anschließend wird die Funktion `rotated(...)` aufgerufen. Ihr wird der Rotationswinkel übergeben, um den sich das Rad drehen soll. In der Funktion selbst, wird der Ausgangswinkel bestimmt und die Rotation des Rades „simuliert“. Dazu wird mehrfach eine Drehung des Bildes aufgerufen bis der „Endwinkel“ erreicht wurde. Für den Anwender scheint es so, als würde sich das Rad „natürlich“ drehen. Danach wird die Funktion `getWert(...)` gestartet, dabei wird ihr der aktuelle Winkel, um den sich das Rad in Relation zur Grundstellung gedreht hat, d.h. die aktuelle Stellung, übergeben. Die Funktion gibt den erdrehten Wert aus.

¹ <http://www.codeproject.com/csharp/rotateimage.asp>

4.5 Datei auslesen

4.5.1 Aufbau der XML-Datei

Damit die Liste der Kategorien und der zugehörigen Begriffe nicht unübersichtlich direkt im Quellcode zu finden ist, wurde entschieden diese in eine XML - Datei auszulagern. Das hat vor allem den Vorteil, dass Begriffe und Kategorien sehr einfach und flexibel ausgetauscht, erweitert oder gelöscht werden können. Jeder Begriff und seine dazugehörige Kategorie werden einheitlich in einem Datensatz zusammengefasst. In Abbildung 17 ist Beispielhaft ein Datensatz dargestellt, der dies verdeutlichen soll.

```
<?xml version="1.0" encoding="utf-8" ?>
<root>
  <Datensatz>
    <Bezeichnung>Kategorie</Bezeichnung>
    <Kategorie>Personen</Kategorie>
    <Begriff>Albrecht Duerer</Begriff>
  </Datensatz>
  ...
</root>
```

Abbildung 17: XML Struktur

4.5.2 Grundlagen zum Auslesen der XML-Datei

Um in C# mit XML – Befehlen arbeiten zu können, müssen die beiden `using`-Direktiven `System.Data` und `System.Xml` in den Quellcode eingebunden werden. Zum Auslesen der Datei wurde die Klasse `XmlTextReader` verwendet. Dazu wurde zuerst ein Objekt dieser Klasse angelegt (s. Abbildung 18).

```
// Anlegen eines Objektes von XmlReader, zum Auslesen
// der XML-Datei
XmlTextReader XmlTR_reader = new XmlTextReader("DataFile.xml");
```

Abbildung 18: Klassenobjekt zum XML-Datei auslesen

Über eine `while`-Schleife wird die Datei dann geparkt (siehe Abbildung 19) und die gewünschten Daten können ausgelesen werden. Die ausgelesenen Daten einer Textzeile im XML – Dokument stehen in dem angelegten Objekt, abgefragt durch `XmlTR_reader.Value`.

```
// solange nicht das Ende der Datei erreicht, lese nächste Zeile
while (XmlTR_reader.Read())
{
  // wenn aktuelle Zeile Textelement enthält, dann weiter,
  if (XmlTR_reader.NodeType == XmlNodeType.Text)
  {
    // Verarbeitung der ausgelesenen Daten, aktuell
    // gelesenen Zeile ist in XmlTR_reader.Value enthalten
    ...
  } // sonst neue Zeile lesen
}
```

Abbildung 19: Schleife zum XML-Datei auslesen

4.5.3 Zufallszahl

Wie schon im Spielkonzept erwähnt, sollte für jede Runde zufällig ein Begriff erraten werden, ohne dass eine Kategorie oder ein Begriff doppelt ausgewählt wird. Um dies zu erreichen, wurde zunächst eine Variable vom Typ `DateTime` erstellt, in der die aktuelle Systemzeit abgelegt wird. Mit dem Millisekundenanteil dieser aktuellen Uhrzeit wird ein Objekt der Klasse `Random` erstellt und initialisiert, das die für das Spiel benötigte Zufallszahl darstellt. Damit der Wert dieser Zufallszahl die maximale Anzahl der vorhandenen Begriffe, die pro Kategorie zur Verfügung stehen, nicht überschreitet, wird die Methode `Next(...)` auf sie angewandt. In dieser Methode wird die untere Schranke durch Eins und die obere Schranke durch die Anzahl der Begriffe einer ausgewählten Kategorie festgelegt und eine weitere Zufallszahl erzeugt, deren Wert innerhalb dieser Schranken liegt. In Abbildung 20 wird die Funktion `erzeuge_zufallszahl()` verkürzt dargestellt.

```
private int erzeuge_zufallszahl( ... )
{
    // Anlegen der Variable vom Typ DateTime mit der aktuellen
    // Systemzeit
    DateTime DT_Zeit = DateTime.Now;
    // Zufallszahl auf Basis der Millisekunden erzeugen lassen
    Random Random_zufallszahl = new Random(DT_Zeit.Millisecond);
    // Zurückliefern der Zufallszahl, zwischen 1 und maximaler
    // Anzahl der Begriffe der gewählten Kategorie
    return Random_zufallszahl.Next( 1, int_nummer);
}
```

Abbildung 20: Erzeugung der Zufallszahl

4.5.4 Ablauf des Auslesens der XML-Datei

Im Folgenden soll der Ablauf des Auslesens der XML-Datei und das zufällige Bestimmen von Kategorie und Begriff grob erläutert werden.

Zu Beginn werden alle benötigten Variablen und Arrays deklariert. Diese werden initialisiert, bevor etappenweise mehrere Schleifen durchlaufen werden. Im ersten Schleifendurchlauf werden zunächst alle vorhandenen Kategorien aus der XML-Datei ausgelesen. Danach wird im zweiten Schleifendurchlauf, wie im Abschnitt B4.5.3 *Zufallszahl* beschrieben, eine Zufallszahl erzeugt, aus der sich die jeweilige Kategorie bestimmen lässt. Um zu verhindern, dass eine Kategorie mehrmals verwendet wird, durchläuft jede ausgewählte Kategorie eine Überprüfung. Dabei wird ermittelt, ob die gewählte Kategorie im aktuellen Spiel schon verwendet wurde. Trifft dies zu wird eine neue Kategorie gesucht, bis eine Kategorie gefunden wurde, die noch nicht benutzt wurde.

Beim Durchlaufen der dritten Schleife werden die Begriffe der zuvor bestimmten Kategorie aus der XML-Datei ausgelesen. Nun wird im vierten Schleifendurchlauf, wie schon bei der Bestimmung der Kategorie beschrieben, mittels Zufallszahl ein Begriff bestimmt und so lange getestet, ob dieser schon im Spiel verwendet wurde, bis ein unbenutzter Begriff gefunden wird. Zum Schluss wird der Begriff für die GUI so formatiert, dass er auf die entsprechenden Zeilen verteilt werden kann. Sowohl der formatierte Begriff als auch die entsprechende Kategorie werden dann in globalen

Variablen abgelegt. Um die jeweiligen Variablen im Spielablauf zu füllen muss an den entsprechenden Stellen im Quelltext die Routine `ermittle_zeileninhalt()` aufgerufen werden.

4.6 Probleme bei der Entwicklung

Bei der Entwicklung des Spiels sind einige Probleme aufgetreten. Zuerst trat das Problem auf, die richtige Konfiguration der Entwicklungsumgebung anzugeben. Dazu musste neben dem .Net-Framework auch die Tablet-SDK zusätzlich zum Visual-Studio installiert werden. Weiterhin musste darauf geachtet werden, dass in jedem Projekt der entsprechende Verweis auf die SDK gesetzt wurde. Erst dann war es möglich die *Microsoft.Ink-API* zu nutzen.

Schwierigkeiten ergaben sich auch, als versucht wurde die PEN-Informationen auszulesen. Es nicht möglich den Neigungswinkel, Druck und zusätzliche Parameter auszulesen. Dieses Problem liegt vermutlich am Treiber. Somit musste eine Änderung des Spielkonzeptes erfolgen, in der diese Informationen nicht mehr berücksichtigt werden. Deshalb werden nun nur noch die Koordinaten der Bewegung selbst und die Zeit, welche zum Ausführen der Bewegung benötigt wird, betrachtet. Auch in Hinblick auf die Zeit gab es Probleme, da diese auf Grund der Trägheit des Tablet manchmal den Wert 0 beträgt. Dieses Problem wurde behoben, indem dieser Fall abgefangen, ein fester Wert zugewiesen und in die Rotationsberechnung Zufallszahlen integriert wurden. Die Integration der Zufallszahlen, sollte außerdem bewirken, dass man das Rad schwerer durch Zeichnen gezielter Linien, manipulieren kann.

Ein weiteres Problem stellen der geringe Speicher (256 MB), sowie die niedrige Prozessorleistung (1 Ghz) des Tablet dar. Aufgrund des hohen Speicherverbrauchs und der langen Umwandlung der Handschriftenerkennung, welche auch in unserem Spiel mehrfach vorkommt, passiert es dass das Spiel für den Anwender träge erscheint.

4.7 Spielanleitung

Vorraussetzung für die Ausführung des Spiels ist das installierte Microsoft .Net Framework. Ist das Framework nicht installiert, lässt sich das Spiel nicht starten. Nachdem die ZIP-Datei *MysticalInk.zip* von der im Vorwort genannten Webseite heruntergeladen wurde, muss diese zunächst entpackt werden. Im nun entpackten Verzeichnis befinden sich dann folgende Dateien:

- *MysticalInk.exe*
- *DataFile.xml*
- *rad.png*

Fehlt eine dieser Dateien, ist das Spiel nicht mehr lauffähig. Das Spiel wird durch Ausführen der Datei *MysticalInk.exe* gestartet. Wenn der Startbildschirm erscheint, kann das Spiel, durch betätigen des Schließen-Buttons(X) im oberen rechten Fensterrand, beendet oder, durch Klicken des Start-Buttons, die Spielerabfrage gestartet werden. Wenn die Spielerabfrage gestartet wurde, erscheint ein Dialogfenster, das den Anwender auffordert, die Spieleranzahl durch Klicken des entsprechenden

Buttons anzugeben. Wird kein Button geklickt passiert nichts. Das Klicken des Schließen-Buttons führt zum Beenden des Spiels. Wurde die entsprechende Spieleranzahl ausgewählt, erscheinen die jeweiligen Felder für die Spielernamen.

Beim Eingeben der Spielernamen müssen folgende Dinge beachtet werden:

- das Schreiben ist erst möglich, wenn der Mauszeiger in einen kleinen Punkt umgewandelt wurde
- wird der Eingabebereich berührt, wird der Inhalt des Feldes gelöscht
- es sollte darauf geachtet werden, nicht über die weiße Eingabefläche hinaus zu schreiben, da sonst beim Wiedereintritt das Geschriebene gelöscht wird
- wird der weiße Eingabebereich verlassen, beginnt die Überprüfung des Namens
- wird das Fenster Spielernamen prüfen mit Yes beantwortet, kann der Name nicht mehr verändert werden

Erst wenn alle Spielernamen eingegeben wurden startet das eigentliche Spiel. Hier ist es wichtig zu warten bis alle Komponenten geladen wurden. Erkennbar ist dies am Hinweistext, der den ersten Spieler erst dann auffordert das Rad zu drehen, wenn das Laden beendet ist.

Das Rad wird durch das Aufzeichnen einer Linie gesteuert. Bevor versucht wird, diese Linie zu zeichnen ist darauf zu achten, dass der Zeiger sich in einen Punkt verwandelt, wenn er sich über dem Rad befindet. Der Punkt zeigt an, dass mit dem Zeichnen begonnen werden kann. Weiterhin ist zu Beachten, dass je länger die Linie ist und je kürzer die Zeit ist, die benötigt wird um diese Linie zu Zeichnen, desto schneller und länger dreht sich das Rad.

Nach dem Stillstand des Rades, wird je nach erdrehtem Wert, weiter verfahren. Wurde ein Extra erzielt, wird dieses dem Spieler gutgeschrieben. Wenn eine Niete oder Null erzielt wird, ist der Zug des Spielers beendet, bei Null wird jedoch noch zusätzlich das Punktekonto des Spielers gelöscht. Beim Erzielen eines Zahlenwertes, wird das Eingabefeld zwischen Rad und Spielernamen für die Eingabe eines Buchstabens aktiviert. In diesem Zustand kann weder das Rad gesteuert, noch der Lösungsversuchs-Button betätigt werden.

Bei der Eingabe des Buchstabens sind folgende Punkte zu beachten:

- das Schreiben ist erst möglich, wenn der Mauszeiger in einen kleinen Punkt umgewandelt wurde
- die Interpretation des Buchstabens erfolgt, beim Absetzen des Stiftes
- falsche Eingaben müssen mit dem Löschen-Button gelöscht werden
- erst wenn des Senden-Button betätigt wird, wird der Buchstabe im Lösungswort gesucht
- wird ein Konsonant gefunden, wird die Anzahl der gefundenen Buchstaben mit dem erdrehten Wert multipliziert und dem aktuellen Spieler gutgeschrieben
- Vokale kosten einmalig den erdrehten Wert, egal ob sie einmal, mehrmals oder gar nicht vorkommen
- wird ein Vokal gewählt, obwohl nicht genug Punkte vorhanden sind, dann wird der Spielzug des aktuellen Spielers beendet und der nächste Spieler ist dran
- wird ein Buchstabe nicht gefunden, ist der nächste Spieler dran

Wenn alle Buchstaben gefunden wurden, bekommt der Spieler, der den letzten Buchstaben gefunden hat, Bonuspunkte für die Lösung und eine neue Runde beginnt. Gleiches gilt bei Eingabe der Lösung mittels Lösungsversuch.

Die Lösung kann jederzeit eingegeben werden, es sei denn das Rad wurde gerade gedreht und ein Buchstabe muss eingegeben werden. Wird der Lösungsversuch-Button betätigt, werden die Felder schreibbar geschaltet. Wird dieser dann modifizierte Lösung abgeben-Button erneut betätigt und die eingegebene Lösung überprüft. Ist die Lösung falsch ist der nächste Spieler dran.

Folgende Punkte sind bei der Eingabe der Lösung zu beachten:

- Buchstaben nacheinander eingeben, d.h. warten bis die Überprüfung eines Feldes abgeschlossen ist, bevor ein neues Feld betreten wird
- das Schreiben ist erst möglich, wenn der Mauszeiger in einen kleinen Punkt umgewandelt wurde
- wird der Eingabebereich berührt, wird der Inhalt des Feldes gelöscht
- es sollte darauf geachtet werden, nicht über die weiße Eingabefläche hinaus zu schreiben, da sonst beim Wiedereintritt das Geschriebene gelöscht wird
- wird der weiße Eingabebereich verlassen, beginnt die Überprüfung der Eingabe und das Resultat wird in das entsprechende Feld geschrieben

Sind alle drei Runden abgeschlossen, erscheint ein Dialogfeld, das den Highscore enthält und abfragt, ob das Spiel nochmal gestartet oder beendet werden soll.

5 Fazit zum Prototypen des Spiels

Zusammengefasst kann gesagt werden, dass die gesetzten Ziele, welche das Spielkonzept beinhaltet, fast vollständig umgesetzt wurden. Nur technische Einschränkungen bzgl. des Auslesens aller Werte, die der Stift liefern könnte, haben dazu geführt, dass uns einige wertvolle Elemente zur Gestaltung des Spiels verwehrt blieben. Die im Spielkonzept geplanten Funktionalitäten wurden in allen anderen Punkten vollständig umgesetzt und erreicht. Dazu gehören die Steuerung des Rades, die Handschrifteneingabe der Buchstaben, das Auslesen der Begriffe und Kategorien aus einer XML-Datei und die gesamte Spielablaufsteuerung, einschließlich der Gewinnermittlung.

6 Ausblick

Der Prototyp Mystical Ink ist so angelegt, dass er einfach erweitert werden kann. So wäre eine denkbare Möglichkeit, das Spiel im "Mehrspielermodus" so zu erweitern, dass jeder Spieler einen eignen Tablett benutzt und die Kommunikation untereinander bzw. mit dem Spiel über LAN oder WLAN realisiert wird. Weiterhin wäre ein Highscore, welcher sich speichern und laden lässt, wünschenswert. In Bezug auf die Begriffe wäre es angebracht, dass die Daten in der XML-Datei verschlüsselt werden, um zu verhindern, dass diese Datei einfach angesehen werden kann und die Lösung somit zugänglich ist. Gestaltungselemente, wie zum Beispiel austauschbare Hintergrundbilder, Sound oder verschiedene Farbthemen wären eine kreative Erweiterung.

Interessant wäre auch, das Tablett nicht bzw. nicht nur durch den Stift zu bedienen, sondern mit Hilfe einer Kamera, welche am Tablett montiert wird und durch das Kippen des Tablett-PCs das Rad in Bewegung versetzt. Für die Buchstabeneingabe bzw. den Lösungsversuch, wäre es denkbar eine Sprachsteuerung zu integrieren, bei der die einzelnen Buchstaben bzw. die gesamte Lösung angesagt werden kann.

IV. Literaturverzeichnis

1 Bücher

- [L01] Jarrett, Rob; Su, Philip: Building Tablet PC Applications. Redmond, Washington: Microsoft Press, 2002 - 576 S. - ISBN 0-7356-1723-6
- [L02] Crooks II, Clayton E.: Developing Tablet PC Applications. Hingham, Massachusetts: Charles River Media, 2004 - 355 S. - ISBN 1-584450-252-5

2 Links

- [L03] Diverse: Google Groups BETA. URL <http://groups.google.com> [14.11.2005] - Aktualisierungsdatum: ständige Aktualisierung
- [L04] Corti, Sascha: Microsoft Windows XP für Tablet PC mit Hilfe der .NET Framework Pen-, Ink- und Reco-API Klassenbibliotheken programmieren URL <http://codezone.skypro.tv/245/html/frame.htm> [14.11.2005] - Aktualisierungsdatum: 01.09.2002 - mailto: saschac@microsoft.com
- [L05] Microsoft: Microsoft Tablet PC - Programming the Input Panel Using the PenInputPanel Class URL http://msdn.microsoft.com/library/default.asp?url=/library/en-us/tpcsdk10/lonestar/whitepapers/designguide/tbcon_usingthepeninputpanel.asp [14.11.2005] - Aktualisierungsdatum: 22.04.2004
- [L06] Hassler, Chris: Tablet Creations Tutorials URL <http://nicecreations.us/tutorials.htm> [14.11.2005]
- [L07] Microsoft: Microsoft Education Pack for Windows XP Tablet PC Edition. URL <http://www.microsoft.com/downloads/details.aspx?familyid=9D346916-B526-487E-919D-0BCE568DEF39&displaylang=en> [14.11.2005] - Aktualisierungsdatum: 24.07.2005
- [L08] Microsoft: Update for Microsoft Windows XP Tablet PC Edition Development Kit 1.7. URL <http://www.microsoft.com/downloads/details.aspx?FamilyID=69640b5c-0ee9-421e-8d5c-d40debee36c2> [14.11.2005] - Aktualisierungsdatum: 14.11.2005
- [L09] Microsoft: Tablet PC Developer. URL <http://www.devx.com/TabletPC/Door/20301> [14.11.2005] - Aktualisierungsdatum: ständige Aktualisierung
- [L10] Einstein Technologies: Tablet Enhancements for Outlook . URL <http://www.tabletoutlook.com/> [14.11.2005]
- [L11] Microsoft: Windows XP Downloads. URL <http://www.microsoft.com/windowsxp/tabletpc/downloads/default.asp> [14.11.2005]
- [L12] Arten von Tablet-PCs <http://www.tablet-cc.de/> [07.02.2006]
- [L13] PDF Annotator <http://www.ograh.com/en/pdfannotator>[03.02.2006]
- [L14] Webplanet Concept X7 <http://www.easy-use.de/index.html?lang=de&target=d160.html>[03.02.2006]
- [L13] xThink Calculator <http://www.easy-use.de/index.html?lang=de&target=d160.html>[03.02.2006]
- [L13] nTcMedic <http://www.easy-use.de/index.html?lang=de&target=d160.html>[03.02.2006]