Notizen und Übersichten zu

PotiPat Patiencen CS

Ulrich Müller

Inhaltsverzeichnis

PotiPatCS.DLL

Allgemeines

Karten

QCard32 bietet zwei vollständige Kartendecks mit jeweils 3 Jokers und zusätzlich 6 Kartenrückseiten. Die Karten sind durchnumeriert :

1 – 52	Deck 1
53 – 104	Deck 2
105 – 109	Kartenrückseiten
110 – 113	Joker

Enums

SuitCode

ColorCode

ValueCode

FaceCode

AreaCode

Struktur

CardPos

Exceptions

QCard32Eception

Klasse QCard32Pool

Pool von Spielkarten (zwei Pakete & jeweils 2 Joker) mit Zeichen- und Verwaltungsfunktionen dazu. Die Klasse kapselt die QCard32.DLL. Dort wird auch das Handle (this.Handle der Form auf der die Karten angezeigt werden sollen) für die ZeichenFläche zentral gespeichert. Dieser Parameter fehlt dann bei den entsprechenden Klassenfunktionen aus QCard32.

Konstruktor

Achtung : pro Programm ist nur eine Instanz von QCardPool zulässig (intern wird InitializeDeck verwendet).

QCardPool(IntPtr hWnd)

hWnd : Handle des Controls auf dem die Karten gezeichnet werden sollen (im einfachsten Fall die Hauptform der Anwendung).

Mit default-Werten für die Kartenrückseite (BackNr = 1)
und das Offset bei Leitern (Offset = 16)

QCardPool(IntPtr hWnd, int BackNr, int Offset)

hWnd : wie oben.

NrBack : Nummer der Kartenrückseite (1 – 6). Verwendet beim Zeichnen verdeckter Karten.

Offset : Abstand (in Pixeln, default 16) von senkrecht ausgelegten Karten (Leitern) zueinander.

Eigenschaften

int	**BackNr**
int	**CardWidth** (get)
int	**CardWidthS**
int	**CardHeight** (get)
int	**Offset**

Indexer

Karten des QCard32Pool in der Klasse PoolCard mit den Eigenschaften :

int	**X** : Aktuelle Kartenposition
int	**Y**
bool	**Blocked**
ColorCode	**Color** (get)
bool	**Disabled**
FaceCode	**Face**
SuitCode	**Suit** (get)
ValueCode	**Value** (get)
bool	**User1**
int	**User2**
int	**User3**
int	**User4**

Methoden

AbortDrag

Abbruch einer durch InitDrag gestarteten Drag-Operation. Freigeben der Maus. Karte bleibt auf Ursprungsposition.

Siehe auch : InitDrag, AbortDrag, DoDrag, BlockDrag, EndDrag, EndBlockDrag, ReturnDrag, ReturnBlockDrag

BlockDrag

Durchführen einer Drag-Operation mit mehreren aufeinanderliegenden Karten (Teil-Leitern), Karten werden ständig gezeichnet, Koordinaten werden upgedated.

DealCard

Zeichnen einer Karte and der angegebenen Position, Update der internen Buchhaltung.

DoDrag

Durchführen der Drag-Operation(einzelne Karte), die durch InitDrag gestartet wurde. Die Karte wird ständig gezeichnet, die Koordinaten werden upgedated.

DrawBack

Zeichnen einer Kartenrückseite (1- 6) : Kugeln, Blau, Rot, Berge, Karos, Klavier), keine Buchhaltung.

DrawCard

Zeichnen einer Karte an der angegebenen Position, keine interne Buchhaltung. Nutzung primär für Paint-Ereignis.

DrawSymbol

Zeichnen eines Platzhalters für eine Kartenposition (1 – 3 : X, Kreis, Grau), keine Buchhaltung.

Siehe auch : DrawSymbol, DrawCard, DrawBack, DealCard, RemoveCard

EndBlockDrag

Beenden einer BlockDrag-Operation, Rückgabe der Nr der "unterliegenden" Karte (if any, Blocked false, sonst 0). Zeichnen der karten, Buchhaltung. Freigabe Maus.

EndDrag

Beenden einer einfachen Drag-Operation, Rückgabe der Nr der "unterliegenden" Karte (if any, Blocked false, sonst 0). Zeichnen der Karte, Buchhaltung. Freigabe Maus.

InitDrag

Starten einer Drag-Funktion, Identifikation der durch MouseDown ausgewählten Karte durch deren Kartennummer (0 = keine Karte oder Karte Disabled). Die Maus wird "captured".

RemoveCard

Wiederherstellen des vollständigen Hintergrundes einer Karte, die mit DealCard gezeichnet wurde, keine Buchhaltung.

ReturnBlockDrag

Beenden einer BlockDrag-Operation, Verwendung bei Abbruch Drag, wenn die Spiellogik es verlangt. Die Karten werden an der angegebenen Position gezeichnet. Buchhaltung, Freigabe der Maus.

ReturnDrag

Beenden einer einfachen Drag-Operation, Verwendung bei Abbruch Drag, wenn die Spiellogik es verlangt, die Karte wird an der angegebenen Position gezeichnet. Buchhaltung. Freigabe Maus.

Karten-Klassen Card, PoolCard, BaseCard

Card : BaseCard Konstruktor

Card(CardNr)

Card(CardNr, Left, Top)

Left / Top : default –72 / - 97

Card Eigenschaften

AreaCode	**Area**
int	**CardNr** (get)
bool	**Blocked**
ColorCode	**Color** (get)
bool	**Disabled**
FaceCode	**Face**
int	**PicX**
int	**PicY**
int	**PosC**
int	**PosR**
SuitCode	**Suit** (get)
ValueCode	**Value** (get)
int	**X**
int	**Y**

PicX/PicY Koordinaten letzt gültige Ablageposition, X/Y : akt. Position.
PosC Spalte, PosR Eintragsnummer in der Spalte.

Klasse PoolCard : BaseCard

siehe Indexer von QCard32Pool

Klasse BaseCard

Basisklasse für PoolCard und Card

Struktur CardPos

Kartenposition (für Umlege-Operationen).

int	**CardNr**
AreaCode	**Area**
int	**PicX / PicY**
int	**PosC / PosR**
int	**X / Y**

Talon-Klasse KartenPaket

Konstruktor

Eigenschaften

int	**CardCount** (get)
int	**ID** (get)
int	**Left** (get)
Rectangle	**Range** (get)
int	**Top** (get)
Card	**TopMostCard** (get)

Methoden

AddCard

NewDeck

Paint

RemoveCard

Keller-Klasse LiFoStack

Konstruktor

Aufeinander / nach rechts geschuppt.

Eigenschaften

Left

Range

Top

TopMostCard

Methoden

AddCard

Clear

Paint

RemoveCard

Auslage Klasse AuslageV

Auslage der Patience

Konstruktor

AuslageV(QCard, NrCols, CardWidth, Left, Top, Height)

Kartenlage waagerecht. NrCols : Anzahl senkrechteLeitern, Leiterbreite CardWidth, Gesamthöhe Height, Koordinaten linke obere Ecke Left/Top.
MaxAnzahl Karten pro Leiter fix 40.

Eigenschaften

Left

Range

Top

Methoden

ActualCard

AddCard

AddRange

CardCount

Clear

IsDescendingChanging

IsEnoughSpace

IsUpDownSuit

Paint

RemoveCard

RemoveRange

TopMostCard

XToCol

Ablage-Klasse Ablage48

Ablagebereich der Patience, wahlweise 4 bzw. 8 Karten mit wählbarer Lücke nach der vierten Karte. Waagerecht.

Konstruktor

Ablage48(QCard, NrCols, CardWidth, Left, Top, NrEmpty)

Ablage48(QCard, NrCols, CardWidth, Left, Top)

Eigenschaften

CardCount

Left

Range

Top

Methoden

AddCard

Clear

ColToX

IsAscendingSuit

IsDescendingSuit

Paint

TopMostCard

XToCol

Patience-Klasse PotiPatience

Abstrakte Klasse als Basis zur Beschreibung der Spiellogik. Geplant : AusBau zu einer Klasse Spielfeld mit Elemente für die Kommunikation der Patiencekomponenten untereinander. Das wird bisher in der abgeleiteten Klasse getan.

Konstruktor

PotiPatience(Canvas)

Initialisieren von QCard32Pool. Fehlt BackNr, Hintergrundfarbe des Canvas.

Achtung muß nach Form.Show aufgerufen werden. ! bei Form_Load.

Eigenschaften

Back

Methoden

alle abstract

EndeZiehen

KarteAblegen

Refresh

SpielStarten

StartZiehen

WeiterZiehen

Spiel-Klasse TestPatience : PotiPatience

Spielfeld und Spiellogik. Spezifisch für die jeweilige Patience erstellt.

Konstruktor

TestPatience(Canvas)

Eigenschaften

Methoden

CardToAblage

EndeZiehen

FindDestPos

KarteAblegen

Refresh

SpielAuslegen

SpielStarten

StartZiehen

WeiterZiehen

Form Testrahmen

Form_Load

Mit new TestPatience

Form_Paint

Mit Refresh

Form_MouseDown / Move / Up

Mit StartZiehen / WeiterZiehen / EndeZiehen

Form_DoubleClick

Mit KarteAblegen

Patiencen

Grand Napoleon CS

Spielablauf

Auslegen und Ziel

Grand Napoleon wird mit zwei Spielen mit je 52 Karten gespielt, die alle offen in Reihen a 11 Karten ausgelegt werden. Ziel des Spiels ist es die ausgelegten Karten links - aufsteigend - auf vier Assen und rechts -absteigend - auf vier Königen abzulegen.

Nach dem Auslegen jeder Reihe wird geprüft, ob Karten abgelegt werden können. Asse, Könige oder auch Karten, die auf bereits abgelegte passen. Bei der ersten Reihe können alle Karten zur Ablage herangezogen werden, bei den weiteren nur die äußeren und ggf. die darunter liegenden.

Umlegen

Sind alle Karten ausgelegt, können die Karten der Auslage umgelegt werden. Es gilt : Anlegen an gleiche Suite (Kreuz an Kreuz ...) aufsteigend oder absteigend. In einer Spalte kann beliebig zwischen auf- und absteigend gewechselt werden.

Es können nur jeweils einzelne Karten umgelegt werden. Freiwerdende Spalten können mit einer beliebigen Karte belegt werden.

Passende Karten können wahlweise auch abgelegt werden.

Wiederauslegen

Wenn keine Spielzüge mehr möglich sind, ist ein Einsammeln der Karten - beginnend mit der Karte rechts unten der rechten Spalte bis hin zur Karte links oben der linken Spalte - und daran anschließend ein Wiederauslegen nach den Regeln von Auslegen möglich. Das Wiederauslegen ist zweimal möglich, dann muß das Spiel aufgegangen sein.

Bedienung

Spielstart und Auslegen

Ein Spiel kann über den Menüpunkt Spiel gestartet werden. Es kann zwischen zufälligem Kartenbild ("Neues Spiel"), wählbarem Kartenbild ("Spiel Auswählen", Wahl der Startnummer, "Nächstes Spiel" Startnummer des vorhergehenden Spiel+1) und - ab dem zweiten Spiel - Wiederholen des letzten Spiels ("Spiel Wiederholen") gewählt werden.

Das Spiel wird durch das Programm mit dem Auslegen der ersten Reihe gestartet. Es können jetzt Karten durch Ziehen mit der Maus (Drag and Drop) auf die Ablage gelegt werden.

Können keine Karten mehr abgelegt werden, so kann durch Klick auf den Button "Weiter" die nächste Reihe ausgelegt werden. Ab der zweiten Reihe sind nur noch die jeweils äußeren Karten "ziehbar".

Umlegen und Wiederauslegen

Nach dem Auslegen der letzten Karten verschwindet der Button "Weiter", der Button "Nächst Runde" wird sichtbar. Zwischen beiden Buttons wird links die Nummer der aktuellen Runde und rechts die Anzahl der in der Auslage verbliebenden Restkarten angezeigt.

Es kann jetzt mittels Drag and Drop in der Auslage umgelegt oder in die Ablage abgelegt werden. Wenn nichts mehr geht kann durch Klick auf den Button "Nächste Runde" die erste Reihe der nächsten Runde ausgelegt werden. Intern werden dazu alle Karten eingesammelt (s.o.) und von oben (d.h. die letzte eingesammelte Karte als erste) wieder ausgelegt. Der Button "Weiter" ist wieder sichtbar, der Button "Nächste Runde" nicht.

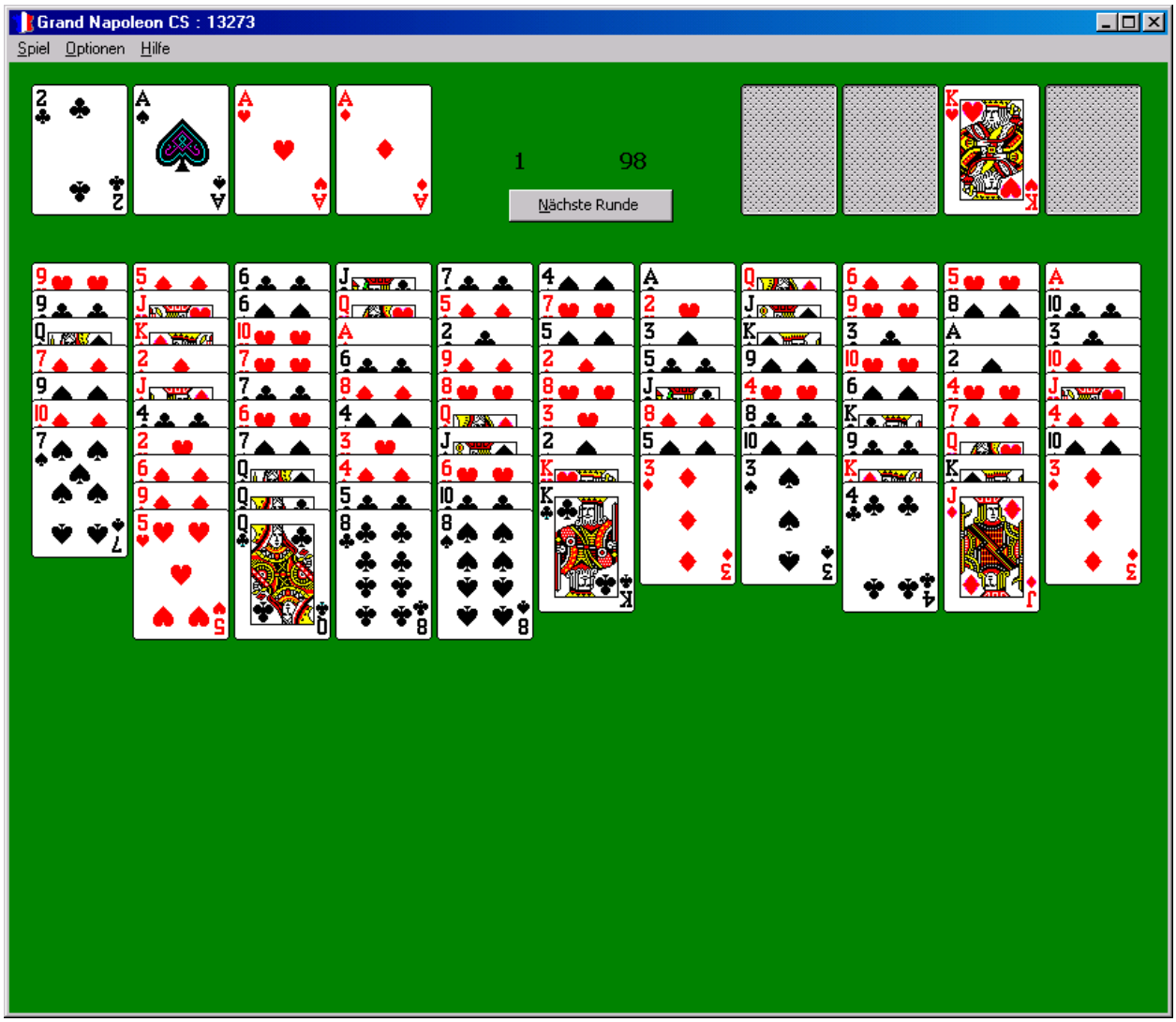
Nach dem Auslegen einer Reihe kann wie üblich abgelegt werden. Nach Klick auf "Weiter" wird die nächste Reihe ausgelegt. Nach Auslegen aller Karten wird "Weiter" unsichtbar und "Nächste Runde" erscheint wieder. Es kann wieder um- und abgelegt werden.

Nach zweimaligem vollständigem Wiederauslegen verschwindet der Button "Nächste Runde".

Spielende

Das Spielende ist erreicht, wenn alle Karten abgelegt sind (Anzeige Restkarten 0). Das Programm zeigt dann "Gewonnen" an.

Programm-Rahmen



frmMain_Load/Closed

Instanzieren mp = New MolPatience(Me.Handle)
Deserialize der persistenzen Daten (Left/Top, Anzahl Spiele/Gewonnen, Datum letzter Besuch)

Serialize der persistenten Daten

frmMain_Paint

If ArbeitsMode > 0 then gn.Refresh()

mnuNeu_Click, mnuNext, mnuWahl, mnuWieder

gn.SpielStarten(SpielNr)

ArbeitsMode = 1

frmMain_MouseDown

gn.StartZiehen(X, Y)

frmMain_MouseMove

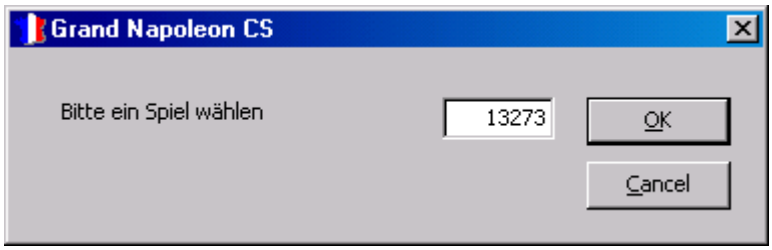
gn.WeiterZiehen(X, Y)

frmMain_MouseUp

gn.EndeZiehen

gn_GameWon

Anzeige



GrandNapPat / PotiPatience

Spiellogik. Siehe auch Moltke, kein BlockDrag, keine Spielhilfe a la DoppelClick.

Verwendet werden Ablage As Ablage48, Auslage As AuslageV und Talon As KartenPaket, kein Keller. Der Talon ist unsichtbar (verschoben über linkoben hinaus).