株式会社ミガロ.

システム事業部 プロジェクト推進室

[Delphi/400] Windowsタブレット用に カスタムソフトウェア キーボードを実装

●はじめに ●ソフトウェアキーボードについて ●TTouchKeyboard コンポーネントでの実装 ●カスタムソフトウェアキーボードでの実装 ●まとめ



^{昭歴} 1972 年 3 月 20 日生まれ 1994 年 大阪電気通信大学工学部卒業 2001 年 4 月 株式会社ミガロ. 入社 2001 年 4 月 システム事業部配属

現在の仕事内容 主に Delphi/400 を使用したシス テムの受託開発を担当しており、要 件確認から納品・フォローに至るま で、システム開発全般に携わってい る。また、Delphi/400の導入支援 やセミナーの講師も行っている。

1.はじめに

ここ数年、業務用としてタブレットを 導入、または導入を検討する企業が増え ている。そのタブレットを導入する際に 重要な検討項目となるのが、OSの選択 である。

Delphi/400 で は iOS や Android の アプリケーションも開発できるが、PC と同様に Windows を搭載したタブレッ トを選択する企業も少なくない。それは、 これまで購入してきた Windows ソフト ウェアや、現在使用中の業務システムを タブレットでもそのまま使いたいという 理由からである。

Windows タブレット用に新たにアプ リケーションを開発する場合、その方法 自体は、PC 用のアプリケーション開発 と同じである。ただし Windows タブ レットのソフトウェアキーボードは、 iOS や Android のそれと比べると操作 性に難があり、不便と感じられることが ある。

そこで本稿では、ユーザーにとって操

作性のよい Windows タブレット用ソフ トウェアキーボードの実装方法を紹介す る。

2.ソフトウェア キーボードについて

2-1. ソフトウェアキーボードとは 本題に入る前に、ソフトウェアキー ボードについて少し説明する。ソフト ウェアキーボードとは、物理的なキー ボード機器を使用せずに画面上でキー ボードを表示し、マウスクリックや画面 タッチによって文字や数値等を入力でき るソフトウェアである。

2-2. Windows 標準のソフトウェアキー ボード

ここでは、Windows 8 搭載のタブレットに装備されている標準ソフトウェアキーボードについて基本的な動作を確認する。

ソフトウェアキーボードを表示する には、タスクバーに表示されているキー ボードイメージのアイコンをタッチす る。【図 1】

先ほど、Windows タブレットの標準 ソフトウェアキーボードは、「操作性に 難があり、不便と感じられることがある」 と記したが、具体的には次のような点で ある。

 アプリケーションの入力項目にフォー カスが移っても、ソフトウェアキー ボードは自動的に表示されず、タスク バーのキーボードアイコンをタッチし て表示させなければならない。【図2】
 ソフトウェアキーボードを開くと、ア プリケーションの画面に被ってしま い入力項目が隠れてしまう場合があ る。【図3】

タブレット用のアプリケーションを 開発する際、上記が問題となることが多 い。そこで、それを改善するため標準の ソフトウェアキーボードは使用せずに、 アプリケーションの中でソフトウェア キーボード機能を実装する方法を紹介す

図1														
q	2 W	₃ e	4 r	5 t	6 y	⁷ u	⁸ i	9 O	° p	<	∎ × ⊠			
	a s	d	f	g	h	j	k	1			Enter		-	
↑ &123	Z 3 Ctrl	×	с あ	V	b	n	m		° <	>	ここをタッ	チする	<u></u>	
											•		^ ⊺∎	
図2						_								
	タムソフトウ	Sample ェアキーボー		×										
	項目1		1 CHax											
		I		100										
	項目2													
	項目 2													
	項目2 項目3 ここを: + 二	「 」 タッチして:	/フトウェ]	アキーボ										
	項目2 項目3 ここを: 表示さ	タッチして) させなけれ	ノフトウェ ばならなし	アキーボ										
	項目 2 項目 3 ここを 表示さ	タッチして) させなけれ	ノフトウェ ばならなし	アキーボ ,1		Ĩ								
	項目 2 項目 3 ここを 表示さ	タッチして) ジゼなけれ	ノフトウェ ばならない	アキーボ ハ		1								
	項目 2 項目 3 ここを: 表示さ	 タッチして: せなけれ	ノフトウェン	アキーボ , 1		ī								
⊠3	項目 2 項目 3 ここを: 表示さ	タッチして) マッチして) せなけれ	ノフトウェン	アキーボ ,)										
⊠3	項目 2 項目 3 ここを: 表示さ	」 タッチしてと :せなけれ	ノフトウェン	アキーボ ハ										
⊠3	項目 2 項目 3 ここを: 表示さ	 タッチして) させなけれ	ノフトウェン	アキーボ ,)		1) J	Sar フトウェア	nple キーボート	- ロ *を実装	×			
⊠3	項目 2 項目 3 ここを: 表示さ	タッチして!	ノフトウェご	アキーボ ハ	F&	1) フスタムソ ^ー 項目 1	Sar フトウェア	nple キーボート	- ロ **を実装	×			
⊠3	項目 2 項目 3 ここを: 表示さ	タッチして) せなけれ	ノフトウェ ばならなし	<i>P</i> + − <i>π</i> ,	с – К &	T U U) リスタムソ [・] 項目1 8 i	Sar フトウェア	nple ≠−ボート	- ロ でを実装	× ,			
⊠3	項目 2 項目 3 ここを: 表示さ	タッチして) させな(けれ) ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	ノフトウェご ばならなし ⁴ r f	^r + − ボ ,, ⁵ t	j y h	1 1 7 4 7 7	コスタムソ ^ー 項目1 ⁸ i k	Sar フトウェア ⁹ O I	nple = - ボート p –	- □ でを実装	× Therefore			
⊠3	項目 2 項目 3 ここを: 表示さ a 、 、 、 、 、 、 、 、 、 、 、 、 、	タッチして: せなけれ っ っ っ っ っ っ っ っ っ っ っ っ っ っ っ っ い ろ	ノフトウェ ばならなし イ r f	ד + – ד, יי ז ל ע	E−Fを ■ • •	T T U T N	コスタムソ ^ー 項目1 ⁸ i k m	Sar フトウェア ⁹ の し	nple ≠ — ボ — ト P 	- □ *を実装 ?	× The second se			
⊠3 ¹ q ¹ R ¹ 2	項目 2 項目 3 「ここを: 表示さ 「 ま示さ 」 、 、 、 、 、 、 、 、 、 、 、 、 、	x 2 2 2 2 2 2 2 2 2 2 2 2 2	ノフトウェご ばならなし A r f c あ	<i>p</i> + − <i>i i g v</i>	- Ка т у h b	T T U J N	コスタムソ ^ー 項目1 ⁸ i k m	Sar フトウェア ⁹ の し 、 次頁	nple = - ボ - ト p - -	- □ でを実装 ? 、 、	× Enter T		ŤD	

る。

2-3. カスタムソフトウェアキーボード を独自実装

タブレットでは、ユーザーインター フェースとしてコンボボックスやボタン 類を多用し、キーボード入力を極力少な くするのがセオリーである。しかし、数 量や金額などの数値に関しては、リスト から選択させるわけにもいかず、キー ボードからの入力が必要になることが多 い。そこで、これから紹介するソフトウェ アキーボードは、タブレットで使用率が 高い、テンキーのキーボードを題材とし て説明していく。

3.TTouchKeyboard コンポーネントでの 実装

3-1. TTouchKeyboard コンポーネント の実装手順

Delphi/400 で は、Ver.2010 か ら TTouchKeyboard コンポーネントが実 装されている。このコンポーネントを使 用することで、アプリケーション中にソ フトウェアキーボード機能を簡単に実装 できる。

TTouchKeyboard コンポーネントを フォームに配置すると、ソフトウェア キーボードがフォーム上に表示される。 またプロパティの設定だけで、通常の文 字キーボードとテンキーのイメージを切 り替えることができる。【図 4】

ここから、TTouchKeyboard を使用 したソフトウェアキーボードの実装手順 について説明する。

①コンポーネント配置

新規フォーム上に TEdit コンポーネ ントを2個と、TTouchKeyboard コン ポーネントを配置する。【図5】

 TTouchKeyboard のプロパティ設定 TTouchKeyboard コンポーネントの Layout プロパティを「NumPad」に設 定し、テンキーイメージにする。【図 6】
 ③実装完了

TTouchKeyboard コンポーネントと フォームのサイズを整えて実装は完了と なる。【図 7】

TTouchKeyboard コンポーネントの

Layout プロパティを変更した以外、特 にソースを記述する必要はないので、非 常に簡易に実装できる。

3-2. TTouchKeyboard コンポーネント の機能

この節では、実際の動作について確認 する。

アプリケーションを実行すると、 フォーカスが Edit1 にある状態で画面が 表示される。ソフトウェアキーボードの 「1」「2」をタッチすると、Edit1 に「1」 「2」がセットされる。【図 8】

次に、Edit2にフォーカスを移してソ フトウェアキーボードの「4」「5」をタッ チすると、Edit2に「4」「5」がセット される。【図 9】

このようにTTouchKeyboard コン ポーネントは、同一フォーム上のアク ティブなコンポーネントに対して値を セットするため、入力項目があるフォー ム上に配置する必要はあるものの、ソフ トウェアキーボードを簡単に実装したい 場合に便利なコンポーネントである。

ただし、アプリケーションによって は、ソフトウェアキーボードを表示する 領域が画面デザイン上、難しい場合もあ る。そうした場合には、アプリケーショ ンとは別のウインドウにソフトウェア キーボードを分離させる必要がある。そ こで、別のウインドウとしてソフトウェ アキーボードを作成し、アプリケーショ ンに実装する方法を次章で紹介してい く。

4.カスタムソフトウェア キーボードを実装

4-1. ソフトウェアキーボードの開発手順 この章では、【図 10】のように入力画 面とソフトウェアキーボードを別フォー ムとして作成し、ソフトウェアキーボー ドが入力画面に対してキーボードとして 動作するように実装する方法を説明す る。この実装方法では、簡易な TTouchKeyboard コンポーネントを使 わずに独自にソフトウェアキーボードを 作成するため、細かい制御を自由に実装 できる。

まず、ソフトウェアキーボードを新し いフォームとして作成する。今回作成す る画面イメージは【図 11】の通りである。 機能としては、数値入力、マイナス入 力、小数点入力、BackSpace、フォー カスの移動(次項目、前項目)、入力項 目値の全選択機能を実装する。では、作 成手順を順番に説明する。

①コンポーネント配置

【図 11】に従って、新規フォーム上に TBitBtn コンポーネントを 16 個配置し、 Caption プロパティを設定する。続いて 【表 1】に従って、各 TBitBtn コンポー ネントの Name プロパティを設定する。 ②フォームのプロパティ設定

フォームの BorderStyle プロパティ を「bsSingle」に設定して、ソフトウェ アキーボードの画面サイズを変更不可に する。次に FormStyle プロパティを 「fsStayOnTop」に設定し、ソフトウェ アキーボードが常に手前で表示されるよ うにする。【図 12】

③グローバル変数の宣言

グローバル変数として DeActivate Form を記述しておく【ソース1】。こ れは DeActivateForm 変数で受け取っ たフォームに対して、別ウインドウのソ フトウェアキーボードでボタンタッチさ れた結果を反映するためである。

④フォームの onCreate イベントハンド ラの実装

フォームの onCreate イベントハンド ラを次のように実装する。【ソース 2】

ここでは、btnKey0~btnKey9、 btnKeyDot、btnKeyMinusのTagプ ロパティに、ボタンのCaptionに該当 する文字コードを設定しておく。今回は 数値と記号のみだが、文字キーボードを 実装する場合には「Ord('A')」といっ た形で、引数にアルファベットを指定す ることもできる。

次に、btnKeyBSのTagプロパティ にはBackSpaceキーに該当する制御 コードを設定し、btnNextとbtnPrior のTagプロパティには、前項目への移 動なのか、次項目への移動なのかを判断 するための区分を設定しておく。ここで 設定した内容は、後で説明するTBitBtn コンポーネントのonClickイベントハ ンドラで使用することになる。そして、 ソフトウェアキーボードのTopとLeft を指定し、初期表示位置を画面右下にな るよう設定しておく。

図4	
図4 通常のキーボード Esc 漢字 1 2 3 4 5 6 7 8 9 0 - ^ ¥ f Esc 漢字 1 2 3 4 5 6 7 8 9 0 - ^ ¥ f Q W e r t y u i o P @ [] Del Caps a s d f g h j k l ; : f Caps a s d f g h j k l ; : f Ctrl Alt 無変換 D J Alt Ctrl f l f T Alt 無変換 D J Alt Ctrl f l f T A S 9 + f C T A S 9 + f C T A S 5 5 7 8 9 + f C T A S 5 7 8 9 + f C T A S 5 7 8 9 + f C T A S 5 7 8 9 + f C T A S 5 7 8 9 + f C T A S 5 7 8 9 + f C T A S 5 7 8 9 + f C T A S 5 7 8 9 + f C T A S 5 7 8 9 + f C T A S 5 7 8 9 + f C T A S 5 7 8 9 + f C T A S 5 7 8 9 + f C T A S 5 7 8 9 7 8 7 8 7 8 7 8 7 8 7 8 7 8 7 8 7	
Y = Y = Y = Y = Y = Y = Y = Y = Y =	
Caps a s d f g h j k l ; :	
▲ Anchors [akLeft,akTop] Color □ dBtnFace Cursor crDefault CustomHint DrawingStyle DrawingStyle dsNormal Enabled ☑ True GradientEnd clSilver GradientStart clGray Height 180 HelpContext 0 HelpKeyword htContext Hint KeyCaptione Layout NumPad Left 8 LiveBinding デザイナ LiveBinding デザイナ	

⑤ TBitBtn コンポーネントの onClick イベントハンドラ

各 TBitBtn コンポーネントの onClick イベントハンドラを実装する。イベント ハンドラはボタンごとに用意するのでは なく、機能ごとに共通化して3つのイベ ントハンドラを用意する。【ソース3】

各イベントハンドラのソースは、 【ソース4】のようになる。そして、各 TBitBtn コンポーネントの onClick イ ベントハンドラを次のように設定する。 【表2】

btnKeyBoardClick では、DeActivate Form 変数で受け取ったフォームを BringToFront を使用してアクティブに し、フォームのアクティブコンポーネン トに対して TBitBtn の Tag プロパティ に設定されている文字コードや制御コー ドを、PostMessage 関数を使用してメッ セージ送信している。この実装で、ソフ トウェアキーボードの数値、記号、「BS」 を押した結果が入力項目に反映されるよ うになる。

btnSelectALL で は、btnKeyBoard Click 同様にフォームをアクティブに し、フォームのアクティブコンポーネン トに対して値の全選択命令をメッセージ 送信している。

btnFocusControl では、btnKeyBoard Click 同様にフォームをアクティブに し、TBitBtn の Tag プロパティに設定 されている区分によってフォームのアク ティブコンポーネントの次項目または前 項目にフォーカスを移動させる命令を メッセージ送信している。

以上でソフトウェアキーボードの作 成は完了である。

4-2. 入力画面への実装手順

続いて、入力画面へソフトウェアキー ボードを実装する方法について説明す る。実装の手順は、次の通りである。

①入力画面の作成

前節のソフトウェアキーボードと同 じプロジェクト内に、新規フォームを追 加して TEdit コンポーネントを3個配 置する【図13】。そして、ソフトウェア キーボードのユニットを参照しておく。 また、プロジェクトオプションのフォー ムの設定では、ソフトウェアキーボード も自動生成の対象としておく。【図 14】 ②フォームの onShow イベントハンド ラの実装

ソフトウェアキーボードは【図 14】 で自動生成されているため、フォームの onShow イベントでは、ソフトウェア キーボードを Show することで表示さ せることができる。【ソース 5】

③フォームの onDeactivate イベントハ ンドラの実装

ソフトウェアキーボードのボタンを押 した際、入力中のフォームからソフト ウェアキーボードへ制御が移る。この時 にソフトウェアキーボードの Deactivate Form 変数に入力中のフォームをセット しておくため、フォームの on Deactivate イベントハンドラを次のように実装す る。【ソース 6】

以上で入力画面への実装は完了であ る。

4-3. カスタムソフトウェアキーボード の機能

実行して動作を確認する。アプリケー ションを実行すると入力画面とソフト ウェアキーボードが表示され、入力画面 のEdit1にフォーカスが設定される。【図 15】

入力画面とソフトウェアキーボード は別々のフォームになっているので、ソ フトウェアキーボードは画面上の好きな 位置に移動できる。

実際にソフトウェアキーボードを使 用して入力してみよう。ソフトウェア キーボードの「1」「2」をタッチすると Edit1 に「1」「2」がセットされ、「全選択」 をタッチすると、入力内容が全選択され た状態となる。【図 16】

この状態で Editl をタッチ長押しで ポップアップメニューを開き、コピーを 選択すると、入力中の値をコピーするこ ともできる。

次に、「BS」をタッチすると、Edit1 に入力されていた「12」がクリアされる。 さらに「次項目」を押せば Edit2 へ移り、 「前項目」を押せば Edit1 へフォーカス を戻すことができる。

このようにソフトウェアキーボード を独自に作成する場合は、TTouchKey board コンポーネントよりも実装に手間 がかかるが、別ウインドウとして制御し たり、独自の機能を実装できる利点があ る。

ここで、Windows タブレットのソフ トウェアキーボードが、どのように改善 されたのかをまとめておく。

1つ目の「アプリケーションの入力項 目にフォーカスが移っても、ソフトウェ アキーボードは自動的に表示されず、タ スクバーのキーボードアイコンをタッチ しなければならない」に関しては、ソフ トウェアキーボードを常に表示させてお くことで、入力項目にフォーカスが移っ ても、すぐに入力できるようになり解決 できる。

2つ目の「ソフトウェアキーボードを 開くと、アプリケーションの画面に被っ てしまい入力項目が隠れてしまう場合が ある」に関しては、画面起動時にソフト ウェアキーボードを入力項目に被らない 位置に表示させることで解決できる。た だし、今回の説明で使用した入力画面は 項目も少なく画面サイズも小さいため、 入力項目に重ならない位置に表示でき た。しかし、入力項目が多い画面では、 ソフトウェアキーボードがどうしても入 力項目に被ってしまうケースが出てく る。

そこで、次節では入力項目にソフト ウェアキーボードが重なってしまった場 合に、ソフトウェアキーボードの位置を 変える拡張方法について紹介する。

4-4. カスタムソフトウェアキーボード の移動

ここでは、【図 17】のように Edit2 に ソフトウェアキーボードが被っている状 態で、Edit2 にフォーカスを移した際に ソフトウェアキーボードを下方向にずら す調整方法を説明する。

この制御を実装するには、入力項目で ある各TEditコンポーネントの onEnterイベントハンドラを利用して実 装する。イベントハンドラは【ソース7】 のように共通化した1つのイベントハン ドラとして用意する。

イベントハンドラのソースを【ソース 8】に示す。

作成した共通イベントハンドラは、各 TEdit コンポーネントの onEnter イベ ントに設定する。

【ソース 8】について、ソースコード のポイントを説明する。

図7				
	🔇 Sample			
	Edit1		Edit2	
		/	*	-
	7	8	9	+
	4	5	6	
	1	2	3	
	0			

図8

G Sample	-	
Edit1	Edit2	
· /	*	-
7 8	9	+
4 5	6	
1 2	3	
0		

ĺ	🙆 Sampl	e			
	1		Edit2		
		/	*	-	
	7	8	9	+	
	4	5	6		Ľ
	1	2	3		
	0				

G Sampl	e		
12		Edit2	
	/	*	-
7	8	9	+
4	5	6	
1	2	3	\leftarrow
0			

図9







90

位置の調整については、最初に TEdit コンポーネントの位置(Top、Left)と ソフトウェアキーボードの位置(Top、 Left)を、スクリーンを基準にして算出 する。

次に、重なり合う部分を計算する IntersectRect 関数を使用して、TEdit コンポーネントとソフトウェアキーボー ドの重なりを求める。今回は上下方向の 重なりを判断している(if rRes.Height > 0 then)。横方向の重なりを判断した い場合は、rRes.Widthを使用すること で判断できる。

そして、重なりがあった場合にソフト ウェアキーボードを、TEditの底辺が Top 位置になるよう下方向に移動させ るのだが、その前に下方向に移動した場 合のソフトウェアキーボードの底辺の位 置を計算し (iBot := rCon.Top + rCon. Height + frmKeyBoard.Height + 5)、 スクリーン下にはみ出ないかを判断する (if iBot > Screen.Height then)。

スクリーン下にはみ出る場合は、 TEdit の Top 位置がソフトウェアキー ボードの底辺になるよう上方向に移動し (frmKeyBoard.Top := rCon.Top frmKeyBoard.Height - 5)、はみ出なけ れば下方向に移動させる (frmKey Board.Top := rCon.Top + rCon.Height + 5)。

実装内容は以上である。

次に、実際にアプリケーションを実行 して動作を確認する。Edit2 と Edit3 に ソフトウェアキーボードが重なっている 状態で、フォーカスを Edit1 → Edit2 → Edit3 と移動させると、ソフトウェア キーボードがフォーカスの移動に合わせ て下方向に移動していくことが確認でき る。【図 18】

また、Edit3 にフォーカスが移動した 際、ソフトウェアキーボードがスクリー ン下にはみ出る場合は、【図 19】のよう に上方向に移動する。

こうした実装を行うことで、前節で述 べていた入力項目が多い画面であって も、フォーカスの移動に合わせてソフト ウェアキーボードの位置を調整し、入力 項目を見えるようにできるので、ユー ザーが使用する際に非常に便利である。 また、この方法を用いれば、離れた位置 にあるソフトウェアキーボードを入力項 目の近くに移動させたり、入力項目から フォーカスが抜けた時に画面外にソフト ウェアキーボードを移動させる、といっ た制御も可能である。

5.まとめ

本稿では、Windows タブレットの標 準ソフトウェアキーボードが持つ不便さ の解消を目的に、アプリケーションによ る実装方法を紹介した。またデスクトッ プPC やノート PC で使用するアプリ ケーションの場合でも、今回のソフト ウェアキーボードを実装することで、 キーボードを使用することなくマウスの みで入力操作が可能な画面設計を行え る。

実装方法については、TTouchKey board コンポーネントを使った簡単な方 法と、独自にソフトウェアキーボードを 作成する方法を取り上げたが、どちらも 有効な方法なので、アプリケーションの 画面設計や用途によって使い分けるとよ い。

また今回は、単純なテンキーのキー ボードを題材にしたが、同様の方法で文 字キーボードの実装や、新しい機能の追 加も可能である。

Windows アプリケーションについて も、タブレットなどのスマートデバイス 端末での使用が増えてきている。そうし た開発を行う中で、デスクトップ PC や ノート PC にはなかった新しい課題に取 り組むことも多い。

開発方法はこれまでと同様であって も、使用するデバイスが変わると、使い 勝手や求められるユーザーインター フェースも違ってくる。こうしたことを 常に頭において設計を工夫することが、 これからのアプリケーション開発では重 要である。

Μ

図10)							
	Sample			×				
				ſ	ジンフトウェアキーボード			
	Edit2				7 8	9 BS		
	Edit3							
					4 5	6 全選択		
	_			_	1 2	3 次項目		
					0.	- 前項目		
_								
図11	1							
	() ソフトウェアキー	ボード		<u>x</u>				
	7 8	9	BS					
	4 5	i 6	全選扨	र				
	1 2	3	次項E	3				
	0.	-	前項目	3				
表1								
	TBitBtn⊐ì	ノボーネント	のNar	meブロバテ	ィの設定			
	Caption	Name		Caption	Name			
	0	btnKeyO		· ·	btnKeyDot			
	1	btnKey1		-	btnKeyMinus			
	2	btnKey2		BS	btnKeyBS			
	3	btnKey3		▲	btnSelAll			
	4 5	ptnKey4		<u> 水坝日</u> 前項日	btnNext			
	р В	htnKevő		門項日	וטנחרדוטר]		
	7	btnKev7						
	8	btnKey8						
	9	btnKey9						

Y-Z1 Y=Z1 y= Y=Z2 procedure IfrakeyBoard.FormCreate(Sender: TObject); b== Utation 0 Utation Utation Utation U Utation U U U U U U U U U U U U U U U U U U <th><pre>Proceedure IfrankeyBoard.FormCreate(Sander: TObject); bond:SU-Tag :: Ord('1'); bond:SU-Tag :: Ord('1'); bond:SU-Tag</pre></th> <th>X</th> <th>12</th> <th></th>	<pre>Proceedure IfrankeyBoard.FormCreate(Sander: TObject); bond:SU-Tag :: Ord('1'); bond:SU-Tag :: Ord('1'); bond:SU-Tag</pre>	X	12	
Image: State of the second state second state of the second state state of	<pre>fmtkcyBoard TimicyGoard fmtkcyBoard fmtkcyBoard TimicyGoard fmtkcyBoard TimicyGoard fmtkcyBoard fmtkcyBoard TimicyGoard fmtkcyBoard fmtkcyBoard TimicyGoard fmtkcyBoard f</pre>		🧏 オブジェクト インスペクタ 🛛 🖳	
プロパデュイベント! BDDPAdde bd_eftroSight BorderSive bsSingle Capton ソフトクスアキーボード Captonic Constraints Carsor cDeftadt Carsor cDeftadt Carsor cDeftadt Carsor cDeftadt Carsor cDeftadt DoubleSuffered False DragNode Chinkery Implementation V=X2 procedure If rakeyBoard.FormCreate (Sender : TObject); bergin Cond Differey? Tag Cond (12); biffkey? Tag Cond (12); biffkey? Tag Cond (12); biffkey? Tag Cond (12);	<pre>JUST 1 40000de bd.eftToRupti BondeStyle bd.eftToRupti BondeStyle bd.eftToRupti BondeStyle bd.eftToRupti BondeStyle bd.eftToRupti BondeStyle bd.eftToRupti Contonient BondeStyle bd.eftToRupti BondeStyle bd.eftToRupti BondeStyle bd.eftToRupti BondeStyle bd.eft BondeStyle bd.eft B</pre>		frmKeyBoard TfrmKeyBoard	
Protestyte bstandtrögigt Caption >>> Caption >>> Caption >>> Caption >>> Caption >>> Constraints Caption DodeBuffered False DodeStar Francespoard-TiresUpsoard-TiresUpsoard-FormOreate(Sender: Tobject); braint Proceedure IfrankeyBoard-FormOreate(Sender: Tobject); braint Francespoard-FormOreate(Sender: Tobject); breacedure IfrankeyBoards: Ord (??); <td><pre>blowded but reference borderStyle bsSingle DorderStyle bsSingle Derived bsSingle Dodde before Dodde free Dodde free Dodde</pre></td> <td></td> <td>プロパティ イベント</td> <td></td>	<pre>blowded but reference borderStyle bsSingle DorderStyle bsSingle Derived bsSingle Dodde before Dodde free Dodde free Dodde</pre>		プロパティ イベント	
V-Z2 V-Z2 V-Z2 V-Z2 brockwide var 1 aptematic 0 constants	<pre>V-A1 var statework is issued and the second an</pre>		BiDiMode bdLeftToRight	
V-Z2 V-Z1 V=Z1 V=Z1 V=Z1 V=Z1 V=Z1 V=Z1 V=Z1 var instanta instanta instanta instanta var instanta instanta instanta var instanta instanta instanta instanta var instanta intsystem intsystem intsystem intsystem intsystem intsystem intsystem intsys	<pre>variable variable variabl</pre>		BorderStyle bsSingle	
Capton y2Pyzy=-x-k-F Centroidh 320 Castoriants (TsizeConstraints) Cd30 True Custorinitit dActiveForm DodeBuffred False DoubleWinter dActiveForm DodeStite False DoubleWinter dActiveForm DodeStite False DoubleWinter dActiveForm DodeStite False DoubleWinter dActiveForm DodeStite FormStyle isstayOnTop Warring false DoubleWinter 0 ''True (Tsuescrame) ''Tota' (Tsuescrame) ''Tota' (Tsuescrame) <td>Capton >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>></td> <td></td> <td></td> <td></td>	Capton >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>			
UentHeight 320 Color □ dBtnFace B Constraints (TiseConstraints) Ct30 I'True CustorHint (TiseConstraints) CustorHint (TiseConstraints) CustorHint (TiseConstraints) DoubleBuffered False DoubleBuffered False DoubleBuffered False DoubleBuffered (TiseConstraints) If promotive fister	Understrugget 320 Corr □ disturace Corr □ disturace Corr □ obefault Customsint □ faile Doublebuffered □ formStyle Status (task) V-Z1 ************************************		Caption ソフトウェアキーボード	
John Street John Face Bib Constraints (TSReConstraints) Cutsor or Default Cutsor or Default DefaultMonitor dmActiveForm Doublebuffred False Doublebuffred John Street Proposition John Street Proposition John Street DefaultStreet John Street Proposition John Street Pr	Understands Image: Constraints Creater introl Cursor Creater introl Creater introl Cursor Creater introl Creater introl Defaultionion Creater introl Creater introl V=721 Var Creater introl Creater introl V=722 procedure If refault action Creater introl Creater introl V=720 procedure If refault action Creater introl Creater introl V=721 var fracter introl Creater introl Creater introl V=722 procedure If refault action Crea		ClientWidth 320	
Image: Constraints Constraints Cursor Orbefault Cursor Orbefault Cursor Orbefault Cursor Orbefault DediaWenter Galae DoubleWenter Galae DoubleWenter Galae Dragdod ddCrag DeActivateForm: Torne: Implementation '/	<pre> Constraints (TsueConstraints) Curso (Toru Curso (Toru</pre>		Color ClBtnFace	
CLSD WThe Customint matchveForm DoubleWorter matchveForm DoubleWorter False DragNode matchveForm DoubleWorter matchveForm DoubleWorter matchveForm DoubleWorter matchveForm DoubleWorter matchveForm DoubleWorter matchveForm War matchveForm Sasar raine reset V-721 Var Var reset DeActivateForm: TFrom: 348 HebContext 0 V-722 procedure TfrakeyBoard.FormOreate(Sender: TObject); berin if raise: ord('7'); btnkey7.lag i: ord('7'); btnkey8.lag i: ord('7');	Classor Classor Custorwint Custorwint DefaultWorter dnActveForm DoddBuffred Grag DragMode dnManual Bable True False Fraise DragMode dnManual Bable True False Fraise Prostote fstavonTop Height 348 V-Z1 ************************************		Constraints (TSizeConstraints)	
Usedom*init Userdaut. DefaultMontor dnActiveForm DefaultMontor dnActiveForm DedieBuffred False Doublebuffred false Promotyle fisstayOnTop Var fisstayOnTop Wars fisstayOnTop DeActivateForm: TForm; DeActivateForm: TForm; implementation 348 V-Z2 procedure IfrmKeyBoard-FormCreate(Sender: TObject); brinkey1.ag := Ord('1'); btnKey1.ag := Ord('1'); btnKey1.ag := Ord('1'); btnKey0.tag := Ord('2'); btnKey0.tag := Ord('2'); <td><pre> Default DefaultWonter DefaultWo</pre></td> <td></td> <td>Cti3D V True</td> <td></td>	<pre> Default DefaultWonter DefaultWo</pre>		Cti3D V True	
Defaultionitor DodSite DodBufferd DragGnd Brabled Fable DragGnd Brabled FormStyle FormStyle FormStyle FormStyle Brabled Understand HepContext 0	DefaultMonitor Default DodSite Frade DragKind dkorag DragKind dkorag DragKind dkorag PromStyle isStayOnTop Indextrante (VideorTeam) PormStyle isStayOnTop Indextrante (VideorTeam) PormStyle isStayOnTop Indextrante (VideorTeam) Indextrante (VideorTeam) Indextrante (VideorTeam) Implementation 0 V-Z2 procedure IffrmKeyBoard.FormCreate(Sender: TObject); btnkey7.trag := Ord('1'); btnkey7.trag := Ord('1'); btnkey7.trag := Ord('1'); btnkey7.trag := Ord('2'); btnkey7.trag := Ord('2'); btnkey7.trag := Ord('2'); btnkey7.trag := Ord('2'); btnkey8.trag := Ord('2'); btnkey8.trag := Ord('2'); btnkey8.trag :: Ord('2'); btnkey8.trag :: Ord('2'); btnkey8.trag :: Ord('2'); btnkey8.trag		CustomHint	
Doddite □Fabe DragNod ddrag DragNod ddrag <t< td=""><td><pre>DodSite</pre></td><td></td><td>DefaultMonitor dmActiveForm</td><td></td></t<>	<pre>DodSite</pre>		DefaultMonitor dmActiveForm	
Doublebufferd □ rate DragNode dwhanual DragNode dwhanual DragNode dwhanual DragNode dwhanual DragNode dwhanual DragNode dwhanual Pressonance dwasname Pr	Doubleturfered False DragWoode dm/Manual Enabled (Trees) PormStyle isStayOnTop Height 348 Height 348 Height 348 DeActivateForm: Thermit DeActivateForm: Torm; implementation Torm; y-Z2 procedure IfrmKeyBoard.FormCreate(Sender: TObject); bein // TBitBtnOTag /: Ord ('7'); btnKey7.1ag :: Ord ('1'); btnKey8.1ag :: Ord ('2'); btnKey0.1ag :: Ord ('2');		DockSite False	
Unsplice Immanual Enabled Winter Immanual Enabled PormStyle IsStayOnTop Wease name (Vease name) Height 340 Height 340 Height 340 DeActivateForm: Tform(y) Immanual Intervention 1 Y-Z1 ************************************	Draghode driving Bradbed If True Bradbed If True FormStyle fsStayOnTop Image: StayOnTop image: StayOnTop Imag		DoubleButtered False	
Enabled 図 True FormStyle isStayOnTop Yeas (Ydeast Tamle) HelpContext 0 V-Z1 Yar Yar frmKayBoard- IfrmKayBoard; DeActivateForm: TForm; implementation implementation // Y-Z2 procedure IfrmKeyBoard-FormCreate(Sender: TObject); // TBitBtnのTagブロパティを設定 btnKey7.Tag := Ord(7,*); btnKey7.Tag := Ord(7,*); btnKey0.Tag := Ord(7,*); btnKey0.Tag := Ord(7,*); btnKey0.Tag := Ord(1,*); btnKey8.Tag := Ord(1,*);	<pre>inabled</pre>		DragMode dmManual	
$\begin{array}{c c c c c c c c c c c c c c c c c c c $	Y-Z1 var yar yar yar procedure TfrmkeyBoard: TfrmkayBoard; DeActivateForm: TForm; implementation Y-Z2 procedure TfrmkeyBoard.FormCreate(Sender: TObject); begin // TBitBinOTagブロバティを設定 binkey7.Tag := Ord('7'); binkey7.Tag := Ord('7'); binkey7.Tag := Ord('7'); binkey8.Tag := Ord('8'); binkey0.Tag := Ord('8'); binkey8.Tag := S08; binhex1.Tag := O; binkey8.Tag := Ord('8'); binkey8.Tag := Ord('8'); binkey8.Tag := Ord('8'); binkey8.Tag := S08; binhex1.Tag := O; binkey8.Tag := S08; binkey8.Tag := Ord('8'); binkey8.Tag := Ord('8'); binkey8.Tag := Ord('8'); binkey8.Tag := S08; binkey8.Tag := S08; binkey8.Tag := S08		Enabled V True	
Yundy Description Height 348 Height 348 Yar frakayBoard: TfrmWayBoard; DeActivateForm: TForm; implementation Y-Z2 procedure TfrmKeyBoard.FormCreate(Sender: TObject); begin // TBitBtnO Tag プロパティを設定 btnKey7.Tag :: Ord('7'); btnKey7.Tag :: Ord('7'); btnKey7.Tag :: Ord('1'); btnKey7.Tag :: Ord('1'); btnKey8.Tag :: \$0: btnNext.Tag :: O;; btnNext.Tag :: O;; </td <td><pre>y-Joint The property interval inte</pre></td> <td></td> <td>FormStyle fsStavOnTon</td> <td></td>	<pre>y-Joint The property interval inte</pre>		FormStyle fsStavOnTon	
Height 348 Y-Z1 Var frakeyBoard: IfrmKeyBoard; DeActivateForm: TForm; implementation Y-Z2 procedure TfrmKeyBoard.FormCreate(Sender: TObject); begin // TBitBtnO) Tag プロパティを設定 btnKey7.1ag btnKey7.1ag btnKey7.1ag implementation // TBitBtnO) Tag プロパティを設定 btnKey7.1ag btnKey7.1ag imkey8.Tag imkey0.Tag imtkey0.Tag imtkey0.T	<pre>Height HeigContext 0 Y-7.1 Y=7.1 Y=T Defact ivateForm: Tform; implementation Y7.2 procedure IfrmKeyBoard.FormCreate(Sender: TObject); begin // TBitBtnOpTag/ID/iFr < EB:2 btnKey4.Tag := 0rd('7'); btnKey4.Tag := 0rd('1'); btnKey4.Tag := 0rd('1'); btnKey5.Tag := 0rd('1'); btnKey5.Tag := 0rd('1'); btnKey5.Tag := 0rd('2'); btnKey5.Tag := 0rd('0'); btnKey5.Tag := 0rd('0'); btnKey8.Tag := 0rd</pre>			
PrepContext 0 ソース1	<pre>HepContext HepContext HepContext V-71 var frmKayBoard-IfrmKayBoard; DeActivateForm: TForm; implementation /-72 procedure IfrmKeyBoard-FormCreate(Sender: TObject); begin binKey7.lag := 0rd('7'); binKey4.lag := 0rd('7'); binKey4.lag := 0rd('1'); binKey0.lag := 0rd('0'); binKey0.lag := 0rd(</pre>		Height 348	
<pre>Y-ス1 Var f_mKayBoard- TfrmKayBoard; DeActivateForm: TForm; implementation //-Z2 procedure TfrmKeyBoard.FormCreate(Sender: TObject); begin // TBitBtnのTagプロパティを設定 btnKey7.Tag := 0rd('7'); btnKey4.Tag := 0rd('1'); btnKey0.Tag := 0rd('1'); btnKey0.Tag := 0rd('1'); btnKey0.Tag := 0rd('5'); btnKey0.Tag := 0rd('5'); btnKey0.Tag := 0rd('5'); btnKey0.Tag := 0rd('0'); btnKey8.Tag := 0; btnPrior.Tag := 1; // ソフトウェアキーボーFO初期表示位置を設定 Top := Screen.WorkArealop + Screen.WorkArealight - Height; Left := Screen.WorkArealop + Screen.WorkArealight - Width; btnKey8.Tag := 0rd('0'); btnKey8.Tag := 0; btnPrior.Tag := 1; // ソフトウェアキーボーFO初期表示位置を設定 Top := Screen.WorkArealop + Screen.WorkArealight - Width; btnKey8.Tag := 0; btnPrior.Tag := 0; bt</pre>	<pre>Y-ス1 var f=rkKayBoard: TfrmKayBoard; DeActivateForm: TForm; implementation /-Z2 procedure TfrmKeyBoard.FormOreate(Sender: TObject); begin //TBitBtnのTagプロパティを設定 btnKey7.Tag := Ord('7'); btnKey1.Tag := Ord('4'); btnKey1.Tag := Ord('4'); btnKey0.Tag := Ord('4'); btnKey0.Tag := Ord('5'); btnKey0.Tag := Ord('5'); btnKey0.Tag := Ord('6'); btnKey0.Tag := Ord('8'); btnKey0.Tag := Ord('6'); btnKey0.Tag := Ord('8'); b</pre>		HelpContext 0	
y-Z1 var r=mKayBoard: IfrmKayBoard; DeActivateForm: IForm; implementation y-Z2 procedure TfrmKeyBoard.FormCreate(Sender: TObject); begin // TBitBtnのTagプロバティを設定 binKey7.Tag := Ord('7'); binKey7.Tag := Ord('1'); binKey0.Tag := Ord('1'); binKey0.Tag := Ord('1'); binKey0.Tag := Ord('0'); binKey0.Tag := Ord('0');	<pre>v-z1 var frmKayBoard: IfrmKayBoard; DeActivateForm: TForm; implementation v-z2 procedure IfrmKeyBoard.FormCreate(Sender: TObject); begin // TBitBtnのTagプロパティを設定 btnKey1.Tag := 0rd('7'); btnKey1.Tag := 0rd('1'); btnKey0.Tag := 0rd('0'); btnKey5.Tag := 0rd('0'); btnKey5.Tag := 0rd('5'); btnKey5.Tag := 0rd('2'); btnKey8.Tag := 0rd('?'); btnNext.Tag := 0; btnNext.Tag := 1; // ソフトウェアキーボーFの初期表示位置を設定 Top := Screen.WorkAreaLeft + Screen.WorkAreaWidth - Width; end;</pre>			
<pre>var frmKayBoard - IfrmKayBoard; DeActivateForm: TForm; implementation yZ2 procedure IfrmKeyBoard.FormCreate(Sender: TObject); begin // TBitBtnのTagブロバティを設定 btnKey7.Tag := Ord('7'); btnKey7.Tag := Ord('7'); btnKey4.Tag := Ord('7'); btnKey0.Tag := Ord('0'); btnKey0.Tag := Ord('1'); btnKey0.Tag := Ord('0'); btnKey0.Tag := Ord('1'); btnKey0.Tag := Ord('1'); b</pre>	<pre>var frekkowBoard: TfrewKeyBoard; DeActivateForm: TForm; implementation</pre>	ソ-		
DeActivateForm: TForm; implementation 'y-Z2 procedure TfrmKeyBoard.FormCreate(Sender: TObject); begin // TBitBtnのTagプロバティを設定 btnKey7.Tag btnKey7.Tag ::::::::::::::::::::::::::::::::::::	DeActivateForm: TForm; implementation Y-Z2 procedure IfrmKeyBoard.FormCreate(Sender: TObject); begin // TBitBotnOTagプロパティを設定 btnKey7.Tag btnKey7.Tag implementation // TBitBotnOTagプロパティを設定 btnKey7.Tag implementation // TBitBotnOTagプロパティを設定 btnKey7.Tag btnKey1.Tag implementation btnKey1.Tag btnKey1.Tag implementation btnKey1.Tag implementation btnKey2.Tag implementation btnKey5.Tag implementation implementation btnKey5.Tag implementation btnKey8.Tag implementation btnKey8.Tag implementation btnKey8.Tag implementation btnKey8.Tag implementation btnKey8.Tag implementation btnKey8.Tag implementation implementation btn		var fan KouPeende Ifan KouPeende	
implementation	implementation ソース2 procedure IfrmKeyBoard.FormCreate(Sender: TObject); begin // TBitBtnのTagブロバティを設定 btnKey7.Tag btnKey7.Tag intKey4.Tag intKey4.Tag intKey0.Tag		DeActivateForm: TForm;	
ソース2 procedure TfrmKeyBoard.FormCreate(Sender: TObject); begin // TBitBtnのTagプロパティを設定 btnKey1.Tag btnKey4.Tag := Ord('7'); btnKey1.Tag btnKey1.Tag := Ord('7'); btnKey1.Tag btnKey1.Tag := Ord('7'); btnKey1.Tag := Ord('8'); btnKey2.Tag := Ord('8'); btnKey0.tag := Ord('8'); btnKey0.tag := Ord('8'); btnKey0.tag := Ord('8'); btnKey0.tag := Ord('8'); btnKey8.Tag := Ord('8'); btnKey8.Tag := Ord('8'); btnKey8.Tag := Ord('8'); btnKey8.Tag := Ord('8'); btnNext.Tag := Ord; := Storeen.WorkArealop := Storeen.WorkArealop := Screen.WorkArealoft + Screen.WorkArealeight - Height;	<pre>ソース2 procedure TfrmKeyBoard.FormCreate(Sender: TObject); begin // TBitBtnのTagプロバティを設定 btnKey7.Tag := Ord('7'); btnKey4.Tag := Ord('4'); btnKey0.Tag := Ord('1'); btnKey0.Tag := Ord('8'); btnKey5.Tag := Ord('2'); btnKey2.Tag := Ord('2'); btnKey2.Tag := Ord('8'); btnKey8.Tag := Ord('8'); btnKey8.Tag := Ord('8'); btnKey8.Tag := Ord('8'); btnKey8.Tag := Ord('3'); btnKey8.Tag := Ord('3'); btnKey8.Tag := 0; btnPrior.Tag := 1; // ソフトウェアキーボードの初期表示位置を設定 Top := Screen.WorkAreaTop + Screen.WorkAreaWidth - Width; end;</pre>		implementation	
<pre>ソース2 procedure IfrmKeyBoard.FormCreate(Sender: TObject); begin // TBitBtnのTagブロバティを設定 btnKey7.Tag := Ord('7'); btnKey4.Tag := Ord('1'); btnKey0.Tag := Ord('1'); btnKey0.Tag := Ord('1'); btnKey0.Tag := Ord('2'); btnKey5.Tag := Ord('2'); btnKey2.Tag := Ord('2'); btnKey0.t.Tag := Ord('2'); btnKey0.t.Tag := Ord('3'); btnKey8.Tag := Ord('1'); btnKey8.Tag := Ord('1'); btnKey8.Tag := Ord('1'); btnKey8.Tag := S08; btnNext.Tag := 0; btnPior.Tag := 1; // Y OF + D = F + ポ - F O 初期表示位置を設定 Top := Screen.WorkAreaTop + Screen.WorkAreaHeight - Height; Left := Screen.WorkAreaLeft + Screen.WorkAreaWidth - Width; </pre>	ソース2 procedure TfrmKeyBoard.FormCreate(Sender: TObject); begin // TBitBtnのTagブロバティを設定 btnKey7.Tag := Ord('7'); btnKey4.Tag := Ord('1'); btnKey1.Tag := Ord('1'); btnKey0.Tag := Ord('0'); btnKey8.Tag := Ord('5'); btnKey2.Tag := Ord('2'); btnKey8.Tag := Ord('2'); btnKey8.Tag := Ord('3'); btnKey8.Tag := Ord('3'); btnKey8.Tag := Ord('3'); btnKey8.Tag := Ord('3'); btnKeyB.Tag := Ord('-'); btnKeyB.Tag := Ord('-'); btnKeyB.Tag := 0; btnPrior.Tag := 1; // ソフトウェアキーボー下の初期表示位置を設定 Top := Screen.WorkArealog + Screen.WorkAreaWidth - Width; end;			
procedure TfrmKeyBoard.FormCreate(Sender: TObject); begin // TBitBtnのTagプロバティを設定 btnKey7.Tag := Ord('2'); btnKey4.Tag := Ord('1'); btnKey0.Tag := Ord('0'); btnKey0.Tag := Ord('0'); btnKey2.Tag := Ord('0'); btnKey2.Tag := Ord('2'); btnKey0.Tag := Ord('0'); btnKey0.Tag := Ord('0'); btnKey8.Tag := Soreen.WorkAreaTop + Screen.WorkAreaHeight - Height; Left := Screen.WorkAreaLeft + Screen.WorkAreaWidth - Width;	<pre>procedure TfrmKeyBoard.FormCreate(Sender: TObject); begin // TBitBtnのTagプロパティを設定 btnKey7.Tag := Ord('7'); btnKey1.Tag := Ord('4'); btnKey1.Tag := Ord('0'); btnKey0.Tag := Ord('0'); btnKey0.Tag := Ord('0'); btnKey2.Tag := Ord('2'); btnKey0.tag := Ord('2'); btnKey0.tag := Ord('0'); btnKey0.tag := Ord('0'); btnKey8.Tag := Ord('0'); btnRey8.Tag := Ord('1'); btnRey8.Tag := Ord('1'); btnRey8.</pre>	ソ-	-72	
bigin // TBitBtnのTagブロバティを設定 btnKey7.Tag := Ord('7'); btnKey7.Tag := Ord('4'); btnKey1.Tag := Ord('1'); btnKey0.Tag := Ord('0'); btnKey0.Tag := Ord('0'); btnKey0.Tag := Ord('5'); btnKey0.Tag := Ord('5'); btnKey0.Tag := Ord('2'); btnKey0.tag := Ord('3'); btnKey8.Tag := Ord('3'); btnKey8.Tag := Ord('-'); btnKey8S.Tag := 0; btnPrior.Tag := 1; // ソフトウェアキーボードの初期表示位置を設定	<pre>begin // TBitBtnのTagブロバティを設定 btnKey7.Tag := Ord('7'); btnKey4.Tag := Ord('1'); btnKey0.Tag := Ord('8'); btnKey5.Tag := Ord('8'); btnKey5.Tag := Ord('8'); btnKey2.Tag := Ord('2'); btnKey9.Tag := Ord('?'); btnKey8.Tag := Ord('9'); btnKey8.Tag := Ord('8'); btnKey8.Tag := Ord('1'); btnKey8.Tag := Ord('1'); btnKey</pre>		procedure IfrmKeyBoard FormCroate(Sender: TObject):	
<pre>// IBitBtn@lagJDIバティを設定 btnKey7.Tag := Ord('7'); btnKey4.Tag := Ord('4'); btnKey1.Tag := Ord('1'); btnKey0.Tag := Ord('0'); btnKey0.Tag := Ord('5'); btnKey2.Tag := Ord('5'); btnKey2.Tag := Ord('2'); btnKeyDot.Tag := Ord('1'); btnKey0.Tag := Ord('0'); btnKey8.Tag := Ord('0'); btnKey8.Tag := Ord('0'); btnKey8.Tag := Ord('3'); btnKey8.Tag := Ord('3'); btnKeyB.Tag := Ord('3'); btnKeyB.Tag := 0rd('1'); btnKeyB.Tag := Ord('1'); btnKeyB.Tag := O; btnPrior.Tag := 1; // ソフトウェアキーボードの初期表示位置を設定 Top := Screen.WorkAreaTop + Screen.WorkAreaHeight - Height; Left := Screen.WorkAreaLeft + Screen.WorkAreaWidth - Width;</pre>	// TBitBtnOp Tag フロバティを設定 btnKey7.Tag := Ord('7'); btnKey4.Tag := Ord('4'); btnKey0.Tag := Ord('1'); btnKey0.Tag := Ord('0'); btnKey5.Tag := Ord('5'); btnKey2.Tag := Ord('2'); btnKey0t.Tag := Ord('2'); btnKey8.Tag := Ord('6'); btnKey8.Tag := Ord('3'); btnKey8.Tag := Ord('3'); btnKeyB5.Tag := \$08; btnKeyB5.Tag := \$08; btnNext.Tag := 0; btnPrior.Tag := 1; // ソフトウェアキーボードの初期表示位置を設定 Top := Screen.WorkArealoft + Screen.WorkAreaWidth - Width; end;		begin	
btnKey4.Tag := Ord('4'); btnKey1.Tag := Ord('1'); btnKey0.Tag := Ord('0'); btnKey0.Tag := Ord('5'); btnKey5.Tag := Ord('5'); btnKey2.Tag := Ord('2'); btnKey0.Tag := Ord('.'); btnKey8.Tag := Ord('6'); btnKey8.Tag := Ord('6'); btnKey8.Tag := Ord('-'); btnKeyBS.Tag := \$08; btnNext.Tag := O; btnPrior.Tag := 1; // ソフトウェアキーボードの初期表示位置を設定 Top := Screen.WorkArealop + Screen.WorkAreaWidth - Width;	btnKey4.Tag := Ord('4'); btnKey1.Tag := Ord('1'); btnKey0.Tag := Ord('0'); btnKey0.Tag := Ord('5'); btnKey2.Tag := Ord('5'); btnKey2.Tag := Ord('2'); btnKey0ot.Tag := Ord('0'); btnKev8.Tag := Ord('6'); btnKey8.Tag := Ord('6'); btnKey8s.Tag := Ord('6'); btnKey8s.Tag := Ord('6'); btnKey8s.Tag := Ord('6'); btnKey8s.Tag := 1; // ソフトウェアキーボードの初期表示位置を設定 Top := Screen.WorkAreaTop + Screen.WorkAreaHeight - Height; Left := Screen.WorkAreaLeft + Screen.WorkAreaWidth - Width; end;		// TBitBtnのTagブロバティを設定 btnKev7.Tag := Ord('7');	
bunkey1.1ag := Ord('0'); btnKey8.Tag := Ord('0'); btnKey5.Tag := Ord('5'); btnKey5.Tag := Ord('2'); btnKey0.tag := Ord('.'); btnKey8.Tag := Ord('6'); btnKey3.Tag := Ord('6'); btnKey8.Tag := Ord('6'); btnKey8.Tag := Ord('-'); btnKeyBS.Tag := \$08; btnNext.Tag := 0; btnPrior.Tag := 1; // ソフトウェアキーボードの初期表示位置を設定 Top := Screen.WorkAreaTop + Screen.WorkAreaHeight - Height; Left := Screen.WorkAreaLeft + Screen.WorkAreaWidth - Width;	binKeyD.Tag := Ord('0'); binKeyO.Tag := Ord('0'); binKeyS.Tag := Ord('2'); binKeyS.Tag := Ord('2'); binKeyOot.Tag := Ord('?'); binKeyS.Tag := Ord('6'); binKeyS.Tag := Ord('3'); binKeyBS.Tag := Ord('-'); binNext.Tag := O; binNext.Tag := 0; binNext.Tag := 1; // ソフトウェアキーボーFの初期表示位置を設定 Top := Screen.WorkAreaTop + Screen.WorkAreaHeight - Height; Left := Screen.WorkAreaLeft + Screen.WorkAreaWidth - Width; end;		btnKey4.Tag := Ord('4'); btnKey1.Tag := Ord('1');	
btnKey8.Tag := Ord('8'); btnKey5.Tag := Ord('5'); btnKey2.Tag := Ord('2'); btnKey0ot.Tag := Ord('.'); btnKey6.Tag := Ord('6'); btnKey6.Tag := Ord('8'); btnKey3.Tag := Ord('3'); btnKeyMinus.Tag := Ord('-'); btnKeyBS.Tag := \$08; btnNext.Tag := 0; btnPrior.Tag := 1; // ソフトウェアキーボードの初期表示位置を設定 Top := Screen.WorkAreaTop + Screen.WorkAreaHeight - Height; Left := Screen.WorkAreaLeft + Screen.WorkAreaWidth - Width;	btnKey8.Tag := Ord('8'); btnKey5.Tag := Ord('5'); btnKey2.Tag := Ord('2'); btnKeyDot.Tag := Ord('.'); btnKey8.Tag := Ord('6'); btnKey8.Tag := Ord('6'); btnKey8.Tag := Ord('3'); btnKeyBS.Tag := Ord('-'); btnKeyBS.Tag := \$08; btnPrior.Tag := 1; // ソフトウェアキーボードの初期表示位置を設定 Top := Screen.WorkAreaTop + Screen.WorkAreaHeight - Height; Left := Screen.WorkAreaLeft + Screen.WorkAreaWidth - Width; end;		btnKey0.Tag := Ord('0');	
btnKey5.Tag := Ord('5'); btnKey2.Tag := Ord('2'); btnKeyDot.Tag := Ord('8'); btnKey8.Tag := Ord('8'); btnKey8.Tag := Ord('8'); btnKey3.Tag := Ord('3'); btnKeyMinus.Tag := Ord('-'); btnKeyBS.Tag := \$08; btnNext.Tag := 0; btnPrior.Tag := 1; // ソフトウェアキーボードの初期表示位置を設定 Top := Screen.WorkAreaTop + Screen.WorkAreaHeight - Height; Left := Screen.WorkAreaLeft + Screen.WorkAreaWidth - Width;	btnKey5.Tag := Ord('5'); btnKey2.Tag := Ord('2'); btnKey0.Tag := Ord(''); btnKey6.Tag := Ord('6'); btnKey6.Tag := Ord('6'); btnKey3.Tag := Ord('-'); btnKeyMinus.Tag := Ord('-'); btnNext.Tag := 0; btnPrior.Tag := 1; // ソフトウェアキーボードの初期表示位置を設定 Top := Screen.WorkAreaTop + Screen.WorkAreaHeight - Height; Left := Screen.WorkAreaLeft + Screen.WorkAreaWidth - Width; end;		htnKev8.Tag := Ord('8'):	
btnKey2.lag := Ord('2'); btnKeyDot.Tag := Ord('9'); btnKey6.Tag := Ord('6'); btnKey3.Tag := Ord('6'); btnKey3.Tag := Ord('3'); btnKeyMinus.Tag := Ord('-'); btnKeyBS.Tag := \$08; btnNext.Tag := 0; btnPrior.Tag := 1; // ソフトウェアキーボードの初期表示位置を設定 Top := Screen.WorkAreaTop + Screen.WorkAreaHeight - Height; Left := Screen.WorkAreaLeft + Screen.WorkAreaWidth - Width;	btnKey2.lag := Ord('2'); btnKeyDot.Tag := Ord('9'); btnKey8.Tag := Ord('8'); btnKey8.Tag := Ord('8'); btnKey3.Tag := Ord('3'); btnKeyMinus.Tag := Ord('-'); btnKeyBS.Tag := \$08; btnPrior.Tag := 0; btnPrior.Tag := 1; // ソフトウェアキーボードの初期表示位置を設定 Top := Screen.WorkAreaTop + Screen.WorkAreaHeight - Height; Left := Screen.WorkAreaLeft + Screen.WorkAreaWidth - Width; end;		btnKey5.Tag := Ord('5');	
btnKey9.Tag := Ord('9'); btnKey6.Tag := Ord('6'); btnKey3.Tag := Ord('3'); btnKeyMinus.Tag := Ord('-'); btnKeyBS.Tag := \$08; btnNext.Tag := 0; btnPrior.Tag := 1; // ソフトウェアキーボードの初期表示位置を設定 Top := Screen.WorkAreaTop + Screen.WorkAreaHeight - Height; Left := Screen.WorkAreaLeft + Screen.WorkAreaWidth - Width;	btnKey9.Tag := Ord('9'); btnKey6.Tag := Ord('6'); btnKey3.Tag := Ord('3'); btnKeyMinus.Tag := Ord('-'); btnKeyBS.Tag := \$08; btnPrior.Tag := 0; btnPrior.Tag := 1; // ソフトウェアキーボードの初期表示位置を設定 Top := Screen.WorkAreaTop + Screen.WorkAreaHeight - Height; Left := Screen.WorkAreaLeft + Screen.WorkAreaWidth - Width; end;		btnKey2.lag := Urd(2'); btnKeyDot.Tag := Ord('');	
btnKey6.Tag := Ord('6'); btnKey8.Tag := Ord('3'); btnKeyMinus.Tag := Ord('-'); btnNext.Tag := 0; btnPrior.Tag := 1; // ソフトウェアキーボードの初期表示位置を設定 Top := Screen.WorkAreaTop + Screen.WorkAreaHeight - Height; Left := Screen.WorkAreaLeft + Screen.WorkAreaWidth - Width;	btnKey6.Tag := Ord('6'); btnKey3.Tag := Ord('3'); btnKeyMinus.Tag := Ord('-'); btnKeyBS.Tag := \$08; btnNext.Tag := 0; btnPrior.Tag := 1; // ソフトウェアキーボードの初期表示位置を設定 Top := Screen.WorkAreaTop + Screen.WorkAreaHeight - Height; Left := Screen.WorkAreaLeft + Screen.WorkAreaWidth - Width; end;		htpKov9 Tog ·= Ord('9').	
btnKey3.Tag := Ord([3]); btnKeyMinus.Tag := Ord('-'); btnNext.Tag := \$08; btnPrior.Tag := 0; btnPrior.Tag := 1; // ソフトウェアキーボードの初期表示位置を設定 Top := Screen.WorkAreaTop + Screen.WorkAreaHeight - Height; Left := Screen.WorkAreaLeft + Screen.WorkAreaWidth - Width;	btnKey3.Tag := Ord('3'); btnKeyMinus.Tag := Ord('-'); btnKeyBS.Tag := \$08; btnPrior.Tag := 0; btnPrior.Tag := 1; // ソフトウェアキーボードの初期表示位置を設定 Top := Screen.WorkAreaTop + Screen.WorkAreaHeight - Height; Left := Screen.WorkAreaLeft + Screen.WorkAreaWidth - Width; end;		btnKey6.Tag := Ord('6');	
btnKeyBS.Tag := \$08; btnNext.Tag := 0; btnPrior.Tag := 1; // ソフトウェアキーボードの初期表示位置を設定 Top := Screen.WorkAreaTop + Screen.WorkAreaHeight - Height; Left := Screen.WorkAreaLeft + Screen.WorkAreaWidth - Width;	btnKeyBS.Tag := \$08; btnNext.Tag := 0; btnPrior.Tag := 1; // ソフトウェアキーボードの初期表示位置を設定 Top := Screen.WorkAreaTop + Screen.WorkAreaHeight - Height; Left := Screen.WorkAreaLeft + Screen.WorkAreaWidth - Width; end;		htpKov3 Tog ·= Ord('3')·	
btnKeyBS.lag := \$00; btnNext.Tag := 0; btnPrior.Tag := 1; // ソフトウェアキーボードの初期表示位置を設定 Top := Screen.WorkAreaTop + Screen.WorkAreaHeight - Height; Left := Screen.WorkAreaLeft + Screen.WorkAreaWidth - Width;	btnKeyBS.Tag := \$08; btnNext.Tag := 0; btnPrior.Tag := 1; // ソフトウェアキーボードの初期表示位置を設定 Top := Screen.WorkAreaTop + Screen.WorkAreaHeight - Height; Left := Screen.WorkAreaLeft + Screen.WorkAreaWidth - Width; end;		btnKeyMinus.Tag := Ord('-'):	
btnNext.Tag := 0; btnPrior.Tag := 1; // ソフトウェアキーボードの初期表示位置を設定 Top := Screen.WorkAreaTop + Screen.WorkAreaHeight - Height; Left := Screen.WorkAreaLeft + Screen.WorkAreaWidth - Width;	btnNext.Tag := 0; btnPrior.Tag := 1; // ソフトウェアキーボードの初期表示位置を設定 Top := Screen.WorkAreaTop + Screen.WorkAreaHeight - Height; Left := Screen.WorkAreaLeft + Screen.WorkAreaWidth - Width; end;		btnKeyMinus.Tag := Ord('-');	
// ソフトウェアキーボードの初期表示位置を設定 Top := Screen.WorkAreaTop + Screen.WorkAreaHeight - Height; Left := Screen.WorkAreaLeft + Screen.WorkAreaWidth - Width;	// ソフトウェアキーボードの初期表示位置を設定 Top := Screen.WorkAreaTop + Screen.WorkAreaHeight - Height; Left := Screen.WorkAreaLeft + Screen.WorkAreaWidth - Width; end;		btnKeyMinus.Tag := Ord('-'); btnKeyBS.Tag := \$08;	
// ソフトワェアキーボードの初期表示位置を設定 Top := Screen.WorkAreaTop + Screen.WorkAreaHeight - Height; Left := Screen.WorkAreaLeft + Screen.WorkAreaWidth - Width;	// ソフトワェアキーボードの初期表示位置を設定 Top := Screen.WorkAreaTop + Screen.WorkAreaHeight - Height; Left := Screen.WorkAreaLeft + Screen.WorkAreaWidth - Width; end;		btnKeyMinus.Tag := Ord('-'); btnKeyBS.Tag := \$08; btnNext.Tag := 0; btnPrior Tag := 1;	
Left := Screen.WorkAreaLeft + Screen.WorkAreaWidth - Width;	Left := Screen.WorkAreaLeft + Screen.WorkAreaWidth - Width; end;		btnKeyMinus.Tag := Ord('-'); btnKeyBS.Tag := \$08; btnNext.Tag := 0; btnPrior.Tag := 1;	
	ena;		btnKeyMinus.Tag := Ord('-'); btnKeyBS.Tag := \$08; btnNext.Tag := 0; btnPrior.Tag := 1; // ソフトウェアキーボードの初期表示位置を設定 Top := Screen.WorkAreaTop + Screen.WorkAreaHeight - Heigh	ı t :



TBitBtn	onClick	TBitBtn	onClick
btnKeyO	btnKeyBoardClick	btnKeyDot	btnKeyBoardCli
btnKey1	btnKeyBoardClick	btnKeyMinus	btnKeyBoardCli
btnKey2	btnKeyBoardClick	btnKeyBS	btnKeyBoardCli
btnKey3	btnKeyBoardClick	btnSelAll	btnSelectALL
btnKey4	btnKeyBoardClick	btnNext	btnFocusContro
btnKey5	btnKeyBoardClick	btnPrior	btnFocusContro
btnKey6	btnKeyBoardClick		•
btnKey7	btnKeyBoardClick		
btnKey8	btnKeyBoardClick		
btnKey9	btnKeyBoardClick		





```
ソース7
     procedure FormDeactivate(Sender: TObject);
procedure EditEnter(Sender: TObject);
    private
       { Private 宣言 }
ソース8
   procedure TForm1.EditEnter(Sender: TObject);
   var
     rCon, rFrm, rRes: TRect;
pCon: TPoint;
iBot: Integer;
   begin
      if (Sender is TWinControl) and (Assigned(frmKeyBoard)) then
      begin
        '/' コンポーネントの位置をスクリーン座標で求める
pCon.X := (Sender as TWinControl).Left;
pCon.y := (Sender as TWinControl).Top;
        pCon := Form1.ClientToScreen(pCon);
        // コンポーネントのRect
        rCon.Top
                      := pCon.Y;
        rCon.Left := pCon.X;
rCon.Height := (Sender as TWinControl).Height;
rCon.Width := (Sender as TWinControl).Width;
        // キーボードのRect
                    := frmKeyBoard.Top;
:= frmKeyBoard.Left;
        rFrm Top
        rFrm.Left
        rFrm.Height := frmKeyBoard.Height;
        rFrm.Width := frmKeyBoard.Width;
        // コンボーネントとフォームの重なりを求める
IntersectRect(rRes, rCon, rFrm);
        // 重なりが有った場合キーボードを移動
        if rRes.Height > 0 then
        begin

// キーボードを移動した結果のフォームの下位置を求める

// [TEditのTop] + [TEditのHeight] + [キーボードのHeight] + [調整値]
           iBot := rCon.Top + rCon.Height + frmKeyBoard.Height + 5;
           // キーボードがスクリーンの外にはみ出したかどうかを判断
if iBot > Screen.Height then
             frmKeyBoard.Top := rCon.Top - frmKeyBoard.Height - 5 // TEditの上に移動
           else
                                                                               // TEditの下に移動
             frmKeyBoard.Top := rCon.Top + rCon.Height + 5;
        end:
      end;
   end;
```



図19

[🕜 ソフトウェ	アキーボード		
	7	8	9	BS
	4	5	6	全選択
Sample Edit1	1	2	3	次項目
Edit2	0		-	前項目
Edit3				