資料編

目次

図1 載	数材の趣旨	••••P2
図2 幕	数材の目次	•••P3
図 3	1. ソースプログラムの基本記号	••••P4
図4	2. 基本プログラム作成	•••P5
図 5	3. 動作確認	••••P6
図6	4. 必要用語解説	••••P7
図7 2	Kcode Ver3.2 と Xcode Ver4 の表記の違い	••••P8
図 8	2 つの画面がボタンを押すことによってループするアプリケーション	••••P9
図9 8	story board の解説	••••P10
図10	ワークスペースの名称設定	••••P11
図11	BMI計算機	••••P12
図12	Story board におけるプログラムソースを組み込む方法	••••P13
図13	BMI 計算機プログラムソース画面	••••P14
図14	プログラム解説	\cdot · · · P15
図15	応用に使う予定だった電卓のプログラムソース	••••P16
図16	電卓のシミュレータ画面	••••P17
図17	改善後の BMI 計算機プログラムソース画面	••• P18
図18	応用問題の作成	••••P19
図19	応用問題作成後の画面推移見込み	••••P20
図20	四角形面積計算機	$\cdots P21$
図21	四角形面積計算機プログラムソース画面	••••P22
図22	三角形面積計算機	••••P23
図23	三角形面積計算機プログラムソース画面	••••P24
図24	数当てゲーム	••••P25
図25	数当てゲームプログラムソース画面	••••P26,27

図1 教材の趣旨

iPad 穼 10:53 AM			73 % 🔳
→イブラリ 注目 A.	A	Q	
iosで動くアプリを作る。			
iosでお馴染みのiPhone、iPadで動くアプリケーションは というアプリケーションを使うことで簡単に創作するこ きる。このXcodeでは基本objective-Cという言語が使われ ため、まず、objectve-Cについてまとめると↓	Xco とカ てし	ode 、で いる	
Objective-C(オブジェクティブシー)は、プログラミ グ言語の一種で、C言語をベースにSmalltalk型のオブジ クト指向機能を持たせた上位互換言語である。Object CはNeXT、Mac OS XのOSに標準付属する公式開発言 であり、OS Xのパッケージ版に開発環境がDVDで付属 るほか、ユーザ登録をすれば無償でダウンロードで る。現在では主にアップルのMac OS XやiOS上で動作 るアプリケーションの開発で利用される。	ミン ive-語すきす		
*Wikipedia	より	友粋	
ということでObjective-CはC言語がベースになってい Xcodeで使う上で最低限のC言語を学ぶ必要があるため、 ージからまず、C言語から解説していく。	るの	って	
1/44			

図2 教材の目次

iPad 🛜	5:33 PM	充電していません 📟
ライブラリ 再開		E
~	目次 ブックマーク メモ 一	
iosで動くアプリ	を作る。	1
C言語入門		2
Xcode入門1~	~ファイルの作成~	9
Xcode入門2、	~Story board~	12
Xcode入門3 ·	~BMI計算機~	20
Xcodeプログラ	ラム解説	34
Xcode応用編・	~面積計算機~	39
Xcode応用編・	~面積計算機2~	49
Xcode応用編・	~数当てゲーム~	52

図3 1. ソースプログラムの基本記号

		(C言語入門		
1.ソ-	ースプログラム	の基本詞	己号		
まず	、C言語のソー	スプログ	グラムで使われ	る記号	について
	記号	うのキャラ	ラクターとその読	み方	
+	プラス	-	マイナス、ハイフン	*	アスタリスク
/	スラッシュ	(¥)	バックスラッシュ (円)	<	小なり
>	大なり	=	イコール	&	アンド
	オア	,	シングルクォーテー ション	"	ダブルクォーテーシ ョン
%	バーセント	\$	ダラー	#	シャープ
_	アンダーバー	:	עחב	;	セミコロン
	ドット	,	カンマ	!	エクスクラメーショ ン
~	チルダ	^	ハット	[]	角かっこ
{}	波かっこ	0	丸かっこ	<>	カギかっこ

10:53 AM		
עּדּד	${}_{A}\mathbf{A}$	Q
2.基本プログラム作成		
次は簡単にC言語のプログラム構造とプログラム作品 でのプロセスを解説する。	戈から実 行	亍ま
まず、「ようこそC言語へ」などとただ文字を表記す ムを作ってみる。メモ帳や、Jebit Xでsample1という 作成し、そこに以下のように入力して欲しい。	トるプロ? 5ファイ)	グラ ルを
○ ○ ○ ○ ○ ○ sample1.c ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○		
{ printf("ようこそC言語へ!\n"); printf("C言語を始めましょう!\n"); //printf("こ言語を始めましょう!\n");←でも可 return 0; }		
100% ÷ //	<u>k</u>	
これがそれぞれ何を表しているかと言えば、まず緑	の部分。	
<pre>• include<stdio.h></stdio.h></pre>		
これはstdio.hファイルをこの場所に読み込むようニ 指示するもので、「スタンダード・アイ・オー・^ う。具体的には、プログラムの本体でprintfという問	1ンパイ 、ッダ」 2 関数を使・	ラに とい うた
めに必要としている。 · · ■ · · · · · · · · · · · · · · · · ·		

図5 3.動作確認



Pad 🛜 5:49 PM		充電して	ていません 🔳
(ライブラリ) ⋮Ξ A	A	Q	
4.必要用語解説			
クリック→ 対象をマウスボタンで押す。 ドラッグ→ 対象をクリックしながら移動させる。 ドロップ→ ドラッグした対象をマウスを離すことにより	配置	冒す	
る。 ドラッグ&ドロップ→ ドラッグとドロップの一連の動	作を	E指	
9。 ファイル→ 作成したデータ アイコン→ 絵で表記されたボタン			
リンク→ 繋かり、繋けるの意 変数→ プログラムのソースコードにおいて、扱われるデ	- 5	マを	
ー定期間記憶し必要なときに利用できるようにするため ータに固有の名前を与えたものである。 厳密には違うが、数学の文字(XやY)をイメージして欲	に、	デ い。	
クラス→ objectve-Cの設計図にあたるもの。今回は文字列 NSStringクラスと変数を利用出来るNSMutalbeStringクラ う2つのクラスを用いプログラミングを行う。使用箇所 グラム解説で説明する。	を払う	及う とい プロ	
※なお、挿入されている画像は二度タップすることで拡大 ます。またiPadの向きを変えることでも大きさが変わるの 分で見やすいように調整して下さい。	、で つで	き 自	
以上の点を踏まえ、早速Xcodeでアプリケーションを作成 こう。	しっ	てい	
 ・・・・・ ・・・・ ・・・・	・・ 後のペ	・・ ージ	



図7 Xcode Ver3.2 と XcodeVer4 の表記の違い

上が Ver3 シリーズ。

下が Ver 4 シリーズ。

表記が日本語から英語に変わっている。

図8 2つの画面がボタンを押すことによってループするアプリケーション

画面 A		画面 B		
		۲		
Carrier 🗢 6:05 PM		Carrier 🗢	6:05 PM	
			HelloOtani	
Move to Window		R	eturn to Window	
			\bigcirc	
	Л			

画面AとBがボタンを押すことでループする。

図9 Story board の解説



図10 ワークスペースの名称設定



図11 BMI計算機

•					
Carrier 🗢	7:10 PM	-	Carrier 🗢	7:11 PM	—
BMI值計算	機		BMI值計算	算機	
身長		cm	身長	170	cm
体重		kg	体重	60	kg
Ē	1算 リセット)		計算 リセ	۲
			BMI	値は 20.7612て	ぎす。
			標準体	本重は 63.58kg	なので
			٢Đ	里想の体重」で	す。
四角形面積	i 三角形面積 数	当て	四角形面	積三角形面積	数当て

図12 Story board におけるプログラムソースを組み込む方法

iPad 중	6:31 PM		充電して	いません 📟
ライブラリ 📃		лA	Q	
では、今配置したア 手順としては、	イコンをプログラムに組み込ん ⁻	でいく。		
①追加したいアイコ	ンを左クリック。			
②そのまま画面Aの。	ようにドラッグ&ドロップ。			
③画面BのようにNa す。	ume(=変数)とTypeを決め、	connect ?	を押	
これはそれぞれのア	イコンに対して変数を代入してい	いる。		
HODIC C	BMICAlculator.accedepte) — MainStoryboard storyboard Xade No None Provide Test Hall I I I I I I I I I I I I I I I I I I	View Onterface ViewC. () 12/12/83. 2812PyCospanyName troller : Outles, Action, or Outles		
	・ III ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	は残り10ペーシ	・・ です	

図13 プログラムソース画面



図14 プログラム解説

iPad ᅙ	6:33 PM		充電して	ていません 📟
7	イブラリ	лA	Q	
	Xcodeプログラム解説			
	ViewController.mでどんなことをやっているか解説する ViewController.mの画面と見比べながら見て欲しい。 まず計算処理の部分。	0		
	・- (IBAction) 変数 :(id)sender { 動作 }			
	これは、ViewController.hのIBActionとリンクしておりる 下の部分にそれぞれのボタンを押した動作を記述してい なる。 例えばBMI値計算機では変数「Touchkeisan」と「Touch のButtonに対して動作が記述されている。	これ。 ハく哥 hrese	より 事に et」	
	・// キーボードを隠す [self textFieldResignFirstResponder];			
	これは決まりで、//計算処理の上にキーボードを隠すタ 力している。 キーボードを隠す処理は決まった形なので覚えて欲しい それをButtonの動作にリンクさせている。	<u></u> 処理を	を入	
	では次の行。			
	・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・)4ページ	ਾਂ ਦ ਰ	

図15 応用に使う予定だった電卓のプログラムソース

00	0	🔄 dentaku.xcodeproj –	– ៣ dentaku	core.n	n			\square
) (dentaku) iOS Device	Build dentaku: Succeeded	12/11/05 at 1	6:13				
Run	Stop Scheme Breal	cpoints				Editor	View	Organizer
	🔹 🕨 📄 dentaku 👌 🧰 dentaku 👌 🖬 dentakucore.r	n > M -performOperation:			▶ 🛄 M > 🛃 > 🚞 >	dentakuViewCont	troller.m $ angle$ M –bo	otan: 🖸 🖸
	<pre>/ dentakucore.m / dentaku // Created by on 12/10/22. / Copyright (c) 2012 _MyCompanyName / !import "dentakucore.h" !interface dentakucore() !property (nonatomic, strong) NSMutableArr. !end !implementation dentakucore !synthesize operandStack = _operandStack;</pre>	All rights reserved. ay *operandStack;		// // // // // #imp @int @pro @end	dentakuViewControll dentaku Created by on 12/1 Copyright (c) 2012 reserved. wort "dentakuViewCon ort "dentakuVoewCon ort "dentakuViewC opperty (nonatomic) E userisinThetHiddleOf opperty (nonatomic, s	.er.m .ey.m .myCompanyName_	All rights	
	<pre>(NSMutableArray *)operandStack if (_operandStack == nil) _operandStac return _operandStack; (void)setOperandStack:(NSMutableArray *) _operandStack = operandStack; (double)popOperand MSNumber *operandObject = [self.operand if (operandObject) [self.operandStack return [operandObject doubleValue]; (void)pushOperand:(double)operand [self.operandStack addObject:[NSNumber (double)performOperation:(NSString *)ope double result = 0; if ([operation isEqualToString:@**"]){ result = [self popOperand] + [self lese if (loperation isEqualToString; double subtrahend = [self popOperand] - subtr } else if (loperation isEqualToString; double subtrahend = [self popOperand] - subtr } else if (loperation isEqualToString; double subtrahend = [self popOperand] if (divisor) result =[self popOperand] if (loperation isEqualToString; result = 0; } }</pre>	<pre>k = [[NSMutableArray alloc] init]; operandStack dStack lastObject]; removeLastObject]; numberWithDouble:operand]]; ration popOperand]; tion]){ popOperand]; e^]]{ nd]; ahend; e^'''']}; and] / divisor; e^ACC"]){</pre>	۲	@imp @syn @syn -{de } - (I } - (I } - (I } }	<pre>lementation dentaku thesize gamen = _ga thesize userlsInTheHiddleD thesize core = _co mtakucore *)core if (!_core) _core = return _core; BAction)botan:(UIBu NSString *digit, = s if (self.userlsInTh self.gamen.text self.userlsInTh } BAction)enterPressd [self.core pushOper self.userlsInTh [self.core pushOper if (self.userlsInTh [self.core pushOper self.userlsInTh [self.enterPress] NSString *operation double result = [s self.gamen.text = [result];</pre>	<pre>NiewController imen; #iddleOfEntering# #iddleOfEntering ifEnteringANumber; re; : [[dentakucore al itton *)sender :ender.currentTitl ieMiddleOfEntering;dig: := idigit; ieMiddleOfEnteringANur ressed: (UIButton * ieMiddleOfEnteringSNur ressed: (UIButton * ieMiddleOfEnteringSNur isd]; n = [sender currentIf.core perform0] NSString stringW.</pre>	<pre>ANumber = ; lloc] init]; le; gANumber){ ext ii]; text doubleVal mber = N0; *)sender gANumber){ ntTitle]; peration:operati</pre>	5; lue]]; ation];
)	[selt pushOperand:result]; return result; gend			@end	I			
								11.

図16 電卓のシミュレータ画面

•		
Carrier 🗢 6:43 PM	0	Carrier 중 6:44 PM ➡ 0123456789
7 8 9 4 5 6 1 2 3 AC 0 enter	*	7 8 9 * 4 5 6 / 1 2 3 + AC 0 enter -

図17 改善後の BMI 計算機プログラムソース画面

0	00	🖱 🗾 BMICalculator.xcodeproj — 🖻 Vie	wCo	ntro	roller.m 👄
(Build Succeeded Yesterday at 14	4:57		
R	un	Stop Scheme Breakpoints No Issues			Editor View Organizer
	•	🕨 📩 BMICalculator > 🧰 BMICalculator > 💼 ViewController.m > 🔟 -Touchkeisan:			🖁 📔 🔸 📄 🗋 👌 🧮 ခြုန် h ViewContr 🤇 No Selection 🗈 🖾
	//2 - (} //#	ここから書き足し void)textFieldResignFirstResponder { [heightText resignFirstResponder]; [weightText resignFirstResponder]; // 身長・体重テキストのキーボードを隠す処理 計算処理 TRestignTouchkeisen(id)sender {	0		<pre>// ViewController.h // BMICalculator // Created by on 12/12/03. // Copyright (c) 2012MyCompanyName All rights reserved. // fimmont dUTKit/UIKit back</pre>
	- `				Ginterface ViewController : UIViewController
		[self textFieldResignFirstResponder]; // 身長テキストフィールドに入力された値 (cm) をメートルに変換 float h eight = [heightText.text floatValue]/100;		۲	@property (weak, nontonic) IBOutlet UTExtField *heightText; //身長テキストフィールド
		// 体量テキストフィールドに入力された値を取得 float weight = [weightText.text floatValue];		۲	@property (weak, nonatomic) IBOutlet UITextField *weightText; //体重テキストフィールド
	ò	<pre>// BMI = 体重(kg) + (身長(m) × 身長(m)) float bmi = weight / (height * height); // BMI計算結果(実数を文字列に変換) NSString *bmiStr = [NSString stringWithFormat: @"%g", bmi]; // EMI+# (k_) = 食馬(-) = 食馬(-) = QMI=#### (co)</pre>		•	@property (weak, nonatomic) IBOutlet UIButton *keisanButton; //計算ボタン @property (weak, nonatomic) IBOutlet UIButton
		// 標平体重 (kg) = 身炎 (m) × 身炎 (m) × BMI標準値 (22) float stdWeight = (height * height) * 22;			*resetButton; //リセットボタン
		// 標準体重の計算結果 (実数を文字列に変換) NSString *stdWeightStr = [NSString stringWithFormat: @"%g", stdWeight]; // *tmWithFormat: @		۲	@property (weak, nonatomic) IBOutlet UILabel * hyoujiLabel; //表示ラベル
		אראבינוס NSString *result = @"";		۲	- (IBAction)Touchkeisan:(id)sender; //計算を押したときに呼ぶ動作
		<pre>// BMLD/18.51よりからければ、 **ビス(株) と表示する。 if (bmi < 18.5) { result = @" (やせ気味」です。"; // BMLD/18.5以上、25より小さければ、「理想の体重」と表示する。 } else if (bmi >= 18.5 & bmi < 25) { result = @" 「理想の体重」です。"; </pre>		۲	- (IBAction)Touchreset:(id)sender; //リセットを押したときに呼ぶ動作
		<pre>// それ以上の場合は、「肥満」と表示する。 } else { result = @"「肥満」です。"; }</pre>			@end
	}	// 判定結果を結果ラベルに設定 hyoujiLabel.text = [NSString stringWithFormat:@"%@%@%@%@%@%@%@", @"BMI値は ", bmiStr, @"です。", @"\n\n標準体重は ", stdWeightStr, @"kgなので\n\n", result];			
۲	//IJ - (Jセット処理 IBAction)Touchreset:(id)sender { // キーボードを聴す [self textFieldResignFirstResponder]; heightText.text = @""; hyoujiLabel.text = @""; //それぞれをクリア nd	4		
			4		

図18 応用問題の作成

iPad 🔶 6:33 PM	充電していません 📟
ライブラリ :=	
Xcode応用編 ~面積計算	[機~
応用として「BMICalculator」にどんどん う。 まず、入門2で作成したViewのリンク機能 にも実装し、手始めに『四角形の面積計算 「Round Rect Button」を図のように3つう 左から「四角形面積」「三角形面積」「数 たら新しい「View Controller」を追加。ie 形で「Label」「Button」「Text Fierd」 「BMI値の画面に戻る」というButtonを た「四角形面積」から新しい「View Con げる。同様に「BMI値の画面に戻る」から	 機能を追加していこ どを「BMICalculator」 送機』をつくる。 追加する。タイトルは 数当て」にする。出来 w内には、BMIで同じ 」を追加し、新しく 配置。その後、実装し troller」にリンクを繋 5、「BMI値」のViewに
The second and the second a	ard Learner Learne
39/61	この章は残り9ページです

図19 応用問題作成後の画面推移見込み



画面 A の 3 つのボタンを押すことで対応した画面 B、画面 C、画面 D の 3 種類の画面に変わる。

```
図20 四角形面積計算機
```

	•				
Carrier 🔶	6:49 PM		Carrier 🗢	6:49 PM	Î
四角形面	面積計算機		四角形面	積計算機	
縦		cm	縦	10	cm
横		cm	横	10	cm
(計算リセット)	四角形の	計算 リセット 縦10cmで 横10cmの の面積は100cm2 て	۴ す 。
	BMI値の画面に戻る		(BMI値の画面に戻る	

図21 四角形面積計算機プログラムソース画面

0	00	0			BMIC	alculator.xcode	proj — 🕅 m	nens	eki_	Q.m	_			\square
((B.) iPhone 5.0		Bu	ild Succeeded	Yesterday at 14	:57						
R	un	Stop	Scheme	Breakpoints		No Iss	sues				Editor	r	View	Organizer
	•	🕨 🛛 📩 BM	AICalculator $ angle$ 🔛 BM	ICalculator $ angle$ m n	menseki_Q.m $ angle$	C @implementati	ion menseki_Q			•	🕨 💷 > 🔂 > 🚞 > 🖿) menseki	_Q.h > No Sele	ction 🖸 🖸
	} - ({ } //2 - (}	<pre>[self se [self se [self se [super v // Relea // e.g. BOOL)shou interfac // Retur return (ccから書き!) void)text [tateTex [yokoTex // 縦・横</pre>	tKeisan_0:nil]; ttReset_0:nil]; ttReset_0:nil]; iseworking(0:nil); iseworking(0:nil); iseworking(0:nil); iseworking(0:nil); iseworking(0:nil); tseworking(0:nil); teOrientation m YES for suppo interfaceOrient 定し FieldResignFirstRe テキストのキーボード Touchkeisen 0:0	subviews of t nil; nterfaceOrient rted orientati ation == UIInt tResponder]; 安愿す処理 id)sender {	the main vie tation:(UIIn ions terfaceOrien	w. terfaceOrienta tationPortrait	ition)	6		// // // // // #im @pr @pr @pr @pr	menseki_S.Q BMICalculator Created by on 12, Copyright (c) 2012 rights reserved. mport <uikit b="" uikit<="">. aterface menseki_Q = roperty (weak, nonat UITextField *tate roperty (weak, nonat *keisan_Q; roperty (weak, nonat</uikit>	/12/15. 2MyCo h> : UIView tomic) I Text; tomic) I tomic) I tomic) I	mpanyName Controller BOutlet BOutlet BOutlet UII BOutlet UII	. All Button Button
۲	- (IBAction) // ギーボ [self te // 線テキ、 float ta NSString // 模テキ、 float yo NSString // 四角形 float me // 四角形 NSString // 判定結 hyauji_Q	Touchkeisan_Q:(ードを聴す extFieldResignFi ストフィールドに入力 ite = [tateText. *tateStr = [N ストフィールドに入力 kko = [yokoText. *yokoStr = [N の面積 = 縦 (cm) * enseki_Q = tate の面積計算結果 (実数) *menseki_QStr Ree結果ラベルに設ち).text = [NSStri @"W,h @"\n\n @"\n\n	id)sender { rstResponder]; isithtia (cm) を: text floatValu SString string isithtide実改と1) text floatValu SString string 模 (cm) * yoko; text yoko; [NSString string text floatValu string string string string string string text floatValu string str	実数と文字列に3 ye]; gWithFormat: 文字列に変換 yWithFormat: stringWithFo cormat:@"%@% "cmの", nenseki_QStr	لایگ @"%g", tate]; @"%g", yoko]; rmat: @"%g", m و%و%و%و%و%و% , @"cm2 تع: "	enseki_Q]; @",];			@pr - (- (@en	<pre>*reset_0; roperty (weak, nonat hyouji_0; (IBAction)Touchkeisa (IBAction)Touchreset ad</pre>	tomic) I an_Q:(id t_Q:(id)	BOutlet UII	.abel *
۲	- (} @en	IBAction) // キーボ [self te tateText yokoText hyouji_Q //それぞれ	Touchreset_Q:(i ードを聴す :xtFieldResignFi :.text = @""; .text = @""; .text = @""; .text = @""; .text = @"";	d)sender {										

図22 三角形面積計算機

•					
Carrier 🗢	6:49 PM		Carrier 🗢	6:49 PM	ļ
三角形面積	訂算機		三角形面	積計算機	
底辺		cm	底辺	10	cm
高さ(cm	高さ	10	cm
	計算 リセット)		計算 リセット	
				底辺10cmで	
				高さ10cmの	
			三角形	の面積は50cm2 で	゙す。
E	BMI値の画面に戻る			BMI値の画面に戻る	

図23 三角形面積計算機プログラムソース画面

0	00	🖱 🛃 BMICalculator.xcodeproj — 📠	mens	eki_	T.m	_		\square
(B) iPhone 5.0 Build Succeeded Yesterday at 1	4:57					
F	 un	Stop Scheme Breakpoints No Issues				Editor	View	Organizer
	•	Image: BMICalculator > BMICalculator > m menseki_T.m > No Selection			•	▶ □ > 🖄 > 🚞 > ┣ > 🖸 @	interface mens	seki_T 🖸 🛙
	} - ({ } //2 - (<pre>[self setTekhenText:nil]; [self setTakasaText:nil]; [self setKeisan_T:nil]; [self setReset_T:nil]; [self setReset_T:nil]; [super viewDidUnload]; // Release any retained subviews of the main view. // e.g. self.myOutlet = nil; BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation) interfaceOrientation // Return YES for supported orientations return (interfaceOrientation == UIInterfaceOrientationPortrait); CZから書き足し void)textFieldResignFirstResponder { [takasText resignFirstResponder]; [takasText resignFirstResponder]; [takasText resignFirstResponder]; [takasText resignFirstResponder]; // 成辺・高さテキストのキーボードを隠す処理 IBAction)Touchkeisan_T:(id)sender { // キーボードを隠す [self textFieldResignFirstResponder]; // キーボードを隠す [self textFieldResignFirstResponder]; // な辺テキストフィールドに入力された値 (m) を実数と文字列に変換 float teiben = [teibenText.text floatValue];</pre>		* * * *	// // // // // // // // // // // // //	<pre>menseki_T.h BMIGalculator Created by on 12/12/15. Copyright (c) 2012MyC rights reserved. port <uikit uikit.h=""> terface menseki_T : UIVie operty (weak, nonatomic) : UITextField *teihenText; operty (weak, nonatomic) : *keisan_T; operty (weak, nonatomic) : *reset_T; operty (weak, nonatomic) : hyouj_T; IBAction)Touchkeisan_T:(id IBAction)Touchreset_T:(id</uikit></pre>	ompanyName_ wController IBOutlet IBOutlet UI IBOutlet UI IBOutlet UI d)sender;)sender;	All
۲	}	<pre>// あさてもににしていたいでは、「いたいないないないないないないないないないないないないないないないないないない</pre>			(en	d		
	} @en	teihenText.text = @""; takasaText.text = @""; hyouj_T.text = @""; //それぞれをクリア) 4 V					1

図24 数当てゲーム



図25 数当てゲームプログラムソース画面

Durit	0	00	D				📩 BMICalcula	ator.xcodepro	oj — 🕅 ka	azuat	te.m					\Box
Nor Stop State Notation Image: Stop State State State State Image: Stop State State State	6			B) iPhone 5	5.0		Build Succ	eeded Yeste	rday at 14:5	7						
<pre>ml 4 b McGaladary McGaladary</pre>	R	√ , µn	Stop	Scheme	Brea	kpoints		No Issues				J	Edito	r	View	Organizer
<pre>//Z_DPSHEAL - (wid)tectfielderight:thesponder {</pre>		•		BMICalculator >	BMICalculat	or 👌 🖬 kazuat	e.m $ angle$ M –Touchr	reset_K:		•	▶ []	Manual	B.) 🛅 B.) h ka:	zuate.h 👌 P re	set_K 🛛 🕬
NSString *result = @""; // -の位が1、+の位が4でない場合「おしい。あと十の位だけだ!」と表示する。 if (ichi == 1 & 5 ju > 4 & 5 ju < 4) { result = @"「おしい。あと十の位だけだ!」"; // -の位が1、+の位が4ならば、「大正解!答は14だよ.」と表示する。 } else if (ichi == 1 & 5 ju == 4) { result = @"「大正解!答は41だよ。」"; // -の位が1でなく、+の位が4なら「おしい。あと一の位だけだ!」と表示する。	®	/n	<pre>Stop </pre> Stop Stop Stop Cash and a state of the state of t	Scheme BMCalculator) #BEU #BLCalculator) #BEU #BLCalculator) #BEU #BLCalculator) #BLCalculator) #BLCalculator) #BLCalculator #Comparent Action #Comparent	Breal BMICalculat nFirstResponder]; λトのキーボード n_K: (id) send ignFirstResp (ールドに入力さ Text.text fl (ールドに入力さ Text.text float) = @""; トさければ、「上 「上だよ」"; E、10より小さい && ichi < 1] (「たえ」"; にンゴ!」と表 1) { 「ビンゴ!」; 一桁の数字を入 「ビンゴ!」; 「ビンゴ!」とま (」とだよ」"; にたよ」; 「ビンゴ!」; 「ビンゴ!」; 「「たえ」]; 「ビンゴ!」; 「「たえ」]; 「「「たえ」]; 「」」 「「たえ」]; 「「たえ」]; 「「たえ」]; 「「「たえ」]; 「「」」]; 「「「」」]; 「「」」]; 「「」」]; 「「「た」]; 「「」」]; 「」」」]; 「」」」]; 「」」」]; 「」」]; 「」」」]; 「」」]; 「」」」]; 「「」」]; 「」」」]; 「」」」]; 「」」」]; 「」」」]; 「」」」]; 「」」]; 「」」」]; 「」」」]; 「「」」」]; 「」」]; 「」」」]; 「」」」]; 「」」」]; 「」」」]; 「」」」]; 「」」」]; 「」」」]; 「」」」]; 「」」」]; 「」」」]; 「」」」]; 「」」」]; 「」」」]; 「」」」]; 「」」」]; 「」」]; 「」」]; 「」」]; 「」」]; 「」」」]; 「」」」]; 「」」]; 「」」]; 「」」]; 「」」]; 「」」]; 」]; 「」」]; 」]; 」」]; 」」]; 」]; 」]; 」」]; 」];	kpoints kpoints (a) kazuat (a) kazuat (b) kazuat (c) (c) (c) (c)	e.m) M -Touchr る。 :」と表示する。 :」と表示する。 :」と表示する。	reset_K:		◀ ////////////////////////////////////	<pre>kazua BMICa Creat Creat Copyr reser ichiT operty keisa reset Hyouj IBActi IBActi IBActi</pre>	Manual) te.h lculator ed by o ight (c) ved. UIKit/UI e kazuat (weak, ext; (weak, n_K; (weak, i_K; (weak, i_K; on)Touch on)Touch	Edito Particle Construction (Construction) (Kit.h> (Kit.h> re : UIView nonatomic) nonatomic) nonatomic) nonatomic) ikeisan_K:(j)	Company Company Contro IBOut IBOut IBOut IBOut IBOut d)send	View zuate.h) [2] re yName All ller let UITextF: let UITextF: let UIButton let UIButton let UIButton er; er;	organizer eset_K 🖸 🕄 l rights ield * ield * n * *
			NSStr: // -0 if (id re // } else re	ing *result = の位が1、十の位が、 chi == 1 && j esult = @"「お / 一の位が1、十の e if (ichi == esult = @"「大 / 一の位が1でな	 @""; 4でない場合『お u > 4 && ju い。あと十の(の位が4ならば、 1 && ju == 正解!答は41だく、十の位が4な 	Sしい。あと十の < 4) { 立だけだ!」"; 『大正解!答は 4) { よ。」"; :ら『おしい。あ	位だけだ!」と表示 4だよ.」と表示 と一の位だけだ!」	でする。 する。 と表示する。	•							1

