

# Build a Concentration Game in Objective C

by Todd Bernhard

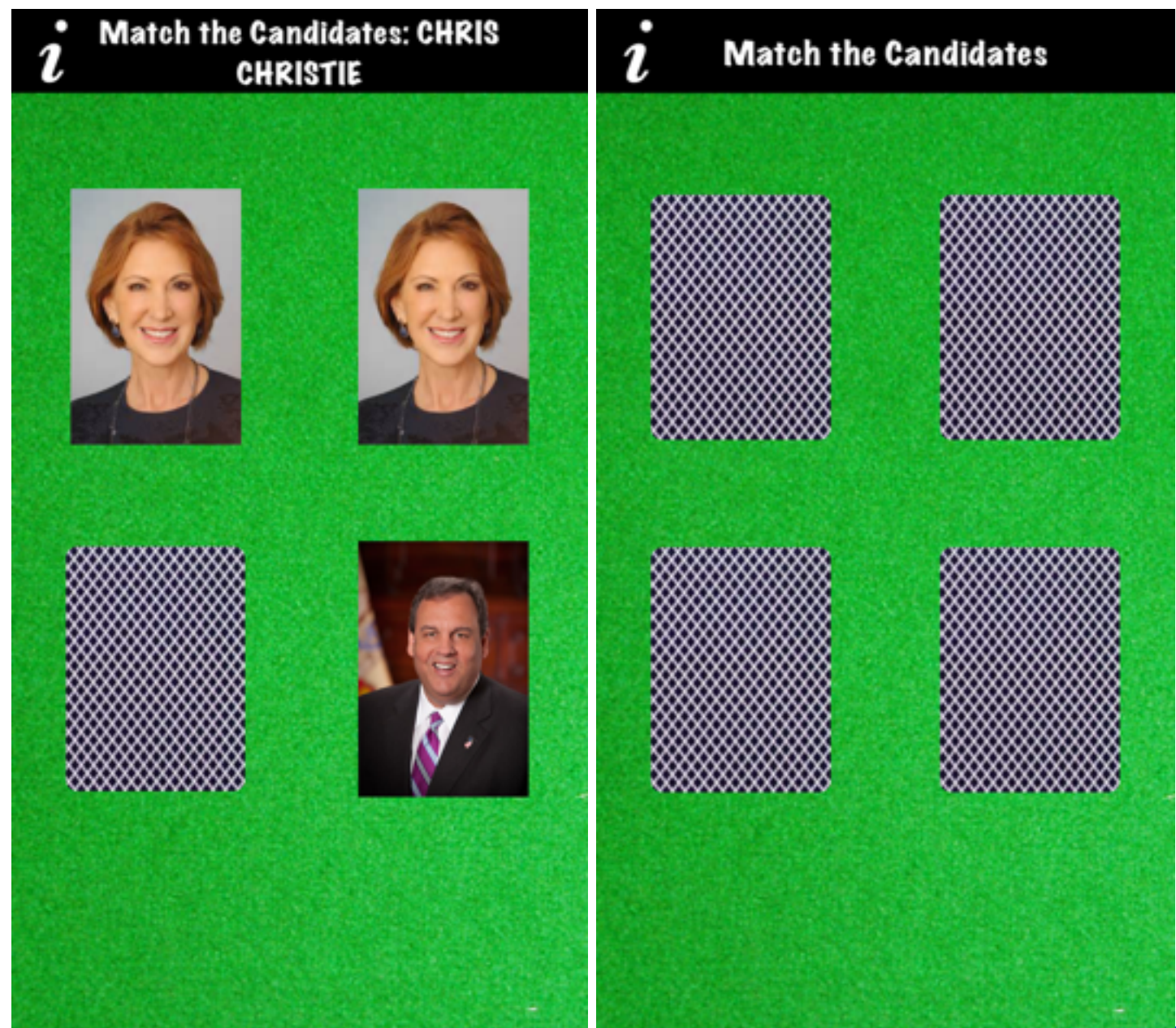


Todd Bernhard  
Todd@ToddBernhard.com  
@ToddBernhard

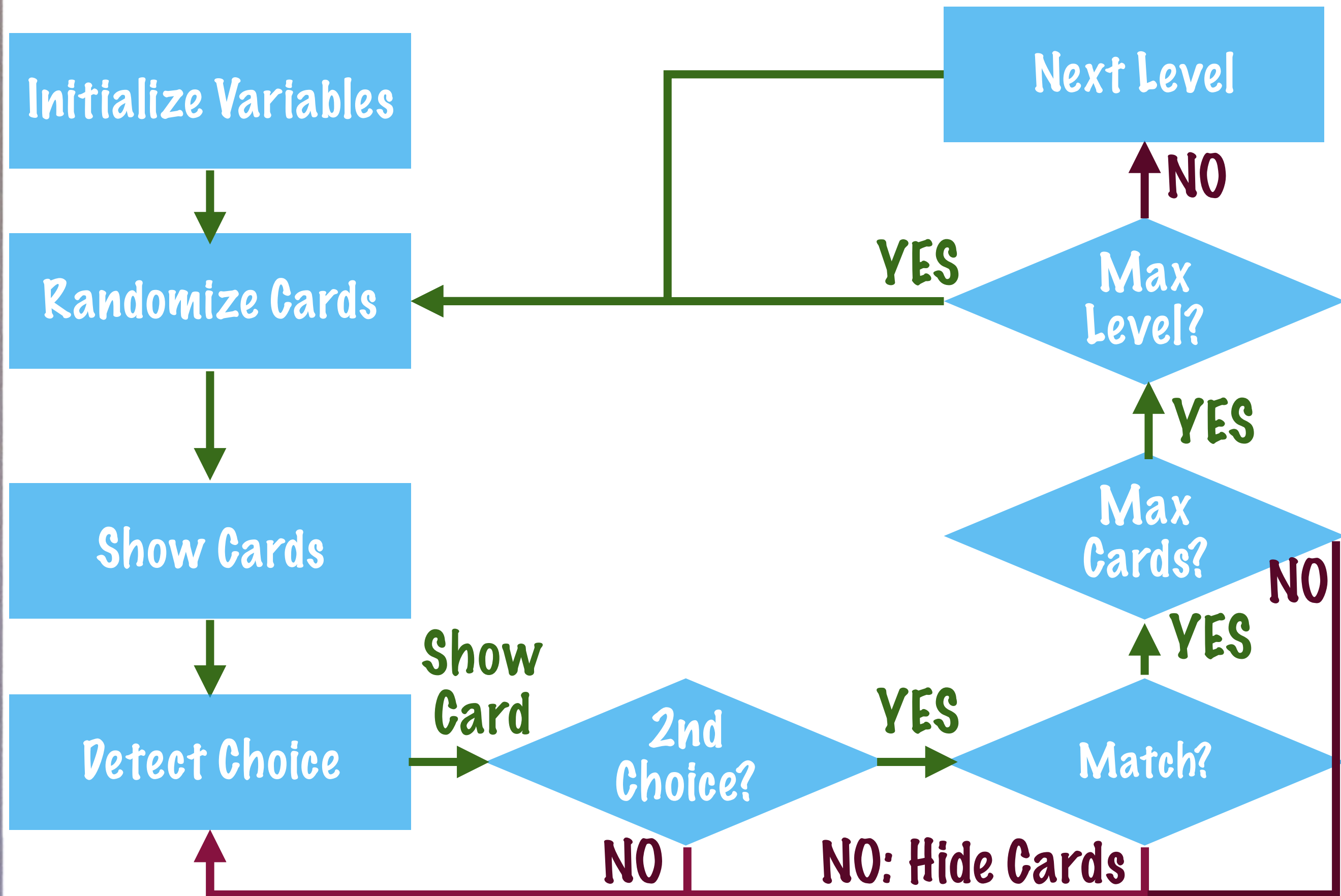


# Easy!

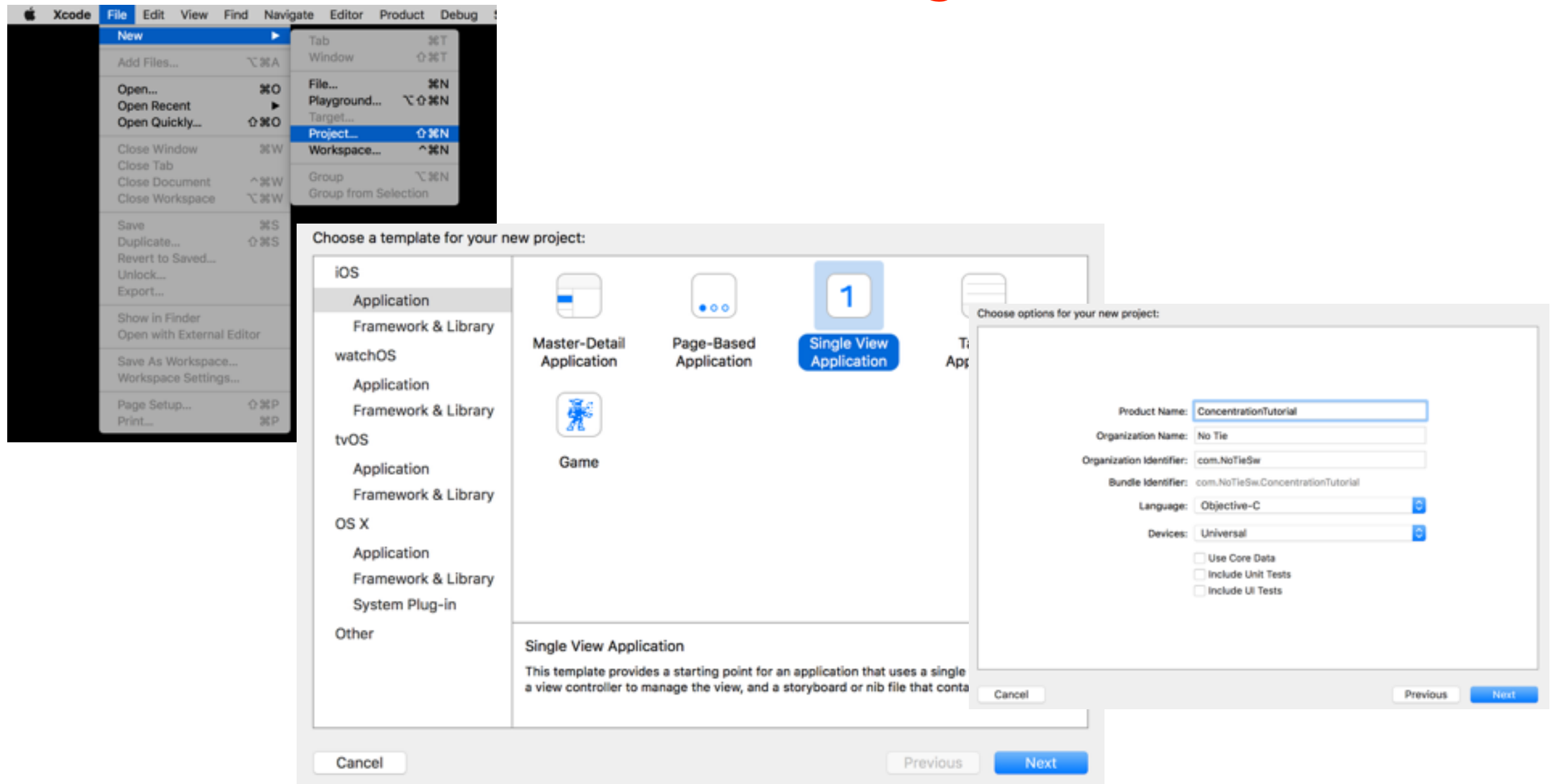
- Easy to Play
- Easy to Program
- Just add graphics!







# Step 1: Xcode, File -> New Project -> Single View



# Initialize Variables: Add to ViewController.m

```
- (void)viewDidLoad {  
    [super viewDidLoad];  
    // Do any additional setup after loading the view, typically from a  
    nib.  
    [self initializeVariables];  
}
```

# Initialize Variables

```
-(void)initializeVariables {
```

```
    level = 1;  
    baselevel = 1;  
    maxlevel = 10;
```

Number of levels for game

```
    deviceType = [UIDevice currentDevice].model;  
    if ([deviceType isEqualToString:@"iPad"])  
    { ipad = true; }  
    else { ipad = false; }
```

Might want to know if iPad

```
    fontstring = @"MarkerFelt-Wide";  
    fontcolor = [UIColor blackColor];
```

Adjust font family and color on demand

```
    CGRect screenBound = [[UIScreen mainScreen] bounds];  
    CGSize screenSize = screenBound.size;  
    screenWidth = screenSize.width;  
    screenHeight = screenSize.height;
```

Screen width & height auto-detected

```
    fontsize = 24;  
    barheight = 60;  
    buttonsiz = 60;
```

Specify font, menubar, button & card size

```
}
```

# Initialize Variables (cont.)

```
appTitle = @"Match the Candidates";
```

Specific Version

```
imageArray = [[NSMutableArray alloc] initWithObjects:@"Ben  
Carson",@"Bernie Sanders",@"Bobby Jindal",@"Carly Fiorina",@"Chris  
Christie",@"Donald Trump",@"George Pataki",@"Hillary Clinton",@"Jeb  
Bush",@"Jim Gilmore",@"Jim Webb",@"John Kasich",@"Lawrence  
Lessig",@"Lincoln Chafee",@"Lindsey Graham",@"Marco Rubio",@"Martin  
O'Malley",@"Mike Huckabee",@"Rand Paul",@"Rick Perry",@"Rick  
Santorum",@"Scott Walker",@"Ted Cruz",nil];
```

Image list



# Declare Variables

```
#import "ViewController.h"
```

```
@interface ViewController ()
```

```
@end
```

```
BOOL ipad;  
id deviceType;  
NSString * fontstring;  
NSString * appTitle;  
NSString * alertTitle;  
NSString * alertMessage;  
id fontcolor;
```

```
NSInteger baselevel, level, maxlevel, screenwidth, screenheight, maxcolumn,  
maxrow, barheight, fontsize, buttonsize, picWidth, picHeight, lastChoice,  
numMoves, gridsize;
```

```
NSMutableArray * imageArray;  
NSMutableArray * gridArray;  
NSMutableArray * revealedArray;
```

```
UIView * gridView;  
UIView * topMenuView;  
UIAlertView * alert;
```

```
UIButton * settingsButton;  
UIButton * helpButton;  
UILabel * topTitle;
```



# Add ResetGrid Routine

```
- (void)viewDidLoad {  
    [super viewDidLoad];  
    // Do any additional setup after loading the view, typically from a  
    nib.  
    [self initializeVariables];  
    [self resetGrid];  
}
```

```
-(void)resetGrid {  
    lastChoice = -1;  
    numMoves = 0;
```

Starting condition: no previous choice, zero moves

```
    gridArray = [[NSMutableArray alloc] init];  
    revealedArray = [[NSMutableArray alloc] init];
```

Create empty grid of  
cards, and revealed cards

```
    switch (level) {  
        case 1 : gridSize = 4; maxcolumn = 2; break;  
        case 2 : gridSize = 6; maxcolumn = 3; break;  
        case 3 : gridSize = 8; maxcolumn = 4; break;  
        case 4 : gridSize = 12; maxcolumn = 4; break;  
        case 5 : gridSize = 16; maxcolumn = 4; break;  
        case 6 : gridSize = 20; maxcolumn = 5; break;  
        case 7 : gridSize = 24; maxcolumn = 6; break;  
        case 8 : gridSize = 30; maxcolumn = 6; break;  
        case 9 : gridSize = 36; maxcolumn = 6; break;  
        case 10 : gridSize = 42; maxcolumn = 7; break;  
        case 11 : gridSize = 48; maxcolumn = 8; break;  
    }
```

Determine grid size  
based on level

...

```
int ctroffset = (arc4random() % imageArray.count);
```

```
for (NSInteger ctr = 0; ctr < gridsize; ctr++) {  
    [revealedArray setObject:@"0" atIndexSubscript:ctr];  
    [gridArray setObject:@"" atIndexSubscript:ctr];  
}
```

```
for (NSInteger ctr = 0; ctr < gridsize/2; ctr++) {
```

```
    NSInteger fixedctr = ctr + ctroffset;  
    if (fixedctr >= imageArray.count) {  
        fixedctr = ctr + ctroffset - imageArray.count;  
    }
```

```
NSString * object = [imageArray objectAtIndex:fixedctr];
```

```
int pos1 = (arc4random() % gridsize);
```

```
BOOL notFree = true;
```

```
while (notFree) {
```

```
    if ([[gridArray objectAtIndex:pos1] isEqualToString:@""]) {  
        notFree = false;  
        [gridArray setObject:object atIndexSubscript:pos1];  
    }
```

```
    else {  
        pos1++;  
        if (pos1 >= gridsize) {pos1 = 0;}  
    }
```

```
}
```

```
int pos2 = (arc4random() % gridsize);
```

```
notFree = true;
```

```
while (notFree) {
```

```
    if ([[gridArray objectAtIndex:pos2] isEqualToString:@""]) {  
        notFree = false;  
        [gridArray setObject:object atIndexSubscript:pos2];  
    }
```

```
    else {  
        pos2++;  
        if (pos2 >= gridsize) {pos2 = 0;}  
    }
```

```
}
```

```
}
```

```
}
```

Randomize starting point in card array

Clear revealed and grid arrays

Cycle through images, starting at randomized offset, filling two spots in grid array, at random, if position empty.



# Add ShowMenus Routine

```
- (void)viewDidLoad {  
    [super viewDidLoad];  
    // Do any additional setup after loading the view, typically from a  
    nib.  
    [self initializeVariables];  
    [self resetGrid];  
    [self showMenus];  
}
```



Match the Candidates: CHRIS  
CHRISTIE

```

-(void)showMenus {
    [topMenuView removeFromSuperview];
    topMenuView = [[UIView alloc] initWithFrame:CGRectMake(0,
0, screenwidth, barheight)];
    topMenuView.backgroundColor = [UIColor blackColor];

    topTitle = [[UILabel alloc]
initWithFrame:CGRectMake(barheight, 0, screenwidth-2*barheight,
barheight)];
    topTitle.textAlignment = NSTextAlignmentCenter;
    topTitle.textColor = [UIColor whiteColor];
    topTitle.font = [UIFont fontWithName:fontstring
size:fontsize];
    topTitle.numberOfLines = 0;
    topTitle.adjustsFontSizeToFitWidth = YES;
    topTitle.userInteractionEnabled = NO;
    topTitle.text = appTitle;
    [topMenuView addSubview:topTitle];

    helpButton = [UIButton buttonWithType:UIButtonTypeCustom];
    [helpButton setFrame:CGRectMake(0, 0, buttonsize,
buttonsize)];
    [helpButton setImage:[UIImage imageNamed:@"Help.png"]
forState:UIControlStateNormal];
    [helpButton addTarget:self action:@selector(showHelp)
forControlEvents:UIControlEventTouchUpInside];
    helpButton.accessibilityIdentifier = @"Help";

    [topMenuView addSubview:helpButton];
    [self.view addSubview:topMenuView];
}

```

Remove if already exists

Initialize with size, color

Add text title

Add Help button



Match the Candidates: CHRIS  
CHRISTIE

# Add ShowGrid Routine

```
- (void)viewDidLoad {  
    [super viewDidLoad];  
    // Do any additional setup after loading the view, typically from a  
    nib.  
    [self initializeVariables];  
    [self resetGrid];  
    [self showMenus];  
    [self showGrid];  
}
```



```

-(void)showGrid {
    [gridView removeFromSuperview];
    gridView = [[UIView alloc] initWithFrame:CGRectMake(0, barheight, screenwidth, screenheight-barheight)];
    gridView.clipsToBounds = YES;
    gridView.backgroundColor = [UIColor colorWithPatternImage:[UIImage imageNamed:@"greenfeltsmall.jpg"]];

    maxrow = gridsize/maxcolumn;
    picWidth = screenwidth / (maxcolumn + 1);
    picHeight = picWidth * 4/3;

    float xspacing = (screenwidth - maxcolumn*picWidth)/maxcolumn;
    float ymargin = xspacing;

    NSInteger column, row, i, offset;
    column = 0;
    row = 0;
    i = 0;
    NSString *item;

    offset = (screenwidth - (maxcolumn*buttonsize))/(1+maxcolumn);
    UIImage * bgImage = [UIImage imageNamed:@"PlayingCardBack.png"];

    while (i < gridArray.count) {
        item = [gridArray objectAtIndex:i];

        UIImage * thisImage = [UIImage imageNamed:[NSString stringWithFormat:@"%d.png", [gridArray objectAtIndex:i]]];
        UIButton *button = [UIButton buttonWithType:UIButtonTypeCustom];
        button.backgroundColor = [UIColor clearColor];
        button.frame = CGRectMake(xspacing/2+column*(picWidth+xspacing), row*(picHeight+ymargin)+ymargin, picWidth,
picHeight);
        [[button imageView] setContentMode:UIViewContentModeScaleAspectFit];
        if ([[revealedArray objectAtIndex:i] isEqualToString:@"1"]) {
            [button setImage:[thisImage stretchableImageWithLeftCapWidth:0.0 topCapHeight:0.0]
forState:UIControlStateNormal];
        }
        else {[button setBackgroundImage:[bgImage stretchableImageWithLeftCapWidth:0.0 topCapHeight:0.0]
forState:UIControlStateNormal];}

        button.accessibilityLabel = [NSString stringWithFormat:@"%ld", (long)i];
        [button addTarget:self action:@selector(choseButton:) forControlEvents:UIControlEventTouchUpInside];
        button.tag = i;
        [gridView addSubview:button];
        column++;
        if (column >= maxcolumn) {column=0;row++;}
        i++;
    }
    [self.view addSubview:gridView];
}

```

remove gridview if already exists

Auto-adjusts card size based on screen size and number of cards

Display playing card or image, if already revealed

Launch choseButton if selected

```

-(void)choseButton:(id) sender {
    UIButton * thisbutton = (UIButton *)sender;
    NSInteger choice = thisbutton.tag;
    NSString * chosenImage = [NSString stringWithFormat:@"%d.png", [gridArray objectAtIndex:choice]];

    topTitle.text = [NSString stringWithFormat:@"%d: %d", appTitle, [[gridArray objectAtIndex:choice]
uppercaseString]];

    if (![revealedArray objectAtIndex:choice] isEqualToString:@"1"]) {
        numMoves++;
        [revealedArray setObject:@"1" atIndex:choice];
        [self showGrid];
        if (lastChoice == -1) { lastChoice = choice;}
        else {
            if (!(lastChoice == choice) && ([gridArray objectAtIndex:lastChoice] isEqualToString:
[gridArray objectAtIndex:choice])) {
                lastChoice = -1;
                BOOL done = true;
                for (int tmpctr = 0; tmpctr < revealedArray.count; tmpctr++) {
                    if ([revealedArray objectAtIndex:tmpctr] isEqualToString:@"0"]) {done = false;}
                }
                if (done) {
                    if (level < maxlevel) {level++;}
                    [self showDone];
                }
                topTitle.text = appTitle;
            }
            else { // reset revealed
                [self showGrid];
                [revealedArray setObject:@"0" atIndex:choice];
                [revealedArray setObject:@"0" atIndex:lastChoice];
            }
            lastChoice = -1;
        }
    }
}
}

```

Detect chosen card

Display chosen card name

Chose blank card

Set choice as first choice

Match made! Now check if all cards matched

If all matched, level up unless max level


hide last two choices since not matches

# Add Graphics

Choose options for adding these files:

Destination: ☒ Copy items if needed

Added folders: ☒ Create groups  
☐ Create folder references

Add to targets: ☒  ConcentrationTutorial

Cancel Finish



# Optional Enhancements

- Usability
- Help
- Set Levels
- Cosmetics
- Custom versions

# Briefly Show Image

```
-(void)choseButton:(id) sender {
```

```
...
```

```
    UIImageView * animatedOverlay = [[UIImageView alloc]
initWithFrame:CGRectMake(0, barheight, screenwidth, screenheight-
barheight)];
```

```
    NSMutableArray * imgArray = [[NSMutableArray alloc] init];
    [imgArray addObject:[UIImage imageNamed:chosenImage]];
    animatedOverlay.animationImages = imgArray;
    animatedOverlay.animationDuration = 0.5; // seconds
    animatedOverlay.animationRepeatCount = 1; // 0 = loops forever
    [animatedOverlay startAnimating];
    [[self view] addSubview:animatedOverlay];
}
```

# Add ShowAll Routine

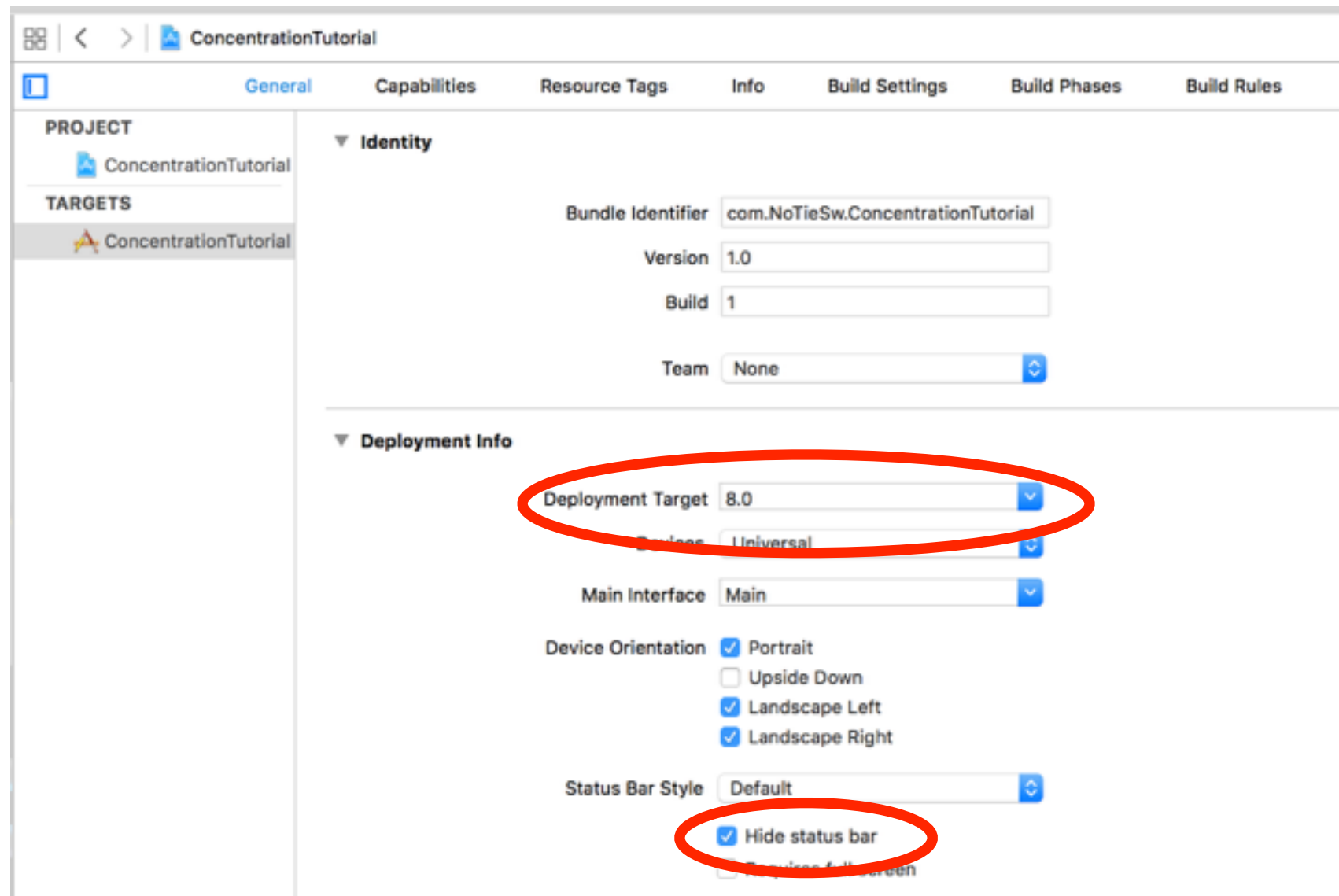
```
- (void)viewDidLoad {  
    [super viewDidLoad];  
    // Do any additional setup after loading the view, typically from a  
    nib.  
    [self initializeVariables];  
    [self resetGrid];  
    [self showMenus];  
    [self showGrid];  
    [self showAll];  
}
```



# ShowAll

```
-(void)showAll {  
    float duration = imageArray.count/5;  
  
    UIImageView * animatedOverlay = [[UIImageView alloc]  
initWithFrame:CGRectMake(0, barheight, screenwidth, screenheight-  
barheight)];  
  
    NSMutableArray * imgArray = [[NSMutableArray alloc] init];  
  
    for (int ctr = 0; ctr < imageArray.count; ctr++) {  
        NSString * chosenImage = [NSString stringWithFormat:@"%d.png",  
[imageArray objectAtIndex:ctr]];  
        [imgArray addObject:[UIImage imageNamed:chosenImage]];  
    }  
  
    animatedOverlay.animationImages = imgArray;  
    animatedOverlay.animationDuration = duration; // seconds  
    animatedOverlay.animationRepeatCount = 1; // 0 = loops forever  
    [[self view] addSubview:animatedOverlay];  
    [animatedOverlay startAnimating];  
}
```

# Hide Status Bar, Support Earlier Versions of iOS



# Select Levels

```
-(void)showDone {
    topTitle.text = [NSString stringWithFormat:@"%d Moves",appTitle,(long)numMoves];

    alertTitle = [NSString stringWithFormat:@"SOLVED in %ld MOVES!",(long)numMoves];
    if (level == maxlevel) {
        alertMessage = @"Well done! Try again!";
    }
    else {
        alertMessage = [NSString stringWithFormat:@"Well done! Now try again, at level %ld", (long)level];
    }
    [self showAlert];
    [self resetGrid];
    [self showGrid];
}

-(void)showHelp {
    alertTitle = [NSString stringWithFormat:@"Welcome to %@!",appTitle];
    alertMessage = @"Select a Level";
    alert = [[UIAlertView alloc] initWithTitle:alertTitle message:alertMessage delegate:nil cancelButtonTitle:@"Continue"
otherButtonTitles:@"1",@"2",@"3",@"4",@"5",@"6",@"7",@"8",@"9",@"10",nil];
    alert.delegate = self;
    [alert show];
}

-(void)showAlert {
    alert = [[UIAlertView alloc] initWithTitle:alertTitle message:alertMessage delegate:nil cancelButtonTitle:@"Continue" otherButtonTitles:nil];
    alert.delegate = self;
    [alert show];
}

- (void)alertView:(UIAlertView *)alertView didDismissWithButtonIndex:(NSInteger)buttonIndex {
    NSString *buttonTitle = [[alertView buttonTitleAtIndex:buttonIndex] lowercaseString];
    // if ([buttonTitle isEqualToString:@"settings"]) {[self showSettings];}

    if ([buttonTitle integerValue] > 0) {
        level = [buttonTitle integerValue];
        [self resetGrid];
        [self showGrid];
    }
    if ([buttonTitle isEqualToString:@"continue"]) {}
    if ([buttonTitle isEqualToString:@"exit"]) {[self exit];}
}

-(void)exit {exit(1);}

- (void)alertView:(UIAlertView *)alertView clickedButtonAtIndex:(NSInteger)buttonIndex { // [alertView removeFromSuperview];
    NSString *buttonTitle = [[alertView buttonTitleAtIndex:buttonIndex] lowercaseString];
    // if ([buttonTitle isEqualToString:@"settings"]) {[self showSettings];}
    if ([buttonTitle integerValue] > 0) {
        level = [buttonTitle integerValue];
        [self resetGrid];
        [self showGrid];
    }
    if ([buttonTitle isEqualToString:@"continue"]) {}
    if ([buttonTitle isEqualToString:@"exit"]) {[self exit];}
}
```

# Show Alerts

```
-(void)showDone {
    topTitle.text = [NSString stringWithFormat:@"%@@: %ld Moves",appTitle,(long)numMoves];
    alertTitle = [NSString stringWithFormat:@"SOLVED in %ld MOVES!",(long)numMoves];
    if (level == maxlevel) { alertMessage = @"Well done! Try again!";}
    else { alertMessage = [NSString stringWithFormat:@"Well done! Now try again, at level %ld",(long)level]; }
    [self showAlert];
    [self resetGrid];
    [self showGrid];
}

-(void)showAlert {
    alert = [[UIAlertView alloc] initWithTitle:alertTitle message:alertMessage delegate:nil
cancelButtonTitle:@"Continue" otherButtonTitles:nil];
    alert.delegate = self;
    [alert show];
}

-(void)showHelp {
    alertTitle = [NSString stringWithFormat:@"Welcome to %@!",appTitle];
    alertMessage = @"Select a Level";
    alert = [[UIAlertView alloc] initWithTitle:alertTitle message:alertMessage delegate:nil
cancelButtonTitle:@"Continue" otherButtonTitles:@"1",@"2",@"3",@"4",@"5",@"6",@"7",@"8",@"9",@"10",nil];
    alert.delegate = self;
    [alert show];
}

- (void)alertView:(UIAlertView *)alertView didDismissWithButtonIndex:(NSInteger)buttonIndex {
    NSString *buttonTitle = [[alertView buttonTitleAtIndex:buttonIndex] lowercaseString];

    if ([buttonTitle integerValue] > 0) {
        level = [buttonTitle integerValue];
        [self resetGrid];
        [self showGrid];
    }
    if ([buttonTitle isEqualToString:@"continue"]) {}
    if ([buttonTitle isEqualToString:@"exit"]) {[self exit];}
}
```



# Select Levels

```
-(void)setLevel:(id) sender {  
    UIButton *thisbutton = (UIButton *)sender;  
    level = thisbutton.tag;  
    [self resetGrid];  
    [self showGrid];  
}
```

# Automatically Create Multiple Versions

```
NSString * bundleID = [[[NSBundle mainBundle] bundleIdentifier]  
lowercaseString];
```

Auto detects app ID at launch...  
com.NoTieSw.Candidates or  
com.NoTieSw.Presidents, etc.

```

NSString * bundleID = [[[NSBundle mainBundle] bundleIdentifier] lowercaseString];

if ([bundleID isEqualToString:@"com.notiesw.matchpresidents"]) {
    appTitle = @"Match the Presidents";
    imageArray = [[NSMutableArray alloc]
initWithObjects:@"AJohnson", @"Arthur", @"BenHarrison", @"Buchanan", @"Carter", @"Cleveland",
@"Clinton", @"Coolidge", @"Eisenhower", @"FDRoosevelt", @"Fillmore", @"Ford", @"GHWBush", @"G
WBush", @"Garfield", @"Grant", @"Harding", @"Hayes", @"Hoover", @"JQAdams", @"Jackson", @"Jeffe
rson", @"JohnAdams", @"Kennedy", @"LBJohnson", @"Lincoln", @"Madison", @"McKinley", @"Monroe",
@"Nixon", @"Obama", @"Pierce", @"Polk", @"Reagan", @"TRoosevelt", @"Taft", @"Taylor", @"Truman",
@"Tyler", @"VanBuren", @"WHHarrison", @"Washington", @"Wilson", nil];
}
if ([bundleID isEqualToString:@"com.notiesw.pokematch"]) {
    appTitle = @"PokeMatch";
    imageArray = [[NSMutableArray alloc]
initWithObjects:@"abra", @"aerodactyl", @"alakazam", @"arbok", @"arcanine", @"articuno", @"be
edrill", @"bellsprout", @"blastoise", @"bulbasaur", @"butterfree", @"caterpie", @"chansey", @"
charizard", @"charmander", @"charmeleon", @"clefable", @"clefairy", @"cloyster", @"cubone", @"
dewgong", @"diglett", @"ditto", @"dodrio", @"doduo", @"dragonair", @"dragonite", @"dratiniagan
ime", @"drowzee", @"dugtrio", @"eevee", @"ekans", @"electabuzz", @"electrode", @"exeggcute", @"
exeggutor", @"farfetchd", @"fearow", @"flareon", @"gastly", @"gengar", @"geodude", @"gloom", @"
golbat", @"goldeen", @"golduck", @"golem", @"graveler", @"grimer", @"growlithe", @"gyarados", @
"haunter", @"hitmonchan", @"hitmonlee", @"horsea", @"hypno", @"ivysaur", @"jigglypuff", @"jolt
eon", @"jynx", @"kabuto", @"kabutops", @"kadabra", @"kakuna", @"kangaskhan", @"kingler", @"koff
ing", @"krabby", @"lapras", @"lickitung", @"machop", @"machop", @"magikarp", @"magmar", @"magn
emite", @"magneton", @"mankey", @"marowak", @"metapod", @"mewtwo", @"moltres", @"mr.mime", @"mu
k", @"nidoking", @"nidoqueen", @"nidoran", @"nidorina", @"nidorino", @"ninetales", @"oddish", @
"omanyte", @"omastar", @"onix", @"paras", @"parasect", @"persian", @"pidgeot", @"pidgeotto", @
"pidgey", @"pikachu", @"pinsir", @"poliwag", @"poliwhirl", @"poliwrath", @"ponyta", @"porygon",
@"primeape", @"psyduck", @"raichu", @"rapidash", @"raticate", @"rattata", @"rhydon", @"rhyhorn",
@"sandshrew", @"sandslash", @"scyther", @"seaking", @"seel", @"shellder", @"slowbro", @"slow
poke", @"snorlax", @"spearow", @"squirtle", @"starmie", @"staryu", @"tangela", @"tauros", @"ten
tacool", @"tentacruel", @"vaporeon", @"venomoth", @"venonat", @"venusaur", @"victreebel", @"vi
leplume", @"voltorb", @"vulpix", @"wartortle", @"weedle", @"weepinbell", @"weezing", @"wigglyt
uff", @"zapdos", @"zubat", nil];
}

```