Course Project: **WhichBill**

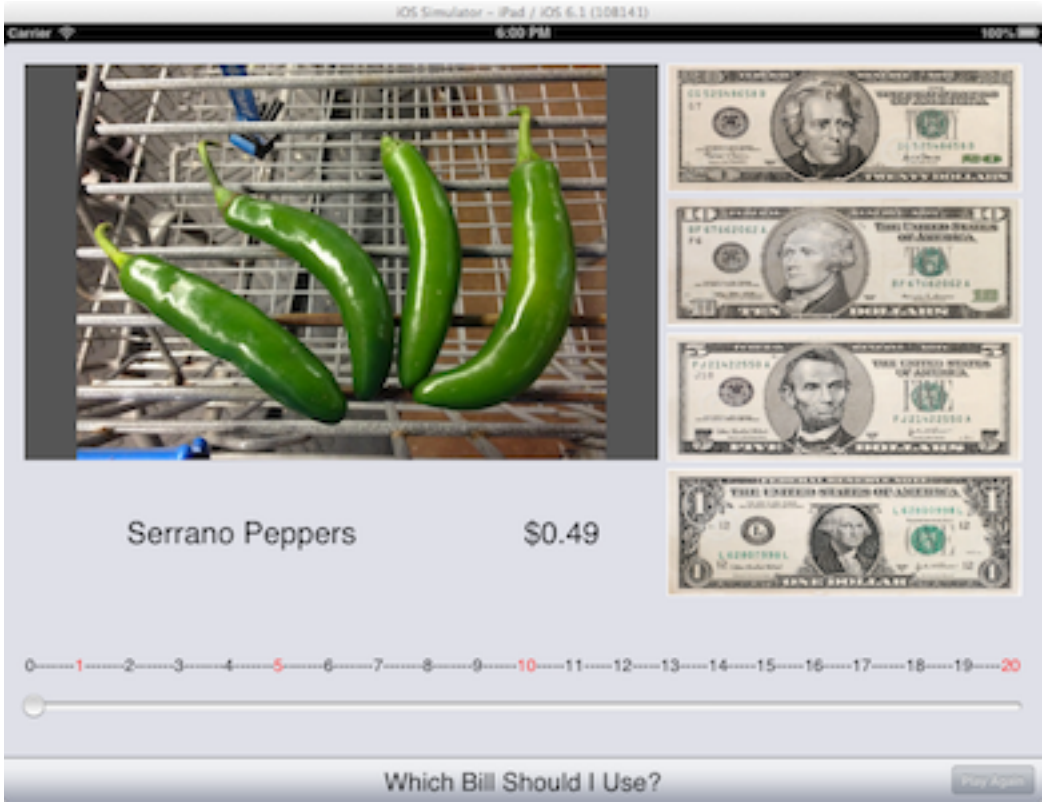
Amar Shah

CECS 590: iOS Development

July 24, 2013

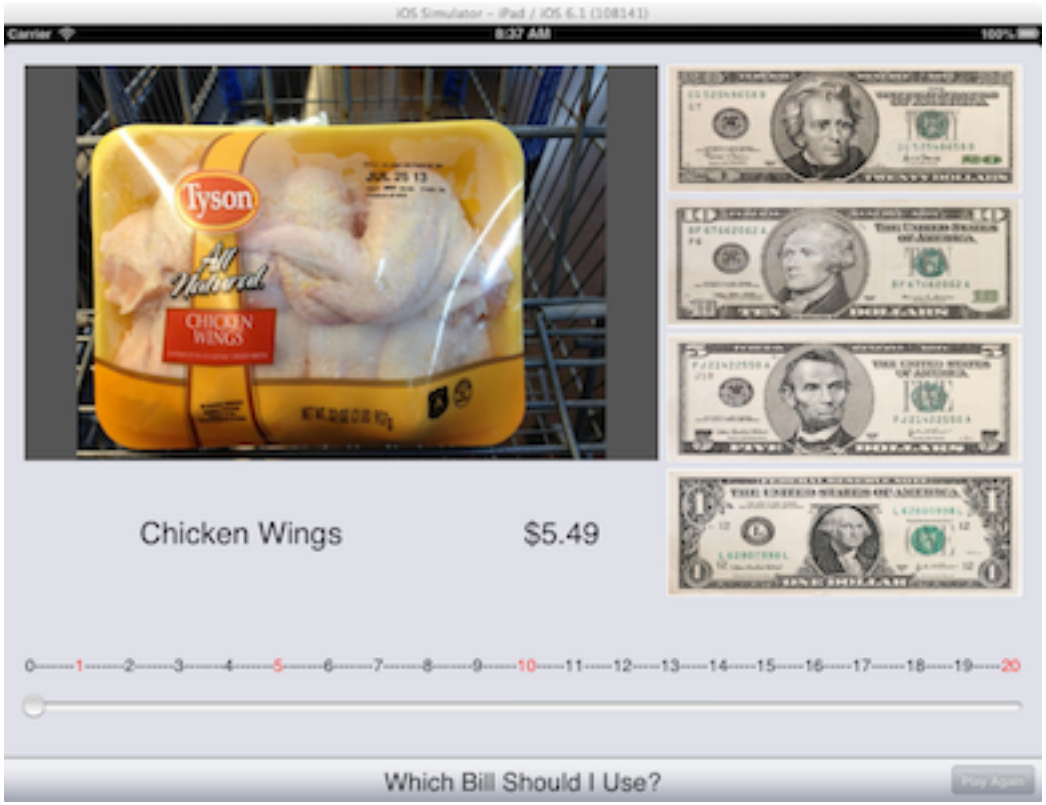
WhichBill

WhichBill is an educational game for students with developmental disabilities.



With **WhichBill,** students with disabilities practice using paper currency for everyday purchases.

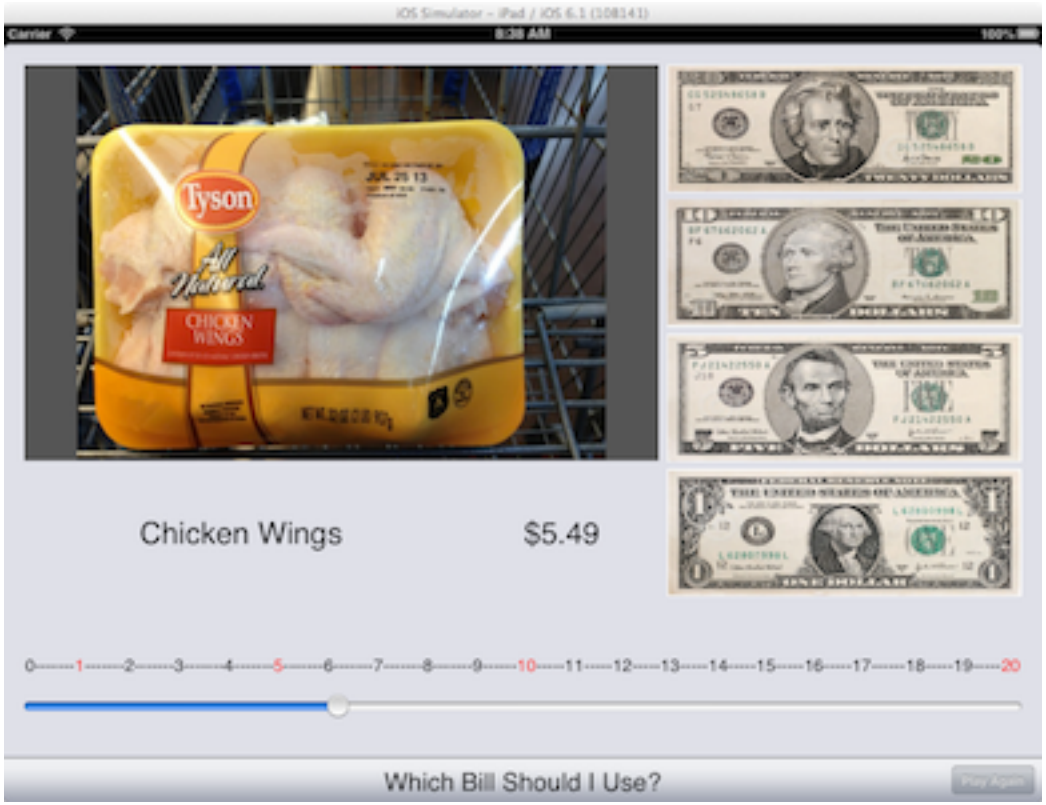
WhichBill presents the player with an image of a grocery item along with its name and cost.



On the right there are four answer buttons, each displaying a different denomination of currency.

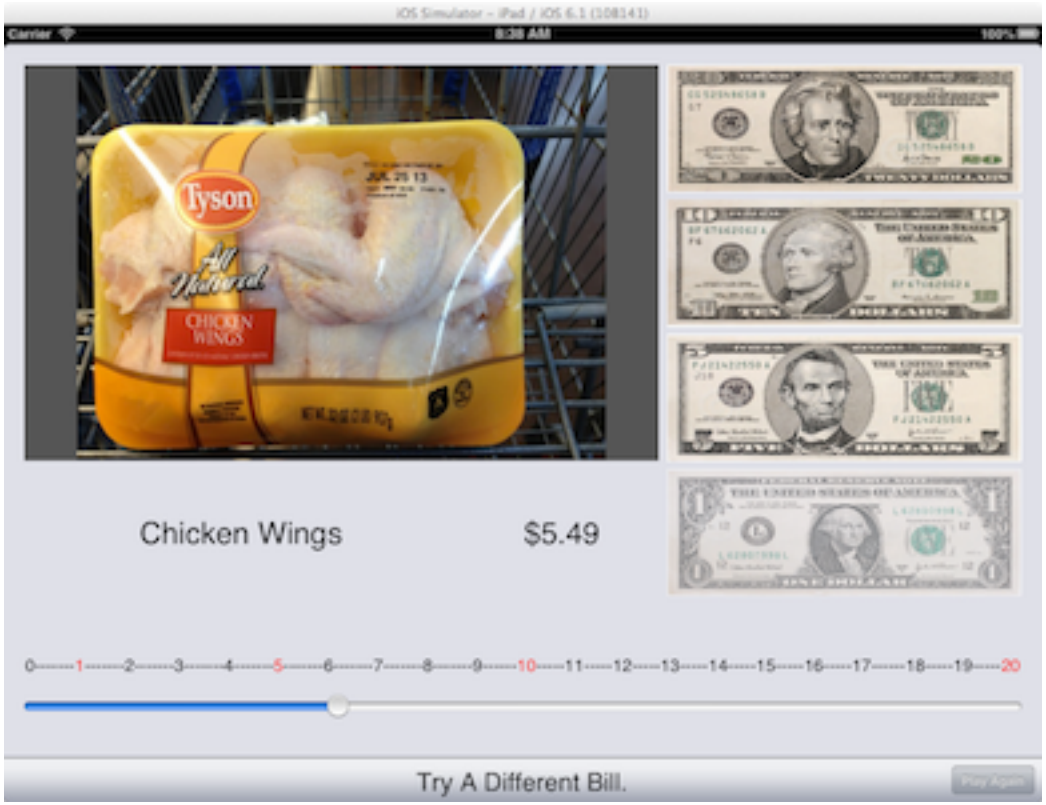
At the bottom there is a number line. The player can move the slider along the line freely.

Using the number line, the player can compare the item's cost with the value of the different bills.



WhichBill's number line is an important educational feature. According to the special education teacher who created the original **WhichBill** board game: "Without the number line, the game is a testing tool, not a teaching tool."

When the player chooses a bill whose value is **too small** or **unnecessarily large**, the button representing that bill is disabled.

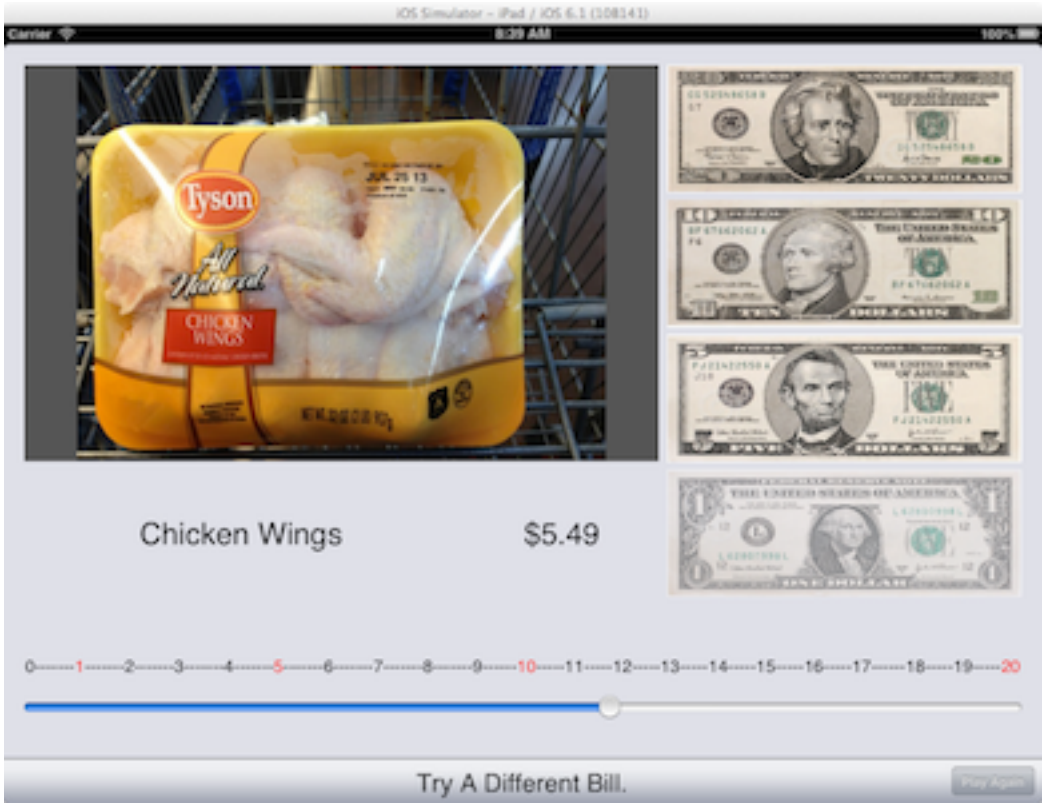


WhichBill asks the player to try a different bill.

As the player slides on the number line past the value of the correct bill, all of the incorrect answers are temporarily disabled.

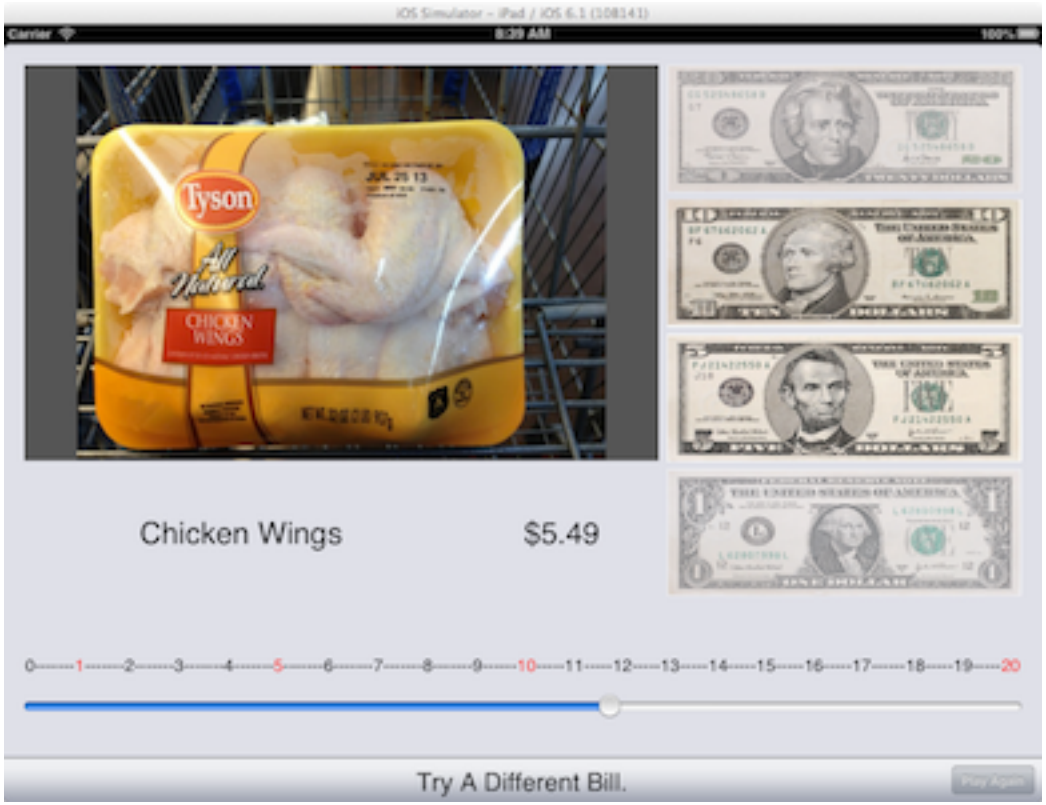
This helps the player find the correct bill even if he has no idea where to start.

As the player slides away from the correct bill's value, incorrect answers become enabled again.



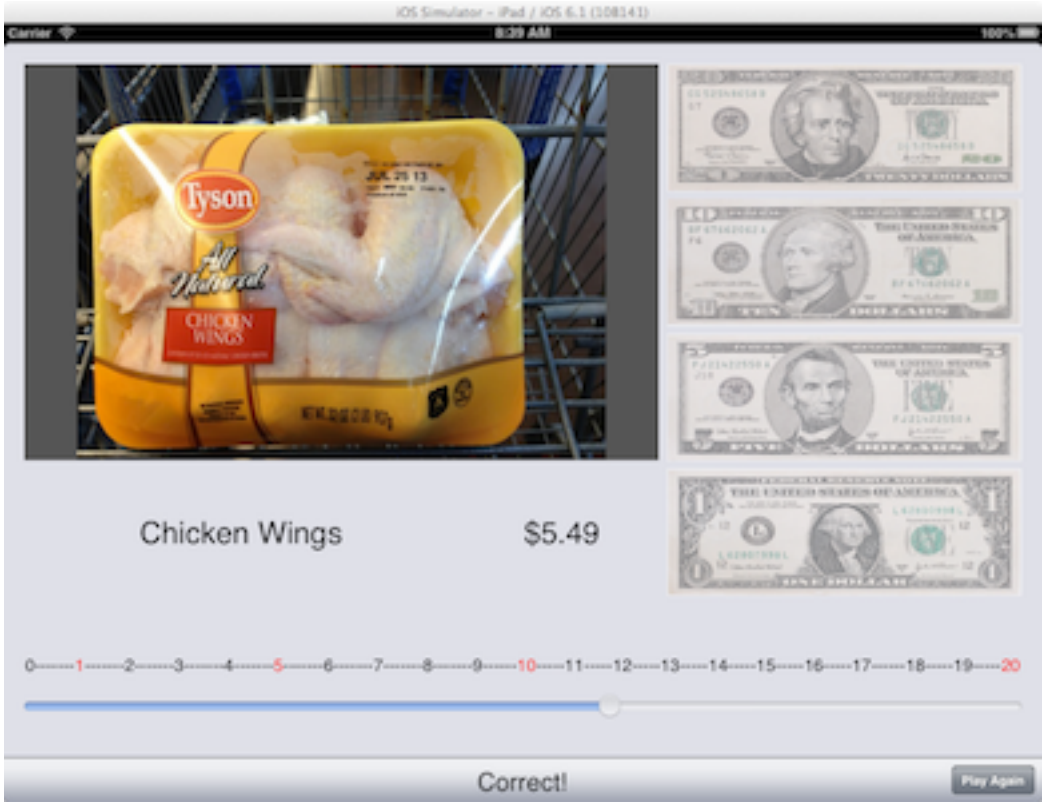
But incorrect answers that have already been guessed remain disabled.

Each incorrect guess is successively disabled.



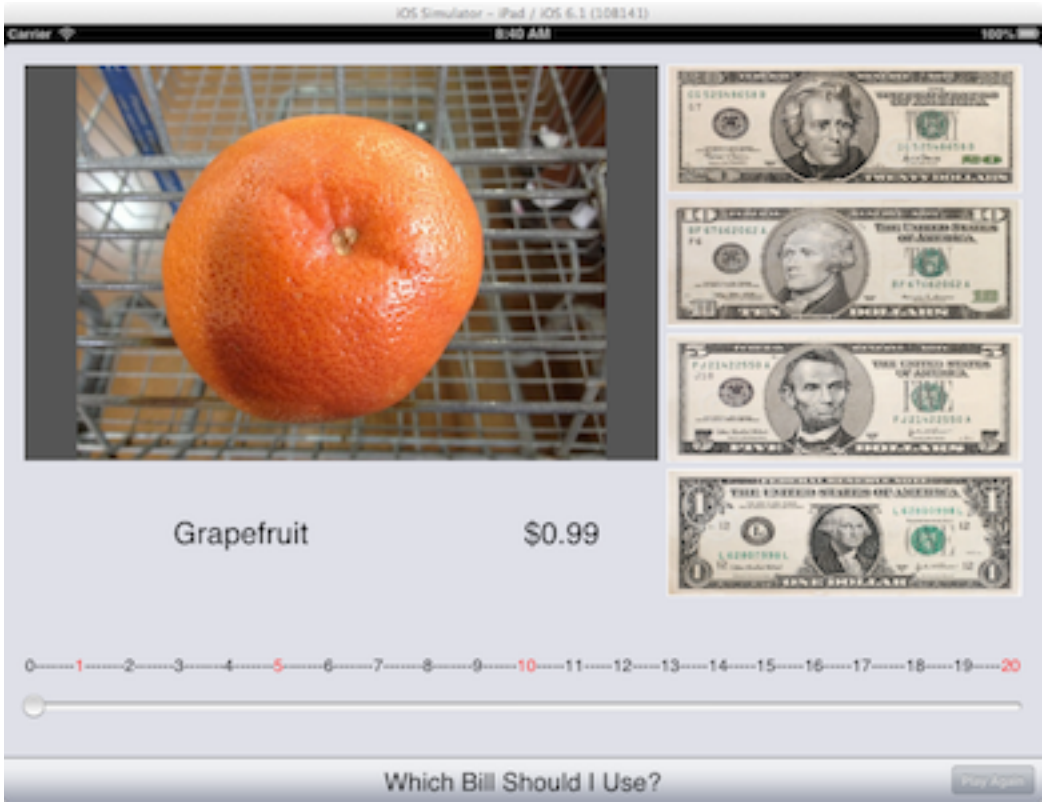
The player can guess incorrectly until only the correct answer remains an option.

WhichBill congratulates the player when she chooses the correct bill. All answer buttons are disabled.



Now the player can choose to **Play Again**.

The next item will always be different from the last.



The game starts again.

Implementation

WhichBill is a single-view application for the iPad only. It is used in landscape orientation.

WhichBill uses a **Model-View-Controller(-Store)** design pattern, as demonstrated in Conway and Hillegass, *iOS Programming:The Big Nerd Ranch Guide, 3rd Edition*.

A **WBItem** represents a single grocery item. It maintains its **name** and **cost** properties, as well as an **imageKey** that refers to the item's image in a separate image store.

```
#import <Foundation/Foundation.h>

@interface WBItem : NSObject
{
    double cost;
}

@property (nonatomic, strong) NSString *name;
@property (nonatomic, copy) NSString *imageKey;

- (id)initWithName:(NSString *)n cost:(double)c;
- (double)cost;
- (void)setCost:(double)c;

@end
```

A **WBItem**'s properties are set during initialization and remain unchanged during the execution of **WhichBill**.

WBItems are managed in a **WBItemStore**.

WBItemStore is a singleton, meaning that only one instance of the class is allowed.

This instance is retrieved with the message **sharedStore**.

```
#import <Foundation/Foundation.h>

@class WBItem;

@interface WBItemStore : NSObject
{
    NSArray *allItems;
}

+ (WBItemStore *)sharedStore;
- (NSArray *)allItems;

@end
```

The current release of **WhichBill** uses a **WBItemStore** containing only ten **WBItem**s with hard-coded **cost** values. Future releases of **WhichBill** will feature more **WBItem**s, with randomly generated **cost**s within defined ranges.

WBItems are created and added to a **WBItemStore** during its initialization. At that time, **UIImage**s corresponding to each **WBItem** in the **WBItemStore** are created and stored in a **WBImageStore**, according to the **imageKey** for each **WBItem.**

WBImageStore is also a singleton.

WBImageStore is similar to, but simpler than, the **BNRImageStore** presented by Conway and Hillegass.

```
#import <Foundation/Foundation.h>

@interface WBImageStore : NSObject
{
    NSMutableDictionary *dictionary;
}

+ (WBImageStore *)sharedStore;
- (void)setImage:(UIImage *)i forKey:(NSString *)s;
- (UIImage *)imageForKey:(NSString *)s;

@end
```

WBItem, **WBItemStore**, and **WBImageStore** together comprise **WhichBill**'s model and store.

An important unit of **WhichBill**'s view is **WBAnswerButton**.

WBAnswerButton is a custom sub-class of **UIButton**.

```
#import <UIKit/UIKit.h>

@interface WBAnswerButton : UIButton

@property BOOL alreadyGuessedWrong;

@end
```

WBAnswerButton differs from **UIButton** in terms of one property, the boolean **alreadyGuessedWrong**.

alreadyGuessedWrong keeps track of whether this button was already chosen by the player as an incorrect answer for the current **WBItem**. Aside from dynamic content, this is the only feature of a view or subview that **WhichBill** implements programmatically.

The appearance of **WBAnswerButton**s and the arrangement of **WhichBill**'s view are established graphically with **WhichBillViewController.xib** and appropriately configured to outlets and actions in **WhichBillViewController**.

WhichBillViewController handles the logic of game-play.

It manages outlets to all dynamic views and implements actions to handle game events.

WhichBillViewController also keeps weak references to the current **WBItem** and the correct **WBAnswerButton**.

```
#import <UIKit/UIKit.h>
#import "WBItem.h"
#import "WBAnswerButton.h"

@interface WhichBillViewController : UIViewController
{
    __weak IBOutlet UIImageView *imageView;
    __weak IBOutlet UILabel *nameLabel;
    __weak IBOutlet UILabel *costLabel;
    __weak IBOutlet UILabel *msgLabel;
    __weak IBOutlet WBAnswerButton *twentyDollarButton;
    __weak IBOutlet WBAnswerButton *tenDollarButton;
    __weak IBOutlet WBAnswerButton *fiveDollarButton;
    __weak IBOutlet WBAnswerButton *oneDollarButton;
    __weak IBOutlet UIBarButtonItem *playAgainButton;
    __weak IBOutlet UISlider *slider;
}

@property (nonatomic, weak) WBItem *currentItem;
@property (nonatomic, weak) WBAnswerButton *correctButton;

- (IBAction)buttonPushed:(id)sender;
- (IBAction)playAgain:(id)sender;
- (void)answerCorrect;
- (void)answerIncorrect:(WBAnswerButton *)answer;
- (void)sliderChanged:(id)sender;

@end
```

WhichBillViewController manages **WhichBill**'s single main view. It is the application's only controller.

Release

WhichBill is completely functional in its current release, and it has been tested on a provisioned iPad.

Although the current release is highly limited by audience, **WhichBill** could even now be used by students and teachers as a fun and engaging educational tool. **WhichBill** will be offered as a free download in the iOS App Store as soon as possible and prudent.

Future releases of **WhichBill** will implement successively more complex and potentially very useful features, such as greater variety in **WBItem**s and a larger **WBItemStore**, management options to tailor the game to the player's ability, and persistent data storage and visualization of long-term student improvement. It is envisioned that **WhichBill** will lead a suite of interactive iOS applications designed to assist special education teachers to meet the needs and track the progress of their students.