# Perfecting Math Skills

Course Project for CSCI 352

Developed by Charles Clayton and Robert Rayburn

### **Overview**

- Background
- Approach
- Class Diagram
- Demo
- Future Work
- Questions

# **Background**

- Create a tool for children to practice basic arithmetic as a game
  - Gives a boost in school
  - Teachers can move into more advanced concepts earlier
- Developed as a Smart Board Application

## **Approach**

- Provide simple interface to allow user to choose between different modes
- Uses basic WPF Application Directives
  - Windows
  - Windows.System
  - Windows.Controls
- Patterns Used:
  - Facade pattern to simplify user interface
  - Factory pattern for creating exitwindows and counters

# Gameplay

- Modes:
  - Addition
    - Generates problems consisting of two random integers less than 26
  - Subtraction
    - Generates problems consisting of two random integers less than 26
    - Checks that solution is nonnegative
  - Multiplication
    - Generates problems consisting of two random integers less than 13
      - Because common multiplication tables are 12x12

# **Class Diagram**

```
main

+first: Window
+add: Button
+subtract: Button
+multiply: Button
+grid: Grid
+add_click(sender:object,e:RoutedEventArgs): void
+subtract_click(sender:object,e:RoutedEventArgs): void
+multiply_click(sender:object,e:RoutedEventArgs): void
```

```
ExitWindow

-grid2: Grid
-exit: Window
-goodjob: TextBox
-right: TextBlock
-attempted: TextBlock
+ExitWindow(correct:int,attempts:int)
-Exit Closed(sender:object,e:EventArgs): void
```

```
-right: int
-attempts: int
+updateBoth(): void
+updateAttempt(): void
+Counter()
```

Counter

#### AddWindow

```
buttonCounter: int = 0
-solution: int
-userAnswer: int
 addition: Window
 addWindowGrid: Grid
-TopNum: TextBlock
 BottomNum: TextBlock
-Symbol: TextBlock
 AnswerBox: TextBox
-Enter: Button
-zero: Button
+backspace: Button
 correct: Window
-incorrect: Window
- randomNum: Random
 top: int
 -bot: int
+Check Window(): void
Window KeyDown(sender:object,e:KeyEventArgs,
                ): void
+Button Click(sender:object,e:RoutedEventArgs): void
 Back Click(sender:object,e:RoutedEventArgs): void
 -Next Click(sender:object,e:RoutedEventArgs): void
+getNum(max:int): int
-one Click(sender:object,e:RoutedEventArgs): void
-two Click(sender:object,e:RoutedEventArgs): void
-three Click(sender:object,e:RoutedEventArgs): void
-four Click(sender:object,e:RoutedEventArgs): void
-five Click(sender:object,e:RoutedEventArgs): void
 -six Click(sender:object,e:RoutedEventArgs): void
 -seven Click(sender:object,e:RoutedEventArgs): void
 eight Click(sender:object,e:RoutedEventArgs): void
 -nine Click(sender:object,e:RoutedEventArgs): void
-zero Click(sender:object,e:RoutedEventArgs): void
 -backspace Click(sender:object,e:RoutedEventArgs): void
+update(): void
+AddWindow()
```

#### Demo

To set up our application you can either use the executable file, or open and run the application in visual studio.

#### **Future Work**

- Create different difficulties
  - Allow users to challenge themselves at higher levels
- Adding more modes
  - Division
  - Mixed
- Creating a timed version
  - Challenge users to solve as many problems as possible

#### Questions

Authors:

Robert Rayburn - robcrayb@ut.utm.edu

Charles Clayton - <a href="mailto:chaeclay@ut.utm.edu">chaeclay@ut.utm.edu</a>

Repo:

https://github.com/SoftwareEngineeringProjectCharlesRobert/PerfectingMathSkills