Final Project Assignment

Due date: _____

Objectives

The purpose of this assignment is to bring together everything you've learned this semester. Use the concepts learned. For example: methods, looping, branching, collections, naming, etc.

The assignment is to create a program of your choosing. I have supplied a couple of example program descriptions below. You can choose to just write one of those, create a variation of one, or something completely different that you have in mind.

Regardless of what program you want to create, you will need to write a project proposal that completely describes the program you want to write and details your plan to complete it. This proposal will be the first 50 points of the project. Mr. Brooks will evaluate each proposal and may ask for changes to the proposal if the project is too hard or too easy (or if there isn't enough detail in the proposal—there should be enough detail that someone else could write the program based solely on your description). Include how you will use each of the skills (described below) in your project. Once your proposal is approved you may begin writing your program. (Note that the descriptions given below for the example programs aren't nearly descriptive enough. You will need to fill in your own details, for example scoring.)

This assignment is worth 300 points. That's three times as much as the other programming assignments. Therefore, you can expect this project to take three times as much time to complete as the other programming assignments, so plan accordingly.

For the program you choose to write there may be additional things required that you haven't specifically learned about in class. If you're unsure how to do something, you can email Mr. Brooks or ask the question in class (because most likely others will have the same question).

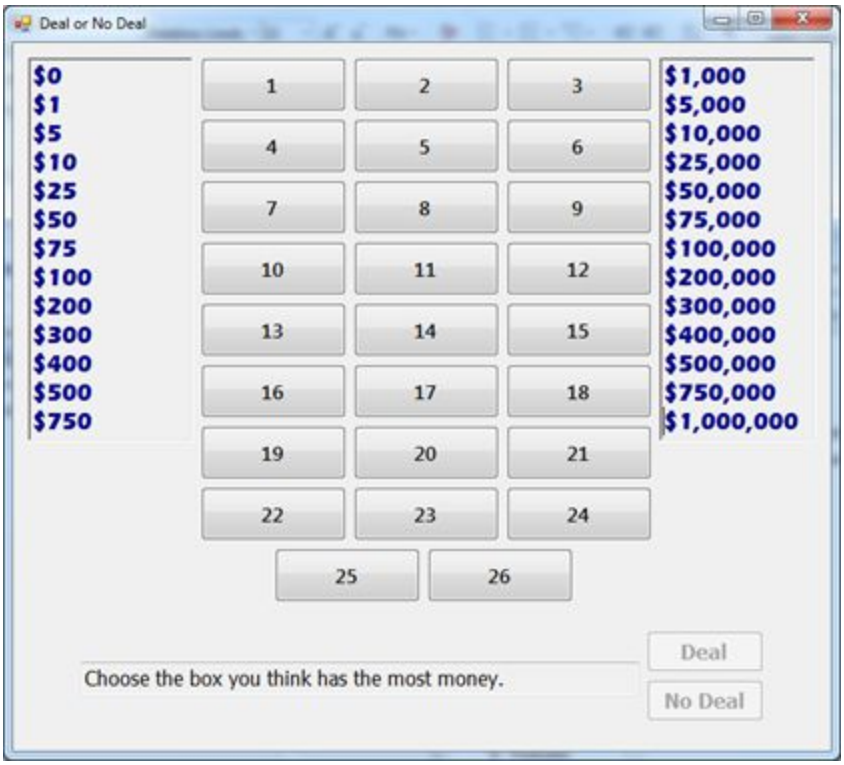
Scoring

Use the skills learned in class.

- 40 points: Variables and expressions
    - Proper variable naming (camelCase)
    - Appropriate use of variables and scope
- 40 points: Branching
- 40 points: Methods
    - Proper method naming (PascalCase)
    - Use of methods (do not copy/paste the same code in multiple places in your application)
- 40 points: Loops
- 40 points: Collections (arrays)
- 50 points: General
    - Program follows design and layout principles, and runs
    - Program does what proposal said it would do

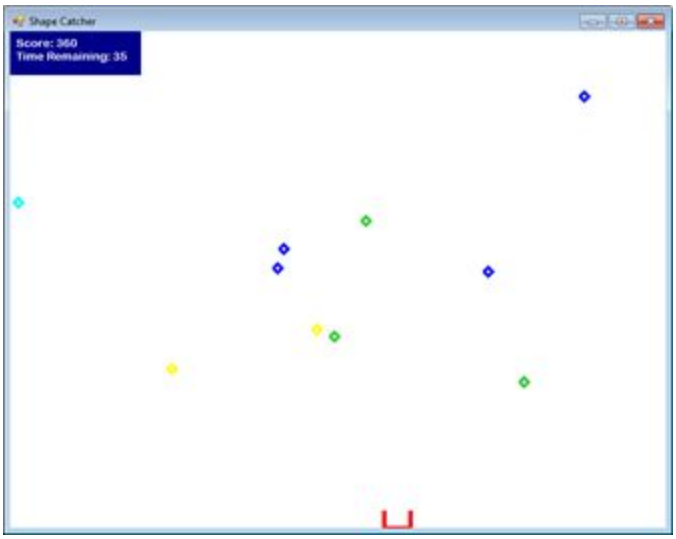
You must understand and be able to explain your code.

Project Suggestion One: Deal or No Deal



This is a program that simulates the "Deal or No Deal" game show. There are 26 boxes, each containing a different amount of money. The player chooses a single box that he thinks contains the most money. That box is saved until the end of the game. The player then chooses six more boxes and opens them immediately. Each opened box reveals the amount of money in that box. The program dims each amount that is revealed so it's easy to tell which amounts are still in play. After opening six boxes, the program makes an offer to the player of some amount of money to quit the game. (The offer should always be the average of whatever amounts are still in play.) The player is then given a choice to take the offer ("deal") and end the game or to refuse the offer ("no deal") and continue with the game. If the player chooses to continue he is instructed to open five more boxes, after which the program makes the player another offer to quit the game (with the same formula and rules). The following rounds (if the player chooses to continue the game) then have the player opening four boxes, then three boxes, then two boxes before making another offer. After that, the player opens only one box for each round. Once the player opens the last box he can open the box that he chose (and was saved) at the beginning of the game to see how much money he won. If the player decides to take the deal and quit the game, the program allows the player to open any of the remaining boxes to see what values are contained there, without changing the amount of money the player wins.

Project Suggestion Two: Shape Catcher



This is a game where shapes are dropped at random times from random locations at the top of the window. The shapes then fall towards the bottom of the window, where the player tries to catch them with a container. The player controls the container by moving the mouse to the left and right.

The shapes fall at different speeds, and the faster shapes are worth more points. When the player catches a shape, the program will play a sound and update the score.

The game is timed and the player tries to get the highest score possible before the time runs out.

Project Suggestion Three: Your Choice

Instead of choosing one of these two examples, you are encouraged to come up with your own idea for a game or other useful program. You can start with one of the above examples and do something a little differently, or you can come up with something completely original.

Other Project Suggestions

- Chess
- Galaga
- Mario!
- Snake
- Pong
- Doom-like game
- Flappy bird
- Asteroids
- Tower defense
- Hangman
- Draw 4
- Risk
- Smash land
- Mr. Potato head
- Checkers
- 2048
- Basketball shooting game
- Minesweeper
- Connect four
- Shade spotter
- Tetris

Note: The class suggestions may or may not qualify for a final project. Use them as suggestions. It all depends on your project proposal.