**Project 1 Rubric**

|  |  |  |
| --- | --- | --- |
| **Item** | **Points** | **Score** |
| Does the project you turned in compile? Does it compile without warnings? | 1 | 1 |
| Are the required tasks all complete? |  |  |
| 1. User can play the game of Set |  |  |
| 2. Cards fly in when game is first dealt | 1 | .5 |
| 3. Cards shrink/grow appropriately to fit available space | 1 | 1 |
| 4. Cards maintain consistent aspect ratio | 1 | 1 |
| 5. Symbols scale with the card | 1 | 1 |
| 6. User can select/deselect cards properly (up to 3, selection is clear) | 1 | 1 |
| 7. Match is indicated to the user when 3 cards match | 1 | 1 |
| 8. User can deselect cards when only 1 or 2 are selected | 1 | 1 |
| 9a. Touch after 3 cards match replaces matched cards with newly dealt cards | 1 | 1 |
| 9b. Touch after 3 cards match causes matched cards to fly away (animated) | 1 | 1 |
| 9c. Touch after 3 cards match causes new cards to fly in (animated) | 1 | 1 |
| 9d. If deck is empty, vacated space is made available to remaining cards | 1 | 1 |
| 9e. Touch selects new card if it wasn’t part of the matched set | 1 | 1 |
| 10. When 3 cards selected don’t match, touch deselects them, selects new card | 1 | 1 |
| 11a. Deal 3 button replaces an existing matched set when appropriate | 1 | 1 |
| 11b. Deal 3 button simply adds 3 new cards to the table if there is no match | 1 | 1 |
| 11c. Deal 3 button is disabled when there are no cards available to deal | 1 | 1 |
| 12. New game button starts over with animation | 1 | 1 |
| 13. Shapes are implemented correctly: diamond, oval, squiggle | 1 | 1 |
| 14. “Striped” is implemented with shading that looks good and is clear | 1 | 1 |
| 15. Colors are clearly distinguishable from each other | 1 | 1 |
| 16. An enum is a meaningful part of the code | 0.5 | 0.5 |
| 17. A closure is a meaningful part of the code | 0.5 | 0.5 |
| 18. UI works in portrait or landscape orientation | 1 | 1 |
| Is the coding style consistent? Did the student use whitespace consistently (indenting consistently, dividing code sections consistently, using spaces between operators consistently, etc.)? Is the code sloppy or otherwise hard to read? Too many comments? Not enough comments? | 1 | 1 |
| Is the UI beautiful? Are elements neatly aligned, sized, and balanced? | 1 | 1 |
| Was the work done on time? (20% penalty per day or fraction of a day late) |  |  |
| Other excellent extras? (Explain.)  Created a landing page before the first game starts that uses an image group with the set logo. | 0-3 | 1 |
| Other problematic aspects? (Explain.)  Went over this bug with Dr. Liddle, but once the columns have adjusted once, if you start a new game only 9 cards show up and the other three are hidden until you deal 3 more, then all 15 show up. I spent hours trying to fix it, and made the changes that Dr. Liddle suggested (using two arrays for dealt and remaining cards) and still couldn’t figure out the bug. |  |  |
| **Total** | ≤ 25 | 25 |

Are there any questions in your mind about your solution? \_\_x\_\_ Yes \_\_\_\_ No

If yes, write your specific questions on the back of this rubric. If it’s about a programming technique, it may help to attach a marked-up printout of the code in question.

See problematic aspects, the occasional bug with rendering all 12 cards.

Your score total must be 25 or less. You don’t need to do anything extra to get 25 points. If you did something extra that is nice, it might compensate for flaws elsewhere. If you specify extra points, it needs to be for something that really is “excellent”, not just interesting.

**Evaluate your Project 1 experience (could be rose/bud/thorn or just a paragraph or two telling me how it went for you):**

I really enjoyed project 1. Even after watching tutorial videos and lectures from Dr. Liddle I still think it takes doing something on your own to really understand what is going on. I think the most important thing that I learned was how MVVM is organized and designed. I never previously understood how the responsibilities of the application were divided between the Model and the ViewModel but I think I understand it pretty well now.

The backend portions of building the model and ViewModel came pretty easily. The hardest parts were understanding how to build layouts, how to programmatically draw ui elements (shapes and cards) with loops that are dependent on ViewModel information, and messing around with spacing and sizing for the columns and rows. Overall I was able to figure most of it out which was super confidence inspiring and I feel like I could build all sorts of things with swift now, which is a good feeling. Definitely going to add the set game to my portfolio!