**Haunted House Project**

Before you begin, take the time to read and reread the directions for this project. It’s important to understand the rules before you jump in. Any misunderstandings can be addressed early, not on due dates. Following directions is paramount to success in anything you do in or out of school.

You will design a game that allows the user to enter no less than Rooms (8) rooms in search of a bag or bags of candy or some sort of treasure. The user must be met with at least one obstacle per room in their search for the candy. The bag of candy or treasure can be hidden. There can be more than one bag of candy in the house. When the user collects all the candy, they win the game and the game is over. The game can be over by the user going into the wrong room and facing an obstacle that they made the wrong decision on how to get around it. In which case, the game can be restarted and played again by the user, this time with new information that will help him or her play at a higher level. A timer or some kind of score keeping device will be something extra to add pressure and competition for the user, but is not necessary.

You will be working on the project with a partner. The project completion date is April 11th, 2018. Presentations will begin Thursday April 12th and finish on Friday April 13th.

The project should include **most** of everything we have learned up to this point. Which includes but is not limited to: Forms, Buttons, RadioButtons, Checkbox, GroupBoxes, Labels, PictureBox, ComboBox, TextBox, CheckedListBox, Button, GroupBox, If Then Else, Try Catch, Select Case, Random Number Generator, Constants, Variables, Data Types, MessageBoxes, InputBoxes, Decision Making Structures, Boolean Expressions, and anything else not mentioned in this rubric. If we covered it, it should be a part of your project. There is one caveat; if something doesn’t fit, don’t use it. Don’t make your project clunky because you wanted to use all the objects, variables, operands and properties that we learned. Students’ may use parts or whole programs from past assignments that we did in class. This should greatly improve development and debugging times.

Your mission is to create a multiform game. Each room is its own form. You can also have sub rooms off those main rooms, for instance a closet in a bedroom, a butler pantry off the kitchen or an upstairs bathroom off the master bedroom. The bag of candy must be challenging to find, and this can include putting a puzzle together, solving a riddle or something along those lines.

**The suggested rooms are:**

(You need at least 6 rooms to meet the specifications of the rubric)

1. Foyer
2. Kitchen
3. Den
4. Downstairs / Upstairs Hallways
5. Master Bedroom
6. Children Bedroom
7. Basement
8. Attic
9. Garage
10. Hidden room behind a wall.

You have total control over the design and content. Utilize all that you currently know and feel free to use the sample programs that we have done in class to guide you through the process.

**GRADING PROCESS:**

There will be three sections for this project.

The first section is based on your **storyline**, which can be laid out in a bullet format for each room, as we discussed in class. **20%**

The second section will be based on the **code** and use of the objects as per the rubric. **40%**

The third section is based on your presentation of your project to the class. Take time to rehearse what you and your partner are going to present and how you will say it. Sell the sizzle not the steak. As you demo your game, I expect you to talk about the logic that you built into the code and why you decided to take that approach. Lastly, students must reflect on the process. Ask yourselves what went right, what went wrong, what changes you would make if you were to get this assignment again in the future. This reflection must be included in your presentation. **40%**

**RUBRIC**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 |
| Problem Solving  **5%** of Coding Grade | Little or no logic employed | Numerous errors when solving problem | Less than 5 errors | No errors |
| Strings and  ToString  **5%** of coding  Grade | Does not include Strings | Includes some strings but they don’t work | Employs 10 strings | Strings function correctly |
| Random number generator  Values  Integers  **5%** of coding grade | Does not employ the rand/val/int functions | Includes rand/val/int but do not function correctly | Has rnd/val/int but some used incorrectly | Has rnd/val/int and they work perfectly |
|  |  |  |  |  |
| Story Line  **20%** of your overall Grade for this project | No Story Line | Has Story Line but is not detailed | Some rooms are detailed, some are not | All rooms are laid out with enough detail to support the flowchart and coding effort |
| Variables  **5%** of coding Grade | Clearly does not understand the use and syntax for variables | Some variables coded correctly some not | Most variables coded correctly | All variables coded correctly |
| Use of  Forms,  Labels,  TextBoxes, MessageBoxes, RadioButtons, CheckBoxes, CleckListBox,  MenuStrips,  Buttons,  PictureBoxes,  If Then Else  Select Case  **15%** of coding Grade | Inappropriate use of controls. Does not convey the student understands how these controls are used. | Some controls coded correctly | Most but not all coded correctly | Exquisite use of controls |
| **5%** Partnership | No synergy generated between partners | Some evidence of better work based on work done together | Clearly both partners learned from the experience of working together | Project includes work that was not covered in class and is evidenced by the partnership working together. |
| Presentation  **40%** of coding Grade | No presentation given | Presentation was not clear did not follow a script. Winged it. | Presentation contained elements of a script but went off script | Presentation was well organized and rehearsed. |