Web Development – Mr. Goldman

Level 3 Project – Safecracker

**Project Overview**

You are a safecracker who is trained in cracking specialized vaults. As you progress, each of the vaults you encounter will become harder to crack, the codes becoming longer and the possibility of the police arriving becoming greater.

**The Pages**

The game will run during a series of rounds. Each round, the player will attempt to crack a more difficult vault and earn some money. Each round will also be composed of turns. All vaults are equipped with an alarm that is set to go off a certain number of turns after the player begins to try and crack the vault. Incorrect entries can speed up the alarm. Once the alarm goes off, the player has 3 more attempts to crack the vault and escape before the police arrive. The user may not escape until the safe has been cracked. Afterwards, escape is automatic.

While trying to crack the vault, the user may enter a combination or use an item. The interface for this is up to you. Whatever method of interface you choose, your program must adhere to the following rules:

* Only an entry that is the exact same length as the combination will be accepted as an attempt to crack the code.
* If the entry is incorrect, the user will be told whether the numeric value of the combination is higher or lower than the numeric value of the attempt.

**Arrest**

If the alarm clock goes off and the player uses up his or her 3 more attempts, the police will arrive. At this time, the player will have one opportunity to use an item or accept arrest.

Once the player is arrested, the game ends. The player’s final score is the amount of money (s)he has.

**Vault Combinations**

The combination of a vault is composed of X digits, each digit ranging from 1 to no greater than 9. ***No combination may include zeros***. At the beginning of the round, the combination will be created randomly. The user will be told how many digits are in the combination and what the range of numbers is. For example, in the first round, the user will try and crack a code of 3 digits, each digit ranging from 1 to 3.

***Note for Programmers: You cannot simply generate a random number between 111 and 333 because you may wind up with zeroes. You will have to generate each digit in the code separately, concatenate them together, and convert them back to a number.***

**Time**

The time on the clock will be determined by round. There will be a base number of attempts for each round, representing how much time a user has to crack a code. That number will be modified by a random number (based on the round number).

Refer to the chart below for the formula.

The time that comes off the clock will also be random. Each time the user unsuccessfully tries to crack the code, the possibility of more time coming off the clock grows. The time that comes off the clock will be based on a random number between 0 and the number of attempts to crack the code.

On the first attempt, it might take no time or it might take 1 second. On the second attempt, it might take 0, 1, or 2 seconds. And so on…

| Round | Number of Digits | Range of Digits | Time on clock. |
| --- | --- | --- | --- |
| 1 | 3 | 1-3 | 5 + (0 - 1) |
| 2 | 3 | 1-4 | 6 + (0 - 2) |
| 3 | 3 | 1-5 | 7 + (0 - 3) |
| 4 | 3 | 1-6 | 8 + (0 - 4) |
| 5 | 4 | 1-4 | 9 + (0 - 5) |
| 6 | 4 | 1-5 | 10 + (0 - 6) |
| 7 | 4 | 1-6 | 12 + (0 - 7) |
| 8 | 4 | 1-7 | 14 + (0 - 8) |
| 9 | 5 | 1-5 | 16 + (0 - 9) |
| 10 | 5 | 1-6 | 18 + (0 - 10) |
| 11 | 5 | 1-7 | 20 + (0 - 11) |
| 12 | 6 | 1-6 | 22 + (0 - 12) |
| 13 | 6 | 1-7 | 24 + (0 - 13) |
| 14 | 6 | 1-8 | 26 + (0 - 14) |
| 15 | 7 | 1-7 | 28 + (0 - 15) |
| 16 | 7 | 1-8 | 30 + (0 - 16) |
| 17 | 7 | 1-9 | 33 + (0 - 17) |
| 18 | 8 | 1-8 | 36 + (0 - 18) |
| 19 | 8 | 1-9 | 39 + (0 - 19) |
| 20 | 9 | 1 -9 | 42 + (0 - 20) |

After round 20, the number of Digits increases by 1 while the range is always 1 - 9. The time on the clock always increases by 3 turns with the random range increasing by 1.

When a player cracks a vault, (s)he finds money. The amount of money found is based upon the round. The formula for calculating the amount of money in a vault is the round number \* (1 - 6) \* 100. So that in the first round, the user will find $100, $200, $300, $400, $500, or $600.

In between each round, the user can spend his/her money at the Safecracking Store. A user can hold only one item unless he or she buys a bag. A bag can hold a certain number of items based on the size of the bag ***(a bag may not hold another bag).*** Item costs and descriptions are in the chart below.

| Item | Cost | Description |
| --- | --- | --- |
| Lockpick | $1,000 | A lockpick has 1 use only and will reveal 1 random digit from the code. |
| Dynamite | $5,000 | Dynamite has 1 use only. It will blow open a safe automatically. However, a random amount of money inside the safe may be destroyed. In order to determine the amount of money destroyed, first determine the amount of money inside the vault. Afterwards, use the same formula to generate another amount and subtract it from the first. |
| Lawyer | $50,000 | A lawyer has 1 use. If the player is caught by the police, (s)he can buy his/her freedom and continue cracking safes. A player must have a lawyer in his/her possession when caught. |
| Small Bag | $7,000 | Holds 2 Items (Bags cannot hold other bags). |
| Medium Bag | $14,000 | Holds 4 items (Bags cannot hold other bags). |
| Large Bag | $28,000 | Holds 7 items (Bags cannot hold other bags). |
| Magnet | $3,000 | Will add 0 - 5 seconds to the clock. Each time a magnet is used, there is a 10% chance that it will set off the alarm rather than add time. There is also a 40% chance that the magnet will break and become unusable again. |
| Stealth Suit | $20,000 | The player can choose to escape at any time, regardless of whether or not the safe has been cracked. A stealth suit has a lifetime of 1 – 4 uses (generated when bought) before it wears out and is unusable. |
| Henchman | $75,000 | A Henchman will hold off the police while a safecracker continues to work on the safe. A henchman can last anywhere from 1 - 6 turns against the police (generated at the start of the battle), giving the player that many more chances to crack the safe (the seconds on the clock no longer matter). After those turns are up the henchman surrenders and is hauled away. The Henchman will begin battling police as soon as the 3 turn grace period is up and the countdown to each henchman's surrender will begin. If the player has more than one henchman in a bag, they will fight police separately so that only one will be captured at a time. Once all of the henchmen are gone, the player is arrested. |

**Programming Requirements**

|  |
| --- |
| * You must code your own HTML and CSS. Use of a drag and drop interface or the usage of code downloaded from the internet is not permitted. |
|  |
| * Use of deprecated code is not permitted. Use an HTML 5 reference as your guide. |
|  |
| * Your home page must be named *index.html*. |
|  |
| * You must comment your name into the top of every page. |
|  |
| * Your code must be structured in a consistent and legible manner |
|  |
| * Your text must be organized through the use of sectioning tags (div, span, p, h1 – h6). Tables are acceptable for data but not for page design. |
|  |
| * Your interface must be smooth and easy to figure out for a client. |
|  |
| * Your pages must be formatted using CSS. |
|  |
|  |

*You project is to be submitted via Moodle. Compress all of the necessary files into a zip or rar file.*