Web Development – Mr. Turner / Ms. Moran

Project – Safecracker

**Project Overview**

You are the notorious safecracker, Sinister Von Vaultzenshtein. Hidden somewhere in the city of Wealthopolis, there is the finest set of jewels ever cut by humankind. The mayor of Wealthopolis has declared that no one could ever crack through all of the vaults of her city and steal the jewels. Challenge accepted. Work your way through the vaults of Wealthopolis until you discover the jewels and retire into the lap of luxury.

**The Page**

Design and implement the *Safecracker* game.

Throughout the course of the game, the player will be called upon to figure out the combinations of vaults throughout the city. Each vault combination will be more complex than the last. If the player cracks the vault in time, he or she will be rewarded with money. Otherwise, the police will come and it’s off to prison.

**The Vault**

Each vault has a combination of X digits, each digit ranging from a minimum of 1 to a maximum of 9, depending on the level of the vault.

Each combination will be created randomly. Refer to the chart below to determine the number of digits in the combination and the range of digits. Display this information on the page.

*No combination may include zeros.*

*Example: During the first round, the player will encounter a vault that has a 3 digit combination. Each digit will range from 1 to 3.*

***Remember that there are no zeroes so a simple random number will not work.***

As the player enters the code and makes his or her guesses, the computer will tell the player whether the guess was too low or too high or correct.

|  |  |  |
| --- | --- | --- |
| Vault Number | Number of Digits | Range of Digits |
| 1 | 3 | 1-3 |
| 2 | 3 | 1-4 |
| 3 | 3 | 1-5 |
| 4 | 3 | 1-6 |
| 5 | 4 | 1-4 |
| 6 | 4 | 1-5 |
| 7 | 4 | 1-6 |
| 8 | 4 | 1-7 |
| 9 | 5 | 1-5 |
| 10 | 5 | 1-6 |
| 11 | 5 | 1-7 |
| 12 | 6 | 1-6 |
| 13 | 6 | 1-7 |
| 14 | 6 | 1-8 |
| 15 | 7 | 1-7 |
| 16 | 7 | 1-8 |
| 17 | 7 | 1-9 |
| 18 | 8 | 1-8 |
| 19 | 8 | 1-9 |
| 20 | 9 | 1-9 |

After vault 20, the number of Digits increases while the range is always 1 - 9.

**The Timer**

Each vault will have a simulated timer. Instead of actual time, the player will get a certain number of attempts to crack the code. The number of attempts the player gets begins at 5 with the first vault and goes up by 1 each round. At every 10th round, the number of attempts added to the “clock” increases by 1.

*The player gets 5 attempts round 1, 6 round 2, 7 round 3, etc… The number of attempts is increasing by 1. For round 9, the player will have 13 attempts. For round 10, the player will have 15 attempts. For round 11, the player will have 17 attempts, and so on. For round 19, the player will have 33 attempts. For round 20, the player will have 36 attempts.*

In addition, add a random number of attempts each round. The range of this random number is between 0 and the number of the current vault.

Each time the player enters a code, countdown the number of attempts to 0.

**The Alarm**

Once the number of attempts remaining on a vault is reached, the alarm goes off. The player now has 3 more attempts before the police arrive. Once the police arrive, the player is arrested and the game is over.

**Cracking the Code**

If the player should discover the proper combination, he or she has cracked the vault and reaps the rewards.

Each vault has money in it. The amount of money found is based upon the vault number. The formula for calculating the amount of money in the vault is:

*vaultNumber \* (a random number between 1 and 6) \* 1000*

In between rounds, the player can take this money and spend it in the *Safecracker Store* or simply move onto the next round.

For each vault the player cracks into, there is a possibility that he or she will find the jewels. Whether or not the jewels are found is based on a growing percentage. The chance that the player will find the jewels is exactly *vaultNumber* percent. This means that the player has a 1% of finding the jewels on the first turn, 2% on the second turn, etc… The best chance the player should ever have of finding the jewels is 99%.

**The Safecracker Store**

There are a number of items available for good safe crackers to use while working. Each item has a specific cost and a specific use. A player may only have 1 item at a time. Refer to the chart below for the cost and description of each item.

| **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 (calculate the amount of money in the safe based on the formula and then calculate how much was destroyed). If the jewels are in the safe, they, too, are destroyed. In a fit of despair the player turns him or herself into the police and loses the game. |
| **Magnet** | $2,000 | Will add 0 - 5 attempts 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. |
| **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. |
| **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. |

**The Interface**

The entire page will be controlled by buttons. The final design is up to you, but there are some guidelines that you must follow.

The Vault

* There will be a button for each possible digit in the combination. *Do not show buttons that are not available in that combination.*
* Display the vault number.
* Display the number of digits in the code.
* Display the code that the player enters.
  + Remember that the player is clicking buttons 1 at a time in order to enter his or her code. Each button clicked will add a number to the code. When the length of the number entered equals the number of digits in the code, that’s when you should check to see if they are the same.
  + Add a clear button so that the user can clear a partial code and start again.
* Display whether the guess is too low or too high or correct.

The Player

* Display the player’s total money.
* The player should have a button that allows him or her to use his or her item.
  + There should only be 1 button and its text should say “Use *(whatever item the user currently has in possession)*”.

The Store

* In between rounds, the vault should either be disabled or replaced by the store (which should either be disabled or replaced when the user is safecracking).
* Each item for sale should have a button. If the player doesn’t have enough money for the item, disable the button.

Losing

* Make a nice screen for the arrest.

Winning

* Make a nice screen for when the jewels are found.

Starting Over

* A Reset button should be available at all times during and after the game. The reset button will clear all of the player’s money and items and will bring the player back to the first vault.

**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. |
| * 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. The file should have the following name:*

*First Name Last Name Safecracker*