



HANGMAN

BY: KENYANA TYLER, PATRICK SNOW, JOHN MCLEOD

OVERVIEW

- What we decided to do: Hangman Trivia Game
- As usual, the player was to guess the letter of an unknown word (its length revealed by “_”) while allowed to make up to five wrong guess: each of which would show another body part until the man was completed and hung.
- To give the conventional hangman game an interesting twist, we incorporated trivia. In the spirit of the course, it was going to be eventually implemented as a Java-like trivia.

THE TEAM

- Parts- John worked with the HMGUI, Patrick and myself worked with the HMChecker class and the methods that went along with that.
- Loss of 25% with Cierra- Since we lost a partner(though it was not her choice due to the military), we had to decide what parts of the code to keep and what to take out. We didn't think we would be able to make the game as fun as we wanted to due to the loss of a partner. Everyone was somewhat assigned a different part as far as coding. Some people received more work and some received less. But in the end we equaled everything out so we wouldn't have to depend on the person with the most work to get everything done.

COMMUNICATION(KENYANA)

- Shared Docs – Before GitHub we used Microsoft word to share documents with one another. Once person would created the document and share it with the rest of us.
- Email – After we had to begin using GitHub, we started communicating daily by email about the project.
- GitHub – We pushed different documents to GitHub as well as code once we started the coding portion of the project.

ROADBLOCKS AND RE-ROUTING (JOHN)



The typical *road blocks*, like group decisions, online/in-class member mix, time constraints, were outweighed with the loss of a team member. Complications:

- Unaware she wasn't going to complete her class until after designed decisions were made.
- Her class was the Checker, so the hub and central to the other class's purposes.

We *re-routed* by reduce to the core code, while trying to keep the Open/Closed principles.

- Dropped many features, like Bonus clue (per dialog box) but allowed as an extension but did not modify and further what we had.

ROADBLOCKS AND RE-ROUTING (JOHN)

Originally, five classes (GUI, Bonus, Checker, & WordBank) down to two:

HMGUI and HMChecker.

HMChecker, original purpose: GUI and event capture

- Re-routed to handles some of logic of picture advancement and game status (won or loose).

HMChecker, original purpose: process correction of letter chosen & hub for other classes.

- Re-routed, to include label updating & to store and randomly select challenge word and its corresponding trivia.



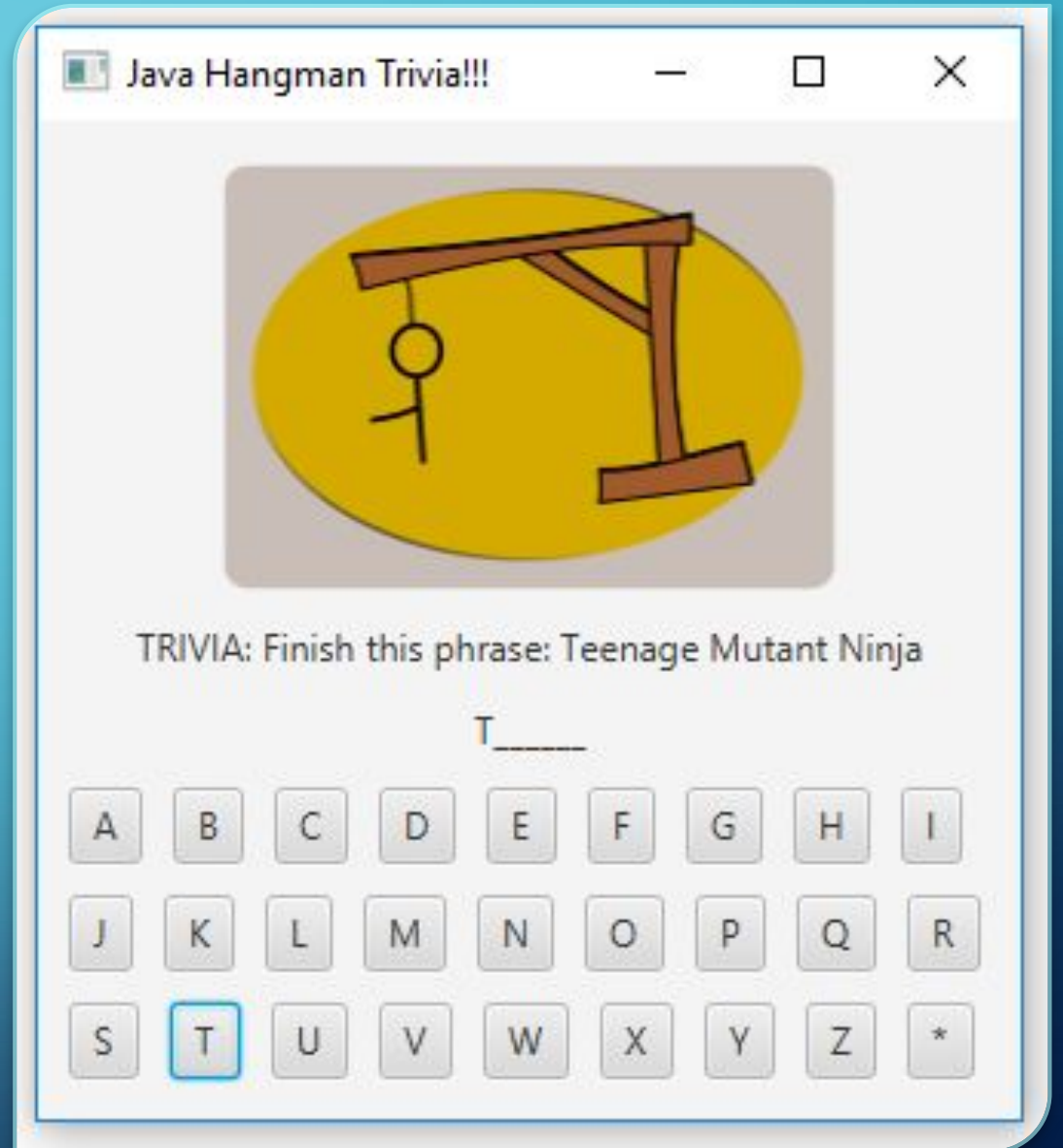
HMGUI

Created using JavaFX

Simple layout

- Buttons, ImageView, and Labels
- Buttons in HBoxes

Controls and HBoxes in a VBox



HMCHECKER(PATRICK)

- Methods: Boolean isCorrect, Boolean isCompleted, String getTriviaQuestion.
- HMGUI would call getTriviaQuestion first to start the game by picking a random question from a string array. getTriviaQuestion will then set a global variable to hold the answer matching the selected question with its length.
- isCorrect when called will return the Boolean value if it matches with the character at the specific index.
- isCompleted returns false until the player has correctly guessed all of the characters in the string.

SCREENSHOT OF CODE

```
public boolean isCorrect(int pos, char ch){

    boolean correctResult = false; //STUBBED OUT
                                   //TO BE IMPLE

    //checks to see if the char at pos is equal
    //if it is true then it will return true and
    //to one less one word to guess to complete
    if (answer.charAt(pos) == ch){
        correctResult = true;
        complete--;
        update++;
    }

    return correctResult;
}

//This method checks if the choice made by the p
//that is, has he completed the challenge word a
public boolean isCompleted (int pos, char ch) {
    boolean challengeWordCompleted = false;
    //I think isCompleted doesn't need pos and c

    //If the complete equals 0 then it means tha
    //guess all of the words correctly after num
    //If complete has not hit 0 yet then it will
    if (complete == 1){
        challengeWordCompleted = true;
    }
    return challengeWordCompleted;
}
```


SCREENSHOT OF CODE (CON.)

```
//After the question gets picked it will change the global variable of answer
// to the corresponding answer that matches with that question.
String[] answers = new String[12];
    answers[0] = "MJOLNIR"; /* MJOLNIR*/
    answers[1] = "LOKI"; /*LOKI*/
    answers[2] = "SIX"; /* SIX*/
    answers[3] = "JACKSPARROW"; /*JACK SPARROW*/
    answers[4] = "VELOCIRAPTOR"; /*VELOCIRAPTOR*/
    answers[5] = "BABERUTH"; /*BABE RUTH*/
    answers[6] = "ROBIN"; /*ROBIN*/
    answers[7] = "MOWGLI"; /*MOWGLI*/
    answers[8] = "TURTLES"; /*TURTLES*/
    answers[9] = "BEAST"; /*BEAST*/
    answers[10] = "KINGKONG"; /*KING KONG*/
    answers[11] = "ANAKIN"; /*ANAKIN*/

    answer = answers[index];
    complete = answer.length();

return combinedText;
}
```

HMCHECKER(KENYANA)

- Method: `updateChallengeWord()`
- This method updated the word that the user was trying to guess. When the game started the answer would be in the format of underscores(----) specifically the length of the word. Every time a person guessed a letter right the letter would be displayed.
- Example: If the word was apple: (a____), (ap____), (app___), (appl_), (apple)

SCREENSHOT OF CODE

```
63 public String updateChallengeWord(){
64
65     String updatedText = ""; //SUTBBED OUT FOR TESTING!!!!!!!!!!!!!!
66                                     //TO BE IMPLEMENTED BY Kenya & Patrick
67                                     //NOTE: Should start with "_ _ _" at the
68                                     //beginning of the game and underscores
69                                     //should match the number of letters or
70                                     //length() of the word.
71
72
73     //This is a little template but I know there is a better to update the word.
74     //Then this. But this is the direction I think this method is trying
75     //to go towards.
76     switch (update) {
77     case -1;
78         for (int i = 0; i < answer.length(); i++){
79             updatedText += "_";
80             break;
81         }
82
83         case 0:
84             updatedText = answer.substring(0, 1);
85             for (int i = 1; i < answer.length(); i++){
86                 updatedText += "_";
87                 break;
88             }
89         case 1:
90             updatedText = answer.substring(0, 2);
91             for (int i = 2; i < answer.length(); i++){
92                 updatedText += "_";
93                 break;
94             }
95         case 2:
96             updatedText = answer.substring(0, 3);
97             for (int i = 3; i < answer.length(); i++){
98                 updatedText += "_";
99                 break;
100            }
```

CODE CONTINUE

```
101     case 3:
102         updatedText = answer.substring(0, 4);
103         for (int i = 4; i < answer.length(); i++){
104             updatedText += "_";
105             break;
106     }
107     case 4:
108         updatedText = answer.substring(0, 5);
109         for (int i = 5; i < answer.length(); i++){
110             updatedText += "_";
111             break;
112     }
113     case 5:
114         updatedText = answer.substring(0, 6);
115         for (int i = 6; i < answer.length(); i++){
116             updatedText += "_";
117             break;
118     }
119     case 6:
120         updatedText = answer.substring(0, 7);
121         for (int i = 7; i < answer.length(); i++){
122             updatedText += "_";
123             break;
124     }
125     case 7:
126         updatedText = answer.substring(0, 8);
127         for (int i = 8; i < answer.length(); i++){
128             updatedText += "_";
129             break;
130     }
131     case 8:
132         updatedText = answer.substring(0, 9);
133         for (int i = 9; i < answer.length(); i++){
134             updatedText += "_";
135             break;
136     }
137     case 9:
138         updatedText = answer.substring(0, 10);
```

CODE CONTINUED

```
142     }  
143     case 10:  
144         updatedText = answer.substring(0, 11);  
145         for (int i = 11; i < answer.length(); i++){  
146             updatedText += "_";  
147             break;  
148         }  
149         default:  
150             break;  
151     }  
152  
153  
154  
155  
156     return updatedText;  
157 }  
158
```

CONCLUSION

- In conclusion, the three of us (John, Kenya, Patrick) were able to pull together and still make a fun hangman trivia game. We started with 4 but ended with 3. Overall it was a great teamwork experience for us all and we were able to learn something from it.