

# The Hangman



Michael Reno  
Verbena Haritzah Assidiqi

565907  
565941



# The Game

Hangman Game is a simple, one-player-game. It consists of a stickman figure which will be hanged if the player lose the game.

A random word would appear as asterisks (\*) on the screen and with minimum hints, the player should guess what the word is. If the player guess correctly before the amount of chances is up, they win and will continue with another word. When they guess incorrectly, the hangman will be hung.



# Functionalities

- Time limit, automatically generated according to the amount of letters to be guessed.
- Options to add more words to the game database through the game settings.
- Hints to help solve the missing letters.
- User-friendly interface (GUI).
- ~~MIT License, because happiness is meant to be shared!~~
- GNU License, because happiness is meant to be shared!





# Requirements

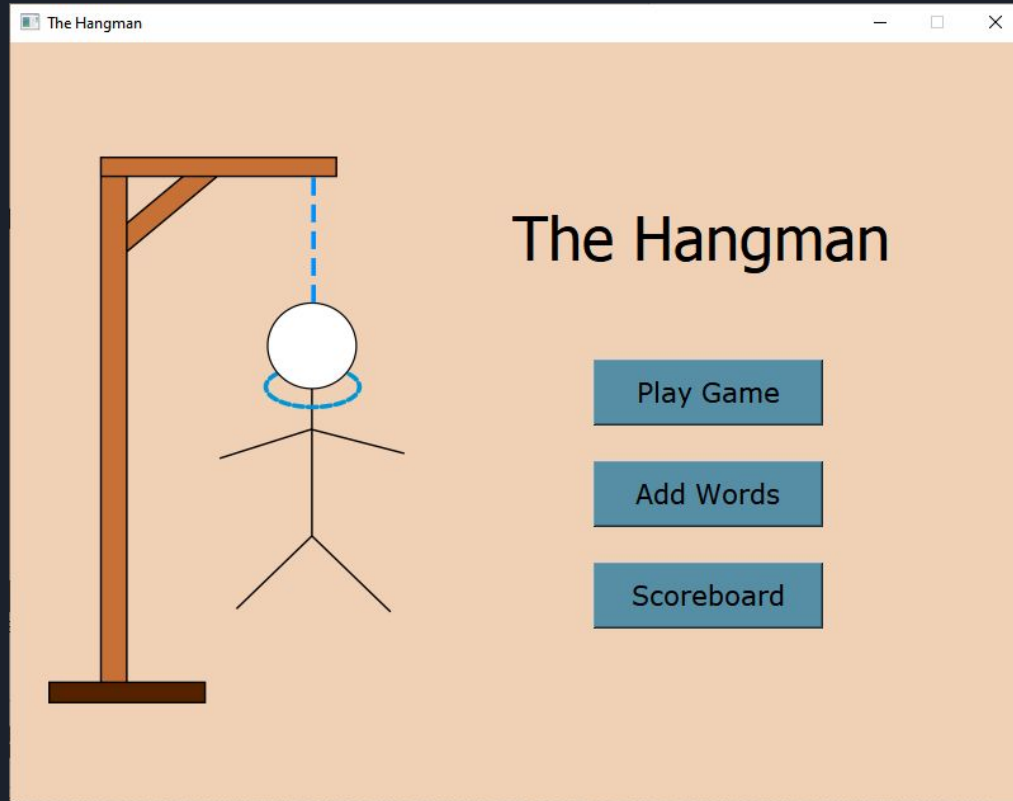
- Qt Creator as the chosen IDE, codes will be written in C++
- Github as the version control
- There should be a minimum of 15 words (or multiple words) in the game database without any addition from the user
- There should be a main menu where the user can either start the game, look at the high scores or add more words
- The time limit should be based on the length of the words (3 mins for a 7 letter word, for example)
- The user will get 10 points for each letter they guess correctly. The player will get an extra 5 points for each letter they guessed correctly in a row.



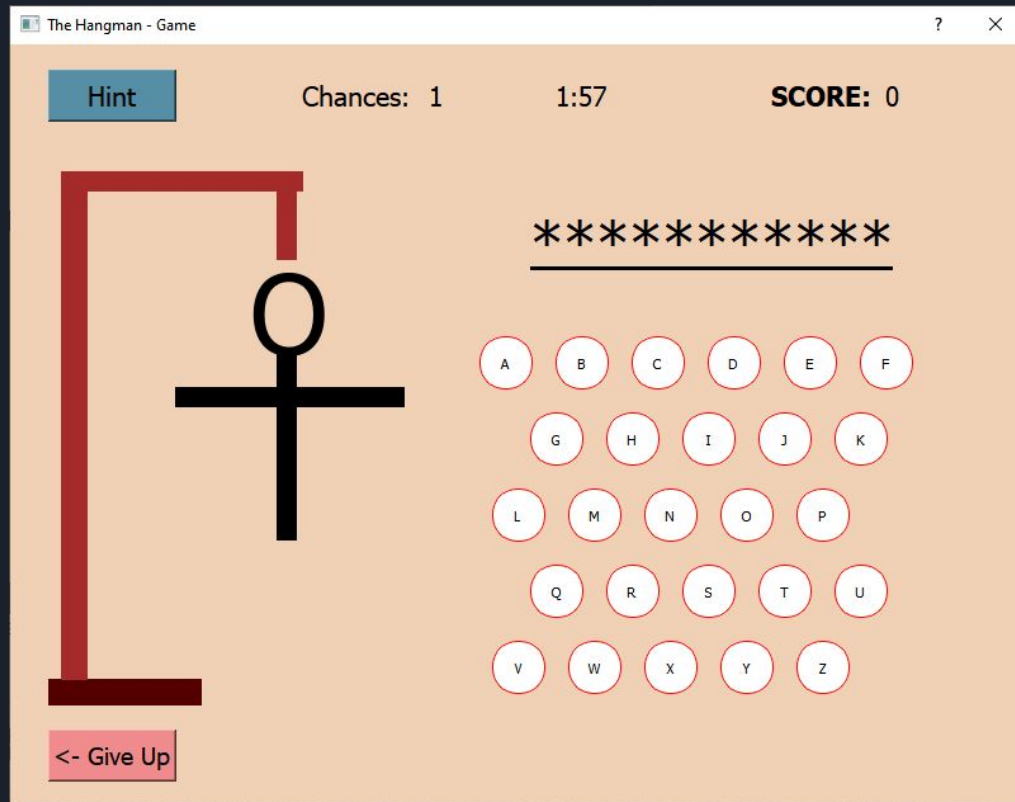
# Time limit cheat sheet

| Number of Letters (n) | Time Limit (in s)    |
|-----------------------|----------------------|
| 3                     | 60                   |
| 5                     | 80                   |
| 7                     | 100                  |
| 9                     | 120                  |
| n                     | $30 + (10 \times n)$ |

# GUI

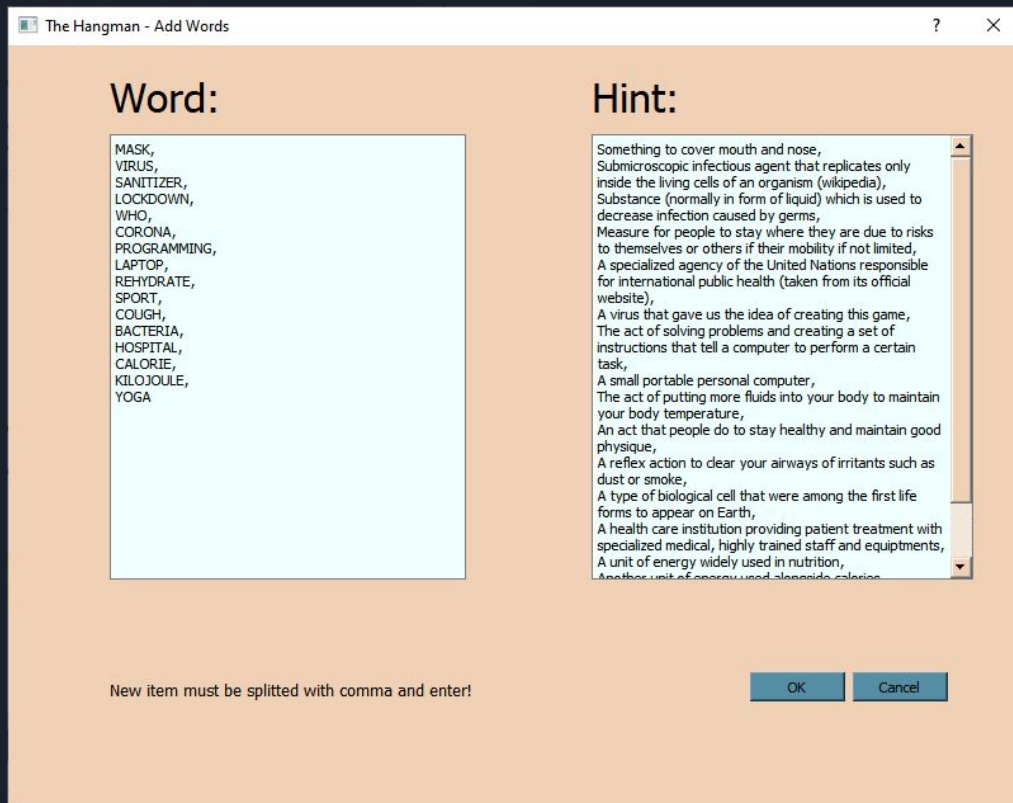


# GUI

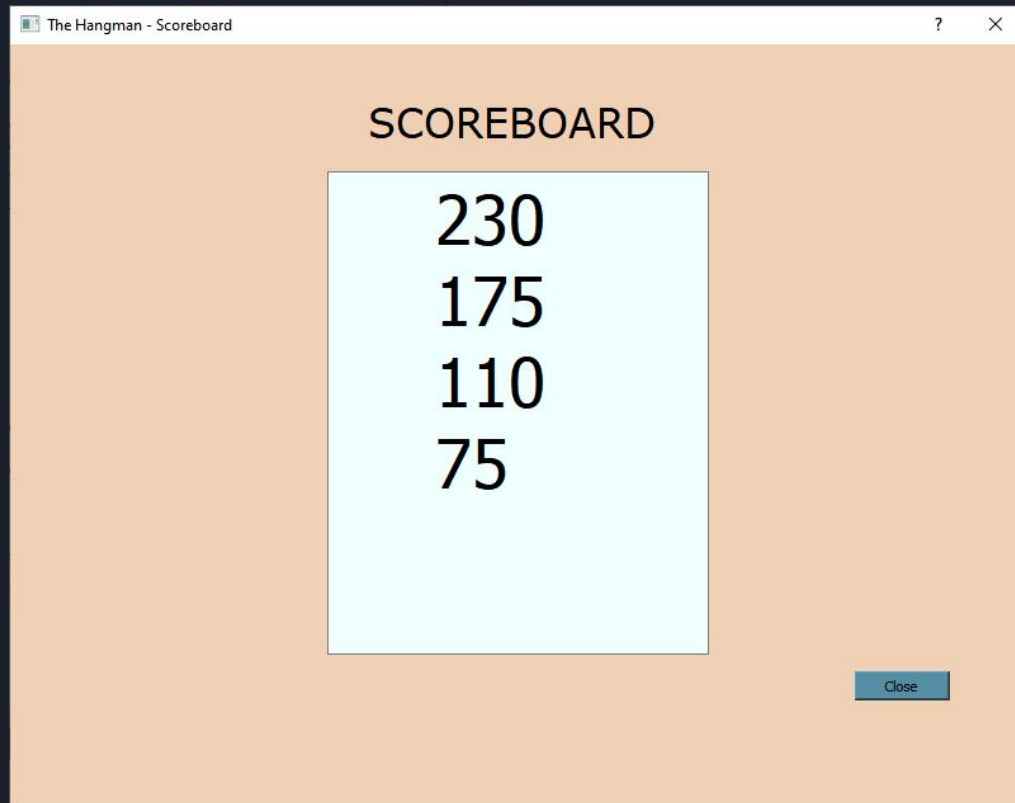




# GUI



# GUI



# Interesting codes

```
// Check whether the input letter match with the puzzle word
// input = character from user, toGuess = puzzle word to be guessed
int gamewindow::check_word(char input, QString toGuess)
{ // Inspiration: http://www.cppforschool.com/project/hangman-game-code.html
  int match = 0;
  //int max_limit = max_guesses;
  for (int i = 0; i < toGuess.length(); i++) {
    // Check with the guessed word whether the letter is already matched
    if (input == guessedWord[i])
    {
      QMessageBox::information(this, "Used letter", "The letter you gave has been entered before.");
      return 0;
    }

    // Is the input correct?
    if (input == toGuess[i]) // When yes
    {
      guessedWord[i] = input;
      consecutive++; // Increase the amount of consecutive guesses to gain higher score
      match++; // To signal that the input is correct
      ui->puzzleWordLabel->setText(guessedWord); // Update label
      // Scoring
      score = data.scoring(consecutive);
      ui->scoreValueLabel->setNum(score); // consecutive stays the same!!!!
      is_finished(guessedWord); // Check whether all the letters are guessed
    }
  }

  return match; // Return an integer to serve as a requirement on the letter_pressed function
}
```

# Interesting codes

```
void gamewindow::letter_pressed()
{
    QPushButton *button = (QPushButton *)sender();
    QString butValue = button->text();
    string input = butValue.toStdString();
    if (check_word(input[0], toBeGuessed) == 0) // When the input letter is not on the puzzle word
    {
        consecutive = 0; // Bring back the amount of consecutive guesses to zero
        wrong_guesses++;
        ui->chanceValueLabel->setNum(max_guesses - wrong_guesses); // Readjust chance limit
        switch (wrong_guesses) {
            case 1: ui->HMLabel1->show();
                    break;
            case 2: ui->HMLabel2->show();
                    break;
            case 3: ui->HMLabel3->show();
                    break;
            case 4: ui->HMLabel4->show();
                    break;
            case 5: ui->HMLabel5->show();
                    break;
            case 6: ui->HMLabel6->show();
                    ui->HMLabel7->show();
                    break;
            case 7: ui->HMLabel8->show();
                    break;
            default: break;
        }
        game_over(wrong_guesses); // Check whether there is any chance left
    }
}
```



# License - GNU General Public License

Copyright (c) 2020 Michael Reno, Verbena Haritzah Assidiqi

GNU LESSER GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

# Vielen Dank!

