

Tollywood Hangman Game: A Cinematic Word Challenge



a.Adithya kumar

Tollywood Hangman Game: A Cinematic Word Challenge

# Tollywood Hangman Game: A Cinematic Word Challenge

Welcome to an exciting adaptation of the classic word-guessing game, Hangman! This unique version, "Tollywood Hangman," brings a delightful cinematic twist by focusing on prominent names from the Telugu film industry, affectionately known as Tollywood. Prepare to test your knowledge of South Indian cinema while enjoying the thrill of a traditional word puzzle.

## What is Hangman?

Hangman is a guessing game where one player thinks of a word, and the other tries to guess it by suggesting letters. For every incorrect guess, a part of a "hangman" figure is drawn. The goal is to guess the word before the entire figure is completed, indicating that the player has run out of attempts. It's a game of strategy, vocabulary, and a bit of luck!

## Tollywood Twist

In this "Tollywood Hangman," the hidden words are the names of famous actors from the Telugu film industry. Each word comes with a helpful hint, usually a well-known title or characteristic associated with the actor. This theme not only makes the game more engaging for Tollywood fans but also offers an entertaining way to learn about or recall these beloved personalities.

## How to Play

The game presents a series of underscores representing the letters of the hidden word. You'll be given a hint to guide your guesses. Your task is to input single letters, one at a time, to reveal parts of the word. Here’s a quick rundown of the rules:

* **Guess a Letter:** Enter one letter at a time.
* **Correct Guess:** If your guessed letter is in the word, it will be revealed in all its correct positions.
* **Incorrect Guess:** If the letter is not in the word, you lose an attempt, and a part of the hangman figure is drawn.
* **Winning:** You win if you guess all the letters in the word before the hangman figure is fully drawn.
* **Losing:** You lose if the hangman figure is completed (meaning you've run out of attempts) before you guess the word.
* **Attempts Left:** The game will keep track of how many incorrect guesses you can still make.

## The Tollywood Hangman Game Code

Below is the Python code that powers this interactive Tollywood Hangman game. This script defines a HangmanGame class that encapsulates all the game's logic, from initializing the word to handling guesses and determining win/loss conditions.

import random  
  
class HangmanGame:  
 HANGMAN\_PICS = [  
 '''  
 +---+  
 |  
 |  
 |  
 ===''',  
 '''  
 +---+  
 O |  
 |  
 |  
 ===''',  
 '''  
 +---+  
 O |  
 | |  
 |  
 ===''',  
 '''  
 +---+  
 O |  
 /| |  
 |  
 ===''',  
 '''  
 +---+  
 O |  
 /|\\ |  
 |  
 ===''',  
 '''  
 +---+  
 O |  
 /|\\ |  
 / |  
 ===''',  
 '''  
 +---+  
 O |  
 /|\\ |  
 / \\ |  
 ==='''  
 ]  
  
 def \_\_init\_\_(self, word\_list):   
 self.word\_list = word\_list  
 self.word, self.hint = random.choice(self.word\_list)  
 self.word = self.word.lower()  
 self.guessed\_letters = []  
 self.attempts\_left = len(self.HANGMAN\_PICS) - 1  
  
 def display\_word(self):  
 result = ""  
 for letter in self.word:  
 if letter in self.guessed\_letters:  
 result += letter + " "  
 else:  
 result += "\_ "  
 return result.strip()  
  
 def guess\_letter(self, letter):  
 letter = letter.lower()  
 if letter in self.guessed\_letters:  
 print("You already guessed '{}'.".format(letter))  
 elif letter in self.word:  
 print("Nice! '{}' is correct.".format(letter))  
 self.guessed\_letters.append(letter)  
 else:  
 print("Oops! '{}' is wrong.".format(letter))  
 self.guessed\_letters.append(letter)  
 self.attempts\_left -= 1  
  
 def display\_hangman(self):  
 print(self.HANGMAN\_PICS[len(self.HANGMAN\_PICS) - 1 - self.attempts\_left])  
  
 def is\_won(self):  
 for letter in self.word:  
 if letter not in self.guessed\_letters:  
 return False  
 return True  
  
 def is\_lost(self):  
 return self.attempts\_left <= 0  
  
 def play(self):  
 print("Welcome to Tollywood Hangman!")  
 print("\nHint: {}".format(self.hint)) # ✅ Fixed: self.Hint → self.hint  
  
 while not self.is\_won() and not self.is\_lost():  
 print("\n" + "-" \* 40)  
 self.display\_hangman()  
 print("Word:", self.display\_word())  
 print("Attempts Left:", self.attempts\_left)  
  
 guess = input("Guess a letter: ").strip().lower()  
 if len(guess) != 1 or not guess.isalpha():  
 print("Please enter a single alphabet letter.")  
 continue  
  
 self.guess\_letter(guess)  
  
 print("\n" + "-" \* 40)  
 if self.is\_won():  
 print("Congratulations! You guessed the word:", self.word)  
 else:  
 self.display\_hangman()  
 print("Game Over! The word was:", self.word)  
  
  
# Word list  
word\_list = [  
 ("NTR", "young tiger"),  
 ("bhAAi", "iconstar"),  
 ("Ramcharan", "globalstar"),  
 ("Prabhas", "rebelstar"),  
 ("mahesh babu", "super star"),  
 ("Balayya", "god of massess"),  
 ("Chirangeevi", "megastar"),  
]  
  
#final result  
game = HangmanGame(word\_list)  
game.play()

## Understanding the Code

The Python script is designed with clarity and modularity. Here's a brief breakdown:

* **HANGMAN\_PICS:** A list of ASCII art strings that visually represent the hangman at different stages of incorrect guesses.
* **\_\_init\_\_(self, word\_list):** The constructor initializes the game, randomly selecting a word and its hint from the provided word\_list. It also sets up variables for guessed letters and remaining attempts.
* **display\_word():** This method shows the current state of the word, revealing correctly guessed letters and obscuring unguessed ones with underscores.
* **guess\_letter(self, letter):** Handles a player's letter guess, checks if it's correct or incorrect, updates the guessed letters, and decreases attempts if the guess is wrong.
* **display\_hangman():** Prints the appropriate hangman ASCII art based on the number of attempts left.
* **is\_won() & is\_lost():** These boolean methods check if the player has won (all letters guessed) or lost (no attempts left), respectively.
* **play():** This is the main game loop. It welcomes the player, displays the hint, continuously prompts for guesses, and updates the game state until a win or loss condition is met.
* **word\_list:** This list stores tuples, where each tuple contains a Tollywood actor's name and their associated hint.

## How to Run the Game

To play the Tollywood Hangman game, simply save the provided code into a Python file (e.g., hangman\_game.py) and run it from your terminal:

python hangman\_game.py

The game will then prompt you to guess letters, providing an interactive command-line experience. Enjoy diving into the world of Tollywood and challenging your word-guessing skills!