This code is for a Hangman game written in **Python**.

Let me explain all the function it step by step: -

**hangman\_graphics:** This is a list of strings that represent ASCII art for different stages of the hangman drawing. Each string in the list corresponds to a different stage of the hangman's appearance as the player makes incorrect guesses.

**select\_word(category):** This function takes a category (like “technology” or “sports” or ..) as input and returns a random word from that category. It contains predefined lists of words for each category. If the category is not recognized, it prints “Invalid category!” and returns None.

**display\_hangman(incorrect\_guesses):** This function takes the number of incorrect guesses as input and prints the corresponding hangman ASCII art from the hangman\_graphics list.

**display\_word(word, guessed\_letters):** This function takes the word to guess and a list of guessed letters as input. It displays the word with blanks for letters that have not been guessed yet and reveals the guessed letters.

**is\_word\_guessed(word, guessed\_letters):** This function checks if all the letters in the word have been guessed. It returns True if all letters in the word are in the list of guessed letters, and False otherwise.

**play\_hangman():** This is the main function that orchestrates the game. It welcomes the player, asks for their name, and prompts them to choose a category. Then, it selects a random word from that category using select\_word(). The player has 7 attempts to guess the word. After each guess, it checks if the guess is correct, updates the hangman display, and displays the word with guessed letters. If the word is guessed correctly within 7 attempts, it congratulates the player. If not, it reveals the correct word and offers to play again.

**Overall**, this code sets up the game environment, manages the game mechanics, and provides a user-friendly experience for playing Hangman.