بسمه تعالی

گزارش پروژه درس برنامه‌نویسی

عنوان پروژه: hangman

نگارنده: ......

# بازی کلاسیک Hangman که شامل بازیکنان می‌شود که حروف یک کلمه را به‌طور تصادفی با تعداد محدودی حدس می‌زنند.

# کتابخانه‌ها

در این پروژه از 2 کتابخانه استفاده شده است:

* + random: کتابخانه‌ای برای کار با اعداد تصادفی. از این کتابخانه برای تولید حروف و همچنین تصمیم برای انتخاب استفاده شده است.
  + string: ماژول رشته پایتون شامل ثابت ها، تابع ابزار و کلاس هایی برای دستکاری رشته است.

توضیح کدها :

# پاسخ به رویدادهای درون بازی

# از تابع قالب برای پر کردن فضاها استفاده کنید

"""

یک کلمه تصادفی را از بین کلمات موجود در فهرست کلمات انتخاب می کند

ارگ:

all\_words (list): فهرست کلمات موجود (رشته ها)

برمی گرداند:

یک کلمه از فهرست کلمات به صورت تصادفی

"""

# انتهای کد کمکی

# -----------------------------------

"""

لیستی از کلمات معتبر ایجاد کنید. کلمات رشته هایی از حروف کوچک هستند.

برمی گرداند:

لیستی از کلمات معتبر

"""

# TODO: کد خود را در اینجا پر کنید

# فهرست کلمات بارگیری را از file:words.txt چاپ می کند

# inFile: فایل

"""

فهرستی از کلمات معتبر را برمی گرداند. کلمات رشته هایی از حروف کوچک هستند.

بسته به اندازه فهرست کلمات، این تابع ممکن است

کمی طول بکشد تا تمام شود

"""

# قابل دسترسی از هر نقطه در برنامه

# TODO: خط زیر را یک بار لغو نظر کنید

# تابع load\_words() را پیاده سازی کرده اید

"""

guess: char، حرفی که کاربر حدس زده است

برمی گرداند: بولی، اگر حرف در کلمه باشد درست است، اگر حرف نباشد غلط است

"""

# TODO: کد خود را در اینجا پر کنید

"""

کلمه حدس زده فعلی را با زیرخط تعیین می کند

ارگ:

کلمه (str): کلمه ای که کاربر حدس می زند

حروف\_ حدس زده (لیست): کدام حروف تاکنون حدس زده شده است

برمی گرداند:

رشته، متشکل از حروف، زیرخط (\_) و فاصله هایی که نشان می دهد کدام حروف تا کنون حدس زده شده اند.

"""

# TODO: کد خود را در اینجا پر کنید

"""

حروفی که حدس زده نشده اند را مشخص کنید

ارگ:

letters\_guessed: فهرست (رشته ها) که کدام حروف حدس زده شده است

برمی گرداند:

رشته، متشکل از حروفی که هنوز حدس زده نشده اند.

"""

# TODO: کد خود را در اینجا پر کنید

"""

یک بازی تعاملی Hangman را اجرا می کند.

ارگ:

کلمه: رشته، کلمه حدس زدن.

"""

# لیست کلمات را در لیست کلمات متغیر بارگذاری کنید

# تا از هر جای برنامه قابل دسترسی باشد

# TODO: کد خود را در اینجا پر کنید

حداکثر # 11 حدس اعمال شد

''''

فقط حرف اول حدس در نظر گرفته می شود

''''

''''

اگر حروف صدادار اشتباه حدس زده شوند 2 حدس از

کل حدس ها کسر می شود

''''

#بررسی می کند که آیا کلمه کامل شده است یا خیر. در صورت تکمیل دستورات تبریک می گوییم و امتیاز را نشان می دهد

#اگر تلاش ها بیش از حد انجام شده باشد

وقتی حدس‌های کاربر تمام شد از کل برنامه خارج شوید

''''

تابعی برای چاپ تابلوی امتیاز

از سه گلزن اول

:برگشت:

چیزی را پس نمی دهد

''''

''''

برای پیدا کردن امتیاز بالاتر از فایل

:برگشت:

بازگشت list\_all

''''

''''

یک تابع برای به روز رسانی تابلوی امتیاز

:برگشت:

هیچ ارزشی را بر نمی گرداند

''''

اگر کاربر امتیاز خود را ذخیره کرد، روی فایل scores.txt بنویسید

# این باعث موفقیت امتیاز ذخیره شده در scores.txt می شود

#این تابع تصمیم گیری است که هر یک از کاربران بخواهند امتیاز نهایی خود را فقط با وارد کردن y یا n ذخیره کنند.

#برنامه فقط در دو حالت متوقف می شود

# 1. اگر کاربر y را تایپ کند و امتیاز ذخیره کند، تابع game\_decision\_c را فراخوانی می کند و از شرط توقف آن تابع پیروی می کند.

#2. اگر کاربر n را تایپ کند و امتیاز را ذخیره نکند، تابع game\_decision\_c را فراخوانی می کند و از شرط توقف آن تابع پیروی می کند.

# نام بازیکن را به عنوان ورودی در نام متغیر می گیرد

#ذخیره مقادیر امتیاز بازگشتی از تابع hangman(word) به متغیر final\_score

# و همچنین عملکرد hangman(word) را اجرا می کند

#دریافت تمام لیست امتیاز بالا از تابع امتیاز بالا

# اگر امتیاز نهایی بازیکن فعلی بیشتر از لیست امتیازات همه بازیکنان باشد

# سپس به بهترین وجه پرسنل جدید را تحریک می کند!

# و به کاربر دسترسی بدهید اگر بخواهد امتیاز خود را ذخیره کند یا در غیر این صورت دستور elif را انجام ندهد

اگر کاربر y را تایپ کند، # امتیاز را در تابلوی امتیاز ذخیره کنید

# فراخوانی تابع game\_decision\_controller\_main().

# فراخوانی برای انجام کار تابع game\_decision\_controller\_main() در صورتی که کاربر n را تایپ کند

اگر امتیاز نهایی بازیکن فعلی کمتر از لیست لیست امتیازات بازیکنان فعلی باشد

# یا امتیاز نهایی بازیکن فعلی برابر با لیست تمام امتیازات بازیکن است

# سپس به کاربر دسترسی می دهد که آیا می خواهد امتیاز خود را ذخیره کند یا خیر.

اگر کاربر y را تایپ کند، # امتیاز را در تابلوی امتیاز ذخیره کنید

# فراخوانی تابع game\_decision\_controller\_main().

# فراخوانی برای انجام کار تابع game\_decision\_controller\_main() در صورتی که کاربر n را تایپ کند

# -----------------------------------

#این کارکرد کنترل تصمیم کاربر است که می‌خواهد بازی کند یا تابلوی امتیاز را ببیند یا بازی را ترک کند.

#برنامه فقط در دو حالت متوقف می شود

# 1. اگر در هنگام درخواست ورودی های مرتبط، q را برای خروج تایپ کنند

# 2. اگر آنها از حدس ها فراتر رفتند و در بازی شکست خوردند

#نمره فراخوانی تابع اصلی. این تابع فرآیند کنترل امتیاز را کنترل می کند

عملکرد # فراخوانی تابلوی امتیاز () . این تابع لیست 3 گلزن برتر را نشان می دهد.

#نمره فراخوانی تابع اصلی. این تابع فرآیند کنترل امتیاز را کنترل می کند

عملکرد # calling scoreboard() اگر کاربر l را تایپ کند. این تابع لیست 3 نفر برتر را نشان می دهد.

پیام های # با تشکر از بازی Goodbye و خروج از برنامه اگر کاربر q را تایپ کند

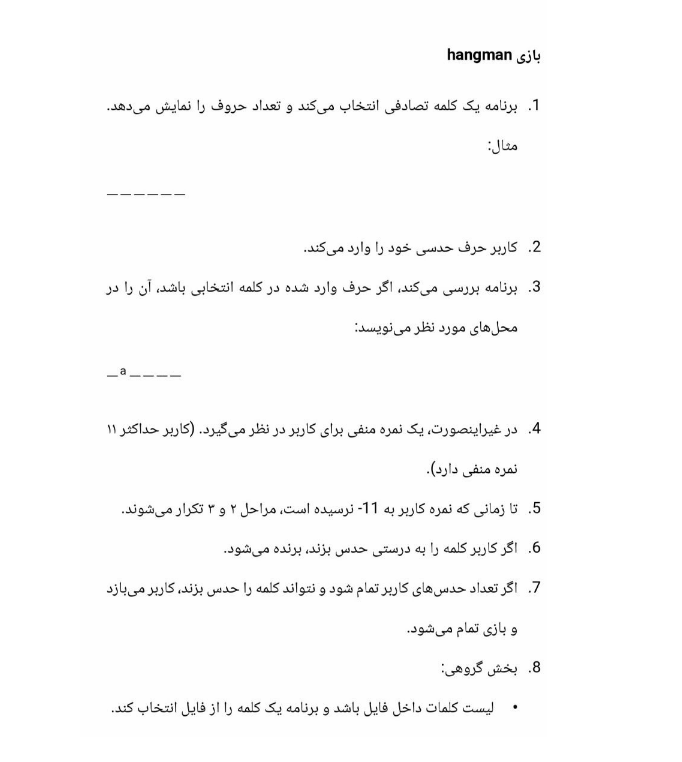
# -----------------------------------

# عملکرد درایور برای برنامه

# این توابع ابتدا در برنامه اجرا می شود.

#این پیام خوشامدگویی را نشان می دهد

* + روال برنامه



# نمونه خروجی برنامه

