

بازی تایپ

پروژه پایان ترم درس مبانی کامپیوتر و برنامه سازی
ترم ۴۰۰۱

در این پروژه از مفاهیمی که در طول ترم آموخته اید استفاده خواهید کرد تا یک بازی تمرین تایپ کردن تحت کنسول پیاده سازی کنید.



ساختار کلی

حتما تا حالا با بازی هایی که برای تمرین تایپ کردن ساخته شده اند، برخورد داشته اید. ساختار کلی این بازی ها به این شکل است که یک سری کلمه روی هدف های مختلف به سمت شما حرکت می کنند و باید قبل از رسیدن به مقصد کلمه نوشته شده روی هدف ها را تایپ کنید تا آنها نابود شوند و به مراحل بعد بروید. (مثل تصویر بالا) و رفته رفته سرعت حرکت این اهداف بیشتر و بیشتر می شود و به دلیل یک ویژگی ذاتی انسان که گریز از نابودی است، به طور ناخواسته و غیر مستقیم سرعت تایپ شما نیز بیشتر خواهد شد.



برای اینکه کار شما سخت نباشد، ساختار بازی را به شکل انتزاعی (abstract) از آن تغییر داده ایم که در ادامه به بررسی آن می‌پردازیم.

برای اینکه روند بازی را متوجه شوید ابتدا یک صف یک طرفه عمودی را در نظر بگیرید:



در واقع هر کدام از این مستطیل های سبزیابی رنگ کلماتی هستند که به سوی کاربر روانه می شوند تا به ترتیب آنها را تایپ کند. و هر مستطیل به محض تایپ شدن کلمه نوشته شده روی آن، از بین می رود و یک فضای خالی در صف اضافه می شود.

و اگر بخواهیم ساختار بالا را در ترمینال با زبان سی شبیه سازی کنیم به شکلی شبیه به ساختار زیر می رسم:

[illegible][illegible]



با یک دوره زمانی مشخص (قابل تغییر) یک کلمه از بالا به این صف افزوده می‌شود و همه کلمات به پایین شیف‌ت می‌خورند. اگر صف پر شود و کلمه ای تایپ نشده باشد، کاربر می‌بازد و بازی تمام می‌شود.

تایپ کردن و حذف کردن هر ۱۰ کلمه را یک موج (wave) از بازی می‌نامیم. بعد از هر موج از بازی دوره زمانی اضافه شدن و پایین آمدن کلمات کاهش پیدا می‌کند تا در انتها به یک ثانیه برسد و پس از آن ثابت می‌ماند.

همچنین با کمی دید انتزاعی تر، می‌توانیم همیشه هدف را پایین‌ترین کلمه (کلمه روی صف) در نظر بگیریم. به عبارتی همواره هدف کاربر بر کلمه روی صف lock شده است.

یعنی کاربر حتما باید در مرحله اول کلمه روی صف را تایپ کند. جلوتر می بینیم که برخی کلمات قبل از اینکه به روی صف برسند، اصلا مشخص نیستند. (با * مخدوش شده اند)

روش دریافت و اعمال ورودی نیز کاراکتر به کاراکتر است. یعنی اگر کاربر در حال تایپ کلمه‌ای بود و نصف آن را تایپ کرد و زمان آن دوره تمام شد و کلمه‌های خانه پایین تر آمدند، کاربر فقط باید نیمه باقیمانده از کلمه را تایپ کند تا آن کلمه حذف شود. از این رو لازم است او بداند تا چه مرحله‌ای از کلمه را تایپ کرده است! پس باید آن بخش از کلمه که تایپ شده است را به رنگ دیگری نمایش دهید.

در این مثال اگر کاربر در این حالت حروف دیگری تایپ کند مثل a b c همگی این حرف نادیده گرفته خواهند شد و نیازی به استفاده از دکمه Backspace نیست. و وقتی که حرف درست یعنی s را تایپ کند، این حرف نیز قلم می‌شود.

[illegible]



برای سادگی دریافت کاراکتر به کاراکتر از ورودی و بدون نیاز به فشردن دکمه اینتر، کد کمکی برای شما آماده شده است که در اختیارتان قرار می‌گیرد و در ادامه نحوه استفاده از آن را می‌بینیم.

برای چاپ رنگی در ترمینال نیز کفایت سؤالتان را در گوگل سرچ کنید تا راه های مختلف آن را ببینید. (:

انواع کلمات

انواع مختلفی از کلمات ممکن است در صف وجود داشته باشند که در ادامه به معرفی آنها و امتیازی که حذف کردن هریک از آنها به کاربر می‌دهد، می‌پردازیم:

۱. **کلمه معمولی:** این نوع کلمات، کلماتی معمولی هستند که از کاراکترهای حروف الفبا و اعداد انگلیسی ساخته شده اند و حداکثر ۱۰ کاراکتر طول دارند، حذف کردن این کلمات ۱ امتیاز به کاربر می‌دهد.

۲. **کلمه بلند:** این نوع کلمات، دارای طول بیشتری نسبت به کلمات معمولی دارند. یعنی طول آنها تا ۲۰ کاراکتر می‌تواند باشد. همچنین علاوه بر حروف الفبا، اعداد و کاراکتر _ (under-line) نیز جزء تشکیل دهنده آنها هستند، حذف کردن این کلمات ۲ امتیاز به کاربر می‌دهد.

۳. **کلمه سخت:** این نوع کلمات، دارای طولی مشابه کلمات معمولی و بلند هستند. اما دارای کاراکترهای سخت (!&%^\$@) هستند، حذف کردن این کلمات ۳ امتیاز به کاربر می‌دهد. اگر کاربر کلمه بلند سختی را در یکی از ۳ خانه باقیمانده به کف صف حذف کند، برای جایزه وارد حالت افتخاری بازی می‌شود که در ادامه توضیح داده شده است.

[illegible][illegible]

خب بقیه کلمات بمونن برای بخش امتیازی :)



حالت افتخاری بازی

اگر کاربر کار خفنی انجام دهد، یکبار وارد این حالت می شود تا امتیاز اضافی و راحت دریافت کند. ویژگی های این حالت به شرح زیر است:

- + وقتی وارد این حالت شویم، طول همه کلمات موجود در صف به ۳ کاهش پیدا می کند.
- + سپس به مدت ۲ برابر دوره زمانی آن موج، صف از حرکت می ایستد و کاربر فرصت دارد تا کلمات ساده را تایپ کرده و حذف کند.
- + بعد از این گذشت این مدت یا خالی شدن صف، بازی به حالت معمولی برمی گردد و کلمات با همان دوره قبلی وارد صف می شوند.

اجرای برنامه

با اجرای برنامه پروژه، کاربر سطح سختی بازی را از بین سه سطح آسان، متوسط و سخت انتخاب می کند.
دوره زمانی ورود کلمه جدید به صف به شرح زیر است:

سطح بازی	دوره زمانی موج اول (ثانیه)	درصد کاهش دوره زمانی
آسان	۱۰	۲۰
متوسط	۸	۳۰
سخت	۵	۴۰

(یعنی مثلاً برای سطح سخت، دوره زمانی موج ها به این ترتیب هستند: ۵ - ۳ - ۱٫۸ - ۱٫۰۸ - ۱) (هر وقت دوره زمانی به مقدار کمتر یا مساوی ۱ رسید، آن دوره را ۱ و آن موج را موج آخر بازی در نظر بگیرید)



پس از این، موج اول بازی شروع شده و کلمات به صف وارد می شوند. به موارد زیر دقت کنید:

- + دوره زمانی هر موج را با دقت میلی ثانیه حساب کنید.
- + کلمات باید از یک فایل متنی که در پوشه پروژه وجود دارد، خوانده شوند.
- + این فایل را خودتان به دلخواه ایجاد کنید، در هر سطر آن یک کلمه بنویسید و در برنامه خود از آن فایل بخوانید. (این کلمات را ۱۰ تا ۱۰ تا بخوانید و موج ها را بسازید)
- + کلمات مبهم در فایل تعریف نمی شوند. باید خودتان در برنامه کلمه اصلی را نگه داشته و آن را به صورت * به کاربر نشان دهید و موقعی که به روی صف رسید کلمه اصلی را نمایان کنید!

بعد از شروع بازی، دو حالت ممکن است رخ دهد:

۱. کاربر همه کلمات را به درستی و قبل از رسیدن کلمه روی صف به کف صف، تایپ کرده و حذف می کند. که در این صورت پس از اتمام هر موج، موج بعدی با دوره زمانی کوتاه تر آغاز می شود و تا جایی پیش می رود که دوره به یک ثانیه می رسد. اگر کاربر این مرحله را نیز با موفقیت گذراند، پیغامی مبنی بر اینکه به درجه خوبی از سرعت تایپ رسیده به همراه امتیازش به اون نشان داده می شود و بعد از ۵ ثانیه برنامه به اتمام می رسد.

۲. کاربر در تایپ کلمات به مشکل می خورد و کلمه روی صف به کف صف می رسد. در این صورت شماره موجی که در آن بوده و امتیازش به همراه سطح سختی به او نشان داده می شود و بعد از ۵ ثانیه برنامه به اتمام می رسد.



نکات پیاده سازی

در صورت عدم رعایت موارد زیر نمره این بخش را از دست خواهید داد:

- نامگذاری ها باید مطابق استاندارد باشد که در کارگاه ها آموخته اید و در تمام قسمت های کد رعایت شود.
- دندانه گذاری ها باید مطابق استاندارد باشد.
- تا حد امکان کد های تکراری را به توابع تبدیل کنید و از نوشتن کد های تکراری در پروژه خودداری کنید.
- برای توابع و متغیر ها اسامی با معنی انتخاب کنید.
- تمام پروژه باید در یک فایل و با زبان برنامه نویسی C پیاده سازی شود.
- همه کلمه ها را یکجا در برنامه load نکنید! چون حجم زیادی دارد و باعث اختلال در اجرای برنامه می شود. در شروع هر موج ۱۰ کلمه از فایل خوانده و استفاده کنید.
- اگر به مسئله ای برخوردید، ابتدا در سایت های مختلف به دنبال راه حل بگردید، و فقط در صورتی که جوابتان را نیافتید به حل تمرین تان مراجعه کنید. (پیشرفت و یادگیری در تلاش برای دیباگ کردن شکل می گیرد و دیباگ کردن کدتان به عهده شماست!)
- در نهایت باید یک فایل زیپ در کوئرا آپلود کنید. که حاوی سه فایل است. کد کمکی و کد پروژه به همراه فایل متنی که کلمات در آن ذخیره شده اند.
- در فایل متنی کلمات خود، انواع کلماتی را که پروژه شما پشتیبانی می کند (چه اصلی و چه امتیازی) قرار دهید. تا در هنگام ارائه دچار مشکل نشوید.
- ارائه و تحویل پروژه به صورت آنلاین، با وبکم و میکروفن و از طریق اسکایپ یا پلتفرم های مشابه می باشد که بعد از پایان ددلاین تاریخ آن مشخص خواهد شد.
- اجرا شدن برنامه و صرفا اضافه شدن و حذف کلمات، بخش زیادی از نمره پروژه را دارد در نتیجه توصیه می شود کارتان را با موارد ساده شروع کرده و سپس به اضافه کردن ویژگی های جدید به کدتان پردازید.
- کاربر باید در هر لحظه امتیازش را ببیند. (قسمتی از پایین یا بالای صفحه را به این بخش اختصاص دهید)



نحوه استفاده از کد کمکی

انجام دادن همزمان چند کار در برنامه های کامپیوتری از مباحثی است که در ترم های بعد با آن آشنا خواهید شد. در این پروژه ما نیاز داریم که همزمان ورودی را از کیبرد بخوانیم و باقی عملیات پردازش را انجام دهیم. برای اینکه درگیری پیچیدگی های این مسئله نشوید کدی کمکی برای شما آماده کرده ایم. که باید از آن استفاده کنید.

در ابتدای تابع main کار های تعیین سطح بازی و منو را انجام دهید و موقع شروع بازی تابع start_listening را صدا بزنید. (به همان شکلی که در کد هست)
پس از این دیگر نمی توانید از تابع scanf استفاده کنید. در عوض با فشردن هر کلید از کیبورد، تابع my_callback_on_key_arrival در کدتان به طور خودکار صدا زده می شود.
در این تابع عملیات پردازش تایپ و حذف کلمه را باید انجام دهید. یعنی مثلاً چک کنید که اگر کاراکتر وارد شده برابر با کاراکتر lock شده از کلمه روی صف بود، آن را به حالت تایپ شده تغییر دهید و منتظر کاراکتر بعدی شوید.
همچنین خط آخر قبل از return در تابع main را نیز پاک نکنید.

انواع کلمات در حالت امتیازی

۱- کلمات حفظی: این کلمات از ابتدای ورودی به صف آشکار می باشند و زمانی که به ابتدای صف می رسند کاراکتر های آن با کاراکتر * جایگزین می شود و کاربر باید این کلمات را حفظ کرده و آنها را تایپ کند. این کلمات در صورتی که توسط کاربر تا ورود دو کلمه به صف توسط کاربر تایپ نشوند باید آشکار شوند و در صورتی که کاربر موفق به تایپ کردن آنها تا قبل از آشکار شدن شود باید خود کلمه به همراه دو کلمه بالاتر آن حذف شود. همچنین حذف این دو کلمه باید به گونه باشد که کلمه بالایی این کلمه حفظی باید به گونه ای باشد که انکار کاربر آن را تایپ کرده و اگر کاراکتر خاص باشد باید عملیات آن انجام شود. (به عنوان مثال اگر کلمه بالایی کلمه حفظی کلمه ای امتیازی باشد باید به کاربر ۵ امتیاز اضافه شود.) دومین کلمه ای که حذف



می شود امتیازی را به کاربر اضافه نمی کند و عملیات آن نیز انجام نمی شود. (به عنوان مثال فرض کنید کلمات ابتدای صف به این ترتیب باشد: حفظی - امتیازی - فرصت. اگر کاربر به موقع این کلمه حفظی را تایپ کند علاوه بر امتیاز این کلمه امتیازی کلمه امتیازی را نیز می گیرد ولی فرصتی به کاربر اضافه نمی شود.) این کلمات ۳ امتیازی هستند. این کلمات بین دو علامت \$ قرار می گیرند.

۲- کلمات امتیازی: این کلمات در صورتی که در جایگاه سوم یا جلوتر (جایگاه اول یا دوم) در صف قرار می گیرند آشکار می شوند و حداقل شامل ۴ کاراکتر خاص (غیر از عدد و حروف)، ۴ حرف بزرگ و ۳ عدد خواهد بود. (تعداد حروف کوچک می تواند متفاوت باشد) این کلمات در صورتی که توسط کاربر تایپ شوند باید به جای یک امتیاز ۵ امتیاز به کاربر اضافه شود. این کلمات در زمان آشکار شدن بین دو علامت ^ قرار می گیرد و در صورتی که تا ورود ۲ کلمه به صف کاربر نتواند کلمه را تایپ کند کلمه عادی ای جایگزین آن می شود و تمام کاراکترهایی که کاربر تایپ کرده نیز پاک می شود.

۳- کلمات بد: این کلمات فقط شامل حداکثر ۴ حروف کوچک هستند و در زمانی که در جایگاه دوم صف یا جلوتر قرار می گیرند آشکار می شوند. این کلمات در زمان آشکار شدن بین دو کاراکتر # قرار می گیرند و در صورتی که در ابتدای صف قرار داشته باشند و تا ورود ۲ کلمه به صف توسط کاربر تایپ نشوند به صورت خودکار حذف و ۳ کلمه به طور یکجا وارد صف می شوند. این کلمات ۲ امتیازی هستند.

۴- کلمات فرصت: این کلمات که در بین دو کاراکتر @ و زمانی که در ابتدای صف قرار می گیرند ظاهر می شوند می بایست شامل حداقل ۳ کاراکتر خاص متفاوت و ۳ عدد متفاوت و یک حرف کوچک و یک حرف بزرگ باشند. در صورتی که کاربر بتواند این کلمات را قبل از اضافه شدن کلمه جدید به صف تایپ کند در صورتی که تعداد جان های باقی مانده کاربر کمتر از ۳ عدد باشد به تعداد جان های باقی مانده کاربر یک واحد اضافه می کند. در صورتی که کاربر نتواند این کلمه را در موعد مشخص تایپ کند کلمه فرصت با افزوده شدن کلمه جدید حذف می شود. این کلمات ۲ امتیازی هستند.



۵- کلمات حساس: این کلمات که بین دو کاراکتر! ظاهر می شوند تنهای ویژگی ای که دارند این است که اگر کاربریه حرف را اشتباه وارد کرد باید تمام چیزهایی که قبل از آن کاراکتر تایپ کرده پاک شود و از اول کلمه را بنویسد. این کلمات ۳ امتیازی زمانی که در ابتدای صف قرار می گیرند چاپ می شوند. همچنین در صورتی که کاربر موفق به تایپ این کلمات شود باید سرعت ورود ۳ کلمه جدید ۲۵ درصد کاهش یابد. (به عنوان مثال اگر فاصله زمانی بین کلمات ورودی ۱ ثانیه باشد این مقدار برای ۳ کلمه باید به ۱۰۵ ثانیه تغییر یابد.)

نکات کلمات امتیازی

- با وجود اینکه کلمات عادی نیز می توانند شامل کاراکترهای \$, #, ^, @, ! باشند اما نباید به گونه ای این کاراکترها در کلمه جای بگیرند که با کلمات خاص اشتباه گرفته شوند. به عنوان مثال اگر در فایل کلمه ای وجود داشت که اول و آخر آن علامت @ بود باید موقع ورود به صف این دو علامت از ابتدا و انتهای کلمه حذف شوند.
- در هیچ کدام از کلمات خاص کاربر نباید کاراکترهایی که در ابتدا و انتهای کلمه قرار می گیرند را تایپ کنند.
- در فایل اگر کلمه ای ۲ حرفی که هر دو حرفش یکی از حروف \$, #, ^, @, ! بود به جای آن کلمه باید کلمه خاصی متناسب با آن کاراکتر و با توجه به توضیحات بالا به صورت رندم تولید کنید و آن کلمه را به جای کلمه دو حرفی وارد صف کنید. (به عنوان مثال اگر در فایل کلمات کلمه ای به شکل \$\$ وجود داشت باید به جای آن کلمه ای حفظی به صورت رندم تولید و وارد صف کنید.)
- اگر در توضیحات کلمات خاص ساختار خاصی ذکر شده بود (مثل کلمات امتیازی) باید حتما موقع تولید آن کلمه ساختار لحاظ شود و اگر توضیح خاصی ذکر نشده بود باید کلمه ای با طول و کاراکترهای رندم تولید کنید.



- کلمات بخش امتیازی دارای نرخ جهش 0.05 هستند. یعنی اگر تعداد کلمات خاص (امتیازی) در فایل کمتر از ۲ بود هر کدام از کلمات عادی با احتمال ۵ درصد می توانند به صورت رندم یه یکی از کلمات خاص تبدیل شوند. تعداد کلمات خاص بعد از رخ دادن جهش ها نباید تعدادشان از ۲ بیشتر شود. (البته تعداد کلمات خاص درون فایل می تواند بیشتر از ۲ باشد و در این صورت نباید جهشی رخ دهد).
- پارامترهای تعریف شده در حالت امتیازی (مانند تعداد کاراکترهای خاص در کلمات امتیازی، تعداد فرصت های کاربر، امتیاز هر کلمه و ...) باید به سادگی در کد قابل تغییر باشد. (در صورت پیاده سازی هریکی از کلمات امتیازی این مورد ضرور است)
- در صورتی که کلمات فرصت را پیاده سازی می کنید حتما باید مورد دوم از موارد امتیازی را نیز پیاده کنید تا نمره این کلمات به شما تعلق بگیرد.
- تمام کلمات امتیازی ای که از ابتدای ورودی به صف آشکار نیستند باید تمام کاراکتر هایشان (به جز کاراکتر اول و آخر) با علامت * جایگزین شده و وارد صف شوند. همچنین تا آشکار شدن کلمه نباید نوع کلمه خاص مشخص شود. (به عنوان مثال کلمه \$hello\$ اگر هنگام ورود به صف آشکار نباشد باید تا زمان آشکار شدن به شکل ***** نمایش داده شود)



موارد امتیازی اضافه

- امکان تعریف کاربر وجود داشته باشد. کاربر در ابتدای اجرا نام خود را وارد کند و به بازی پردازد. نام هر کاربر به همراه امتیازش در فایل ذخیره شود و در اجرای بعدی هر کاربر با وارد کردن نامش، بالاترین امتیاز قبلی اش را ببیند.
- کاربر بیش از یک جان داشته باشد. یعنی در ابتدا ۳ جان به او داده شود و هر بار که کلمه روی صف به کف صف رسید، صف خالی شود و یکی از جان های او کم شود.
- اندازه صف (عمودی و افقی)، دوره زمانی اولیه و طول موج در ابتدا قابل تعریف باشد.
- رابط کاربری گرافیکی برای پروژه پیاده سازی کنید. (خارج از ترمینال)
- در انتهای بازی تمام کلمات صف که از ابتدا تا پایان بازی در ابتدای صف قرار گرفته اند به همراه وضعیت آنها (کاربر توانسته است آنها را تایپ کند یا خیر) در یک فایل ذخیره شوند و در ترمینال به کاربر در کنار امتیاز کاربر و حداکثر امتیاز کسب شده توسط کاربر نشان داده شود.
- امتیاز کاربر را در فایل ذخیره شود و در بار بعدی اجرا، او با بالاترین امتیازش مقایسه شود.
- تمام پارامترهای بخش عادی پروژه (مانند دوره زمانی موج اول در حالت های مختلف بازی) به سادگی قابل تغییر باشند یا از یک فایل متنی settings در کنار کد خوانده شوند.

موفق باشید ;)