Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

IIC1103 – Introducción a la Programación
2 - 2016

Enunciado Tarea 1

Recordatorio:

- **Fecha de entrega:** Jueves 15 de septiembre de 2016, a las 23:50 hrs.
- **Foro de consulta:** `https://goo.gl/SJSrKO`
- Este trabajo es **estrictamente personal**. Recuerda leer la Política de Integridad Académica del DCC disponible en `http://www.ing.uc.cl/ciencia-de-la-computacion/programas/licenciatura/politica-de-integridad-academica/`. Se usará un software **anti-plagio** para detectar similitud entre códigos.

¡Atención!

Ten en consideración que **no se recibirán entregas fuera del plazo**. Tampoco se responderá si te equivocas en entregar el archivo.
Es de tu responsabilidad ir subiendo entregas parciales de tu tarea. Se revisará la última versión que hayas subido al sitio web del curso.

Objetivo

En esta tarea, se espera que practiques los contenidos de variables, operadores, control de flujo utilizando condicionales (*if / elif / else*) y ciclos (`while`), módulos, strings y manejo de interacción con el usuario a través de entradas (inputs) y salidas (outputs). Para esto, deberás programar un juego de palabras entre dos jugadores. Las herramientas que puedes utilizar para realizar esta tarea son todas aquellas que hayan sido tratadas en clases hasta la fecha de la entrega.
Aclaración: Si estimas conveniente, para el desarrollo de esta tarea puedes utilizar los módulos vistos en clases: `math` y `random`.

El juego

En esta sección se explica el juego de esta tarea, similar al juego Text Twist 2 (ver `http://www.juegos.com/juego/texttwist-2`) pero con algunas modificaciones. A continuación se explica los elementos y reglas del juego:

Jugadores

Este juego cuenta con dos jugadores: Jugador 1 y Jugador 2. Ambos jugarán en turnos que son intercalados y competirán por lograr el máximo puntaje.

Palabras y pozo de letras

En cada turno, cada jugador intentará *construir* una palabra utilizando un conjunto de letras, el que se llama *pozo de letras*, que se muestra en pantalla. Las palabras que pueden construir los jugadores tienen largo variable: desde usar 3 letras hasta el total de letras entregadas que contenga el pozo. Considerando lo anterior, las palabras que logren construir los jugadores otorgarán algún puntaje, el que se considerará para obtener un puntaje final por jugador.

El *pozo de letras* se genera al principio del juego y se muestra a ambos jugadores. Las letras en el pozo se pueden repetir. Los jugadores pueden decidir desordenar el pozo, pero las letras que lo componen no varían. Al comienzo del juego se debe decidir la dificultad de este, escogiendo el número de letras en el pozo. Este número puede ser 5, 6 o 7.

Por último, al comienzo del juego, ambos jugadores parten con 0 puntos cada uno y con ninguna palabra creada.

Desarrollo del juego

Al comienzo del juego se deberá decidir el número de letras que se pondrán en el pozo. Este número puede ser 5, 6 o 7. Luego se muestra a los jugadores el pozo que se genera (con el mecanismo explicado más adelante), las acciones posibles para el turno de cada jugador son las siguientes:

1. **Ingresar palabra:** el jugador podrá escribir e ingresar una palabra utilizando un pozo de letras. La palabra será válida si ésta existe en un archivo anexo, no ha sido utilizada anteriormente por el jugador o su contrincante y puede construirse con las letras del pozo de letras. El jugador obtiene el puntaje asociado a dicha palabra y además esta queda inhabilitada para ser utilizada en turnos siguientes (tanto por el mismo jugador como por su contrincante). Luego, se pasa al turno del otro jugador.
   En caso de escribir por primera vez una palabra inválida, el jugador tendrá dos intentos más para escribir una palabra válida. Sin embargo, si se ha escrito una palabra inválida, el jugador podrá realizar cualquiera de las otras acciones, pero se mantiene un registro de la cantidad de intentos que le quedan.
   Si luego de tres intentos no logra escribir una palabra válida, se le descuenta 2 puntos y se pasa al turno del otro jugador.
2. **Reordenar el *pozo de letras*:** en caso que el jugador tenga dificultades en construir una palabra, puede pedir que se reorganicen de otra forma las letras del pozo de letras. Luego de esto, el usuario puede ingresar una palabra, volver a reordenar el *pozo de letras*, o pasar su turno.
   Esta opción no descuenta intentos al usuario. Por ejemplo, si el usuario lleva un intento fallido, puede escoger reordenar las letras y sigue teniendo 2 intentos más. Sin embargo, durante un turno un jugador puede reordenar hasta 3 veces. Si el jugador quiere reordenar por una cuarta vez, su turno se acaba sin penalización de puntaje.
3. **Pasar:** el jugador puede no realizar ninguna acción, pero seguir participando en el juego. Si ambos jugadores pasan dos veces seguidas, el juego termina y gana el que tiene mayor puntaje
4. **Salir del juego:** si un jugador decide salirse del juego, el juego termina, dejando como vencedor al otro jugador.

A continuación se explica la distribución de puntaje para una palabra válida considerando un pozo de n letras:

- Palabra de largo m con $m < n$: otorga $2 * m$ puntos
- Palabra de largo n: otorga $5 * n$ puntos

Fin del juego

El juego puede finalizar por dos motivos:

- Ambos jugadores han pasado dos veces seguidas.
- Un jugador escoge salir del juego.

Tu programa

Tu programa deberá permitir jugar el juego descrito en la sección anterior. Específicamente se pide que el programa permita realizar las siguientes acciones:

1. Dar una bienvenida a los usuarios, explicando las reglas del juego.
2. Iniciar el juego pidiendo los nombres del jugador 1 y jugador 2.
3. Solicitar la cantidad de letras del pozo de letras: 5, 6 o 7.

4. Mostrar el pozo de letras con la cantidad de letras escogidas por el ususario. Para esto, se debe obtener una palabra con dicha cantidad de letras y luego desordenarla.

5. Permitir que los dos jugadores jueguen en turnos intercalados. **Siempre** parte jugando el jugador 1.

6. Permitir al jugador escoger su acción (ingresar palabra, reordenar *pozo de letras*, pasar o salir). Debes tener en consideración lo siguiente:

   - Se debe mostrar (en la pantalla) el pozo de letras que deben ser utilizadas para construir las palabras además de la cantidad de intentos restantes para el jugador correspondiente.
   - **Ingresar palabra:** Debes verificar que la palabra escrita por el usuario sea válida, es decir, que exista, que pueda ser formada con letras del *pozo de letras* y que además no haya sido utilizada en un turno anterior. En caso de ingresar una palabra inválida 3 veces en un mismo turno, entonces se le debe descontar puntaje y pasar al turno del otro jugador. Por otro lado, si es válida, debes calcular el puntaje siguiendo las reglas descritas en la sección anterior y pasar el turno al siguiente jugador.
     Adems se debe dar información sobre la *validez* de la palabra: si la palabra es válida darle un mensaje al jugador diciendo esto. En caso de no ser válida, debes darle la razón: palabra no existe, palabra no puede construirse a partir de las letras del *pozo de letras* o la palabra ya ha sido utilizada anteriormente.
     Por último, tu programa no debe ser *case-sensitive*, es decir, si el usuario ingresa `'HoLa'` es equivalente a ingresar `'hola'`, `'HOLA'`, `'HOla`, etc.
   - **Reordenar las letras:** Cuando el usuario escoge esta opción, se actualiza el orden de las letras del pozo de letras. Este orden se mantiene hasta que algún jugador escoja nuevamente reordenar las letras.
   - **Pasar:** El jugador termina su turno, partiendo el turno del otro jugador. Si los jugadores pasan dos veces seguidas en sus respectivos turnos, el juego debe terminar.

7. Verificar si hay un ganador al finalizar el juego. Se debe mencionar quién es el ganador y el puntaje de cada jugador.

Módulo

Para poder realizar tu programa, tendrás a disposición un módulo llamado `mezclador`. Adicionalmente, debes descargar los dos archivos que contienen las palabras válidas: `palabras_validas.txt` y `palabras.txt`. Los tres archivos: el módulo `mezclador.py` y los dos archivos de palabras deben estar en la misma carpeta que tu archivo `.py` de tu tarea. A continuación se describen las funciones que tiene el módulo:

- `existe_palabra(palabra)`: retorna `True` si el string `palabra` existe, es decir, está en el archivo `palabras_validas.txt`. Esta función retorna `False` en caso contrario.
- `obtener_palabra(cant_letras)`: retorna un `string` de largo `cant_letras`. Este `string` es una palabra que existe y corresponde al pozo de letras que se debe generar al comienzo del juego. Una vez que obtengas la palabra, debes desordenarla pues esta función entrega una palabra válida. Solo se genera un pozo de letras para cada juego.
- `agregar_palabra(palabra)`: agrega el string `palabra` al registro de palabras usadas por los jugadores.
- `palabra_usada(palabra)`: retorna `True` si la palabra ha sido usada anteriormente y se encuentra en el registro de palabras usadas por los jugadores.
- `limpiar_lista()`: vacía el registro de palabras utilizadas.
- `reordenar_palabra(palabra)`: recibe un string `palabra` y retorna una palabra con las mismas letras, pero ordenadas de manera aleatoria.
- `palabras.txt`: archivo que contiene las palabras que puede devolver la función `obtener_palabra(cant_letras)`. Las palabras están en mayúscula.
- `palabras_validas.txt`: archivo que contiene un listado de palabras validas para las palabras que se encuentran en `palabras.txt`. Las palabras están en mayúscula.

Nota: No debes preocuparte de leer los archivos `palabras.txt` y `palabras_validas.txt`, de esto se va a encargar el modulo `mezclador`. Estos solo son para que conozcas las palabras que pueden aparecer en el juego y las palabras válidas, para que puedas probar el correcto funcionamiento de tu programa.

Entrega

Debes guardar tu tarea en un archivo con el formato `tarea01_rut.py`, donde debes reemplazar `rut` con tu RUT. Por ejemplo si tu rut es 12.345.678-9 el nombre de la tarea debería ser `tarea01_123456789.py`. **Recuerda incluir solo el archivo .py de tu tarea, no los archivos ni módulo entregado para realizar esta tarea.**
La entrega se realiza en el buzón electrónico, disponible en la página web del curso: `https://puc.classter.net` hasta el Jueves 15 de septiembre de 2016, a las 23:50 hrs.

Ejemplo

A continuación se muestra una ronda donde el jugador 1 y el jugador 2 juegan en sus respectivos turnos. En este caso el pozo de letras corresponde a {R,T,E,S,N,C,E}.

- Se muestran las reglas del juego y además se pide el nombre de los jugadores y cantidad de letras en el pozo de letras

```
----------------------------------------------------------------------
BIENVENIDOS A TEXT TWIST 3
Las reglas del juego son las siguientes:
1. Cada jugador en su turno debe intentar escribir una palabra valida.
2. Una palabra valida es aquella que:
    - Existe
    - No se haya utilizado anteriormente en algun turno
    - Se pueda construir a partir de tres o mas letras del pozo de letras
3. En su turno, cada jugador tiene 3 oportunidades para escribir una palabra valida.
4. Si ambos jugadores pasan 2 veces seguidas, se acaba el juego

>>> Escribe el nombre del jugador 1: John
>>> Escribe el nombre del jugador 2: Jane
>>> Cantidad de letras a utilizar (5, 6 o 7): 7
```

- Primer turno de los jugadores. El jugador 1 obtiene 8 puntos por ingresar una palabra válida (`tres`). El jugador 2 no puede ingresar la misma palabra, por lo que recibe un mensaje de error señalando esto. Finalmente puede ingresar una palabra válida (`TREN`), por lo que obtiene 8 puntos.

```
----------------------------------------------------------------------
Turno del jugador 1: John
R  T  E  S  N  C  E
Acciones:
(1) Ingresar palabra
(2) Reordenar las letras
(3) Pasar
(4) Terminar el juego
>>> Que quieres hacer? 1

>>> Que palabra quiere ingresar? tres
>>> Palabra valida, gana 8 puntos! Se acaba el turno del jugador 1
----------------------------------------------------------------------
Turno del jugador 2: Jane
R  T  E  S  N  C  E
Acciones:
(1) Ingresar palabra
(2) Reordenar las letras
(3) Pasar
(4) Terminar el juego
>>> Que quieres hacer? 1
```

```
>>> Que palabra quiere ingresar? tres
>>> Palabra no valida: la palabra ya ha sido utilizada en turnos anteriores.
Pierdes una oportunidad, queda(n) 2 oportunidad(es)
--------------------------------------------------------------------------
Turno del jugador 2: Jane
R  T  E  S  N  C  E
Acciones:
(1) Ingresar palabra
(2) Reordenar las letras
(3) Pasar
(4) Terminar el juego
>>> Que quieres hacer? 1

>>> Que palabra quiere ingresar? TREN
>>> Palabra valida, gana 8 puntos! Se acaba el turno del jugador 2
```

- Jugador 1 intenta 3 veces ingresar una palabra no válida. El programa le muestra la razón y actualiza la cantidad de oportunidades que quedan. Por otro lado, el Jugador 2 pasa su turno.

```
--------------------------------------------------------------------------
Turno del jugador 1: John
R  T  E  S  N  C  E
Acciones:
(1) Ingresar palabra
(2) Reordenar las letras
(3) Pasar
(4) Terminar el juego
>>> Que quieres hacer? 1

>>> Que palabra quiere ingresar? tres
>>> Palabra no valida: la palabra ya ha sido utilizada en turnos anteriores.
Pierdes una oportunidad, queda(n) 2 oportunidad(es)
--------------------------------------------------------------------------
Turno del jugador 1: John
R  T  E  S  N  C  E
Acciones:
(1) Ingresar palabra
(2) Reordenar las letras
(3) Pasar
(4) Terminar el juego
>>> Que quieres hacer? 1

>>> Que palabra quiere ingresar? ezn
>>> Palabra no valida: la palabra no existe.
Pierdes una oportunidad, queda(n) 1 oportunidad(es)
--------------------------------------------------------------------------
Turno del jugador 1: John
R  T  E  S  N  C  E
Acciones:
(1) Ingresar palabra
(2) Reordenar las letras
(3) Pasar
(4) Terminar el juego
>>> Que quieres hacer? 2

>>> Letras reordenadas!
--------------------------------------------------------------------------
```

```
Turno del jugador 1: John
E  R  E  C  N  S  T
Acciones:
(1) Ingresar palabra
(2) Reordenar las letras
(3) Pasar
(4) Terminar el juego
>>> Que quieres hacer? 1

>>> Que palabra quiere ingresar? casa
>>> Palabra no valida: la palabra no puede formarse con el pozo de letras.
Pierdes una oportunidad, queda(n) 0 oportunidad(es)

>>> Se pasa al turno del jugador 2, con un descuento de 2 puntos
--------------------------------------------------------------------------
Turno del jugador 2: Jane
E  R  E  C  N  S  T
Acciones:
(1) Ingresar palabra
(2) Reordenar las letras
(3) Pasar
(4) Terminar el juego
>>> Que quieres hacer? 3

>>> Se pasa al turno del jugador 1
```

- Si en dos turnos consecutivos los jugadores pasan, entonces se acaba el juego y se muestran los ganadores.

```
--------------------------------------------------------------------------
Turno del jugador 1: John
E  R  E  C  N  S  T
Acciones:
(1) Ingresar palabra
(2) Reordenar las letras
(3) Pasar
(4) Terminar el juego
>>> Que quieres hacer? 3

>>> Se pasa al turno del jugador 2
--------------------------------------------------------------------------
Turno del jugador 2: Jane
E  R  E  C  N  S  T
Acciones:
(1) Ingresar palabra
(2) Reordenar las letras
(3) Pasar
(4) Terminar el juego
>>> Que quieres hacer? 3

>>> Se pasa al turno del jugador 1
--------------------------------------------------------------------------
Turno del jugador 1: John
E  R  E  C  N  S  T
Acciones:
(1) Ingresar palabra
(2) Reordenar las letras
(3) Pasar
```

```
(4) Terminar el juego
>>> Que quieres hacer? 3

>>> Se acaba el juego!
>>> John (jugador 1) tiene 6 puntos
>>> Jane (jugador 2) tiene 8 puntos
>>> El ganador es: Jane
```