



IIC2115 – Programación como Herramienta para la Ingeniería (I/2019)

## Laboratorio 2 - Programación Orientada a Objetos

### Objetivos

- Aplicar los contenidos de Programación orientada a objetos, a través de la realización de un juego de tablero con interfaz gráfica.

### Entrega

- **Lenguaje a utilizar:** Python 3.6
- **Lugar:** repositorio privado en GitHub. Recuerde incluir todo en una carpeta de nombre **L02**.
- **Entrega parcial:** lunes 25 de marzo a las 16:50 hrs.
- **Entrega final:** jueves 28 de Marzo a las 23:59 hrs.
- **Formato de entrega:** archivo python notebook (**.ipynb**) con la solución y archivos gráficos (**.png**), ubicado en la carpeta **L02**. Suba además, en la misma carpeta, un archivo **README.md** con las instrucciones para ejecutar su tarea. No se debe subir ningún otro archivo a la carpeta. Utilice múltiples celdas de texto y código para facilitar la revisión de su tarea.
- **Descuentos:** se descontará 0.5 pts. por cada hora de atraso y fracción en la entrega final. Tareas que no cumplan el formato de entrega tendrán un descuento de 0.5 pts.
- **Entregas parciales subidas fuera de plazo no serán consideradas.**
- **Tareas con errores de sintaxis y/o que generen excepciones serán calificadas con nota 1.0.**
- Las discusiones en las *issues* del Syllabus en GitHub son parte de este enunciado.

# Introducción

## El juego perdido

En una excavación arqueológica se encontró una misteriosa caja de madera. Dentro de ella, se hallaron unos pergaminos con las instrucciones de un juego antiguo. Lamentablemente, el paso del tiempo deterioró partes de los instructivos.

Pese a esto, una empresa de tecnología ha decidido reconstruirlo en forma de videojuego. Para ello, se ha puesto en contacto contigo, un ingeniero sumamente creativo y hábil para que trabajes en ello. La empresa esta convencida que harás una obra de arte.

## Información de los pergaminos

Los pergaminos que describen el juego fueron estudiados cuidadosamente con el fin de obtener la mayor cantidad de información posible. A continuación se describe lo que ha podido ser descifrado.

## El tablero

Los pergaminos revelaron que el juego se desenvuelve en un tablero cuadrado tipo ajedrez de  $10 \times 10$  celdas. En el se mueven los personajes del juego. La primera celda de la esquina inferior izquierda corresponde a la celda  $(x=1, y=1)$ .

## Bandos

El tipo de piezas descubiertas indica que en el juego participan dos bandos, enemigos entre si. Dado que no se tienen todos los detalles de ellos, por ahora se les decidió nombrar bando **X** y bando **Y**. Aquí aparece tu primera misión. Deberás definir el nombre de cada bando y qué personajes ocupa cada uno, con la consideración de que sean enemigos naturales. Por ejemplo, perros y gatos, ángeles y demonios, ~~ayudantes~~ ~~y alumnos~~, etc. Una vez definidos, deberás crea o descargar los *sprites* necesarios para darles vida en el juego. Si no sabes que son los *sprites*, puedes revisar el siguiente link: [https://es.wikipedia.org/wiki/Sprite\\_\(videojuegos\)](https://es.wikipedia.org/wiki/Sprite_(videojuegos)).

## Personajes identificados

Como se dijo, lamentablemente el significado completo de las piezas no ha podido ser descubierto, pero se logró observar que en los bandos existen piezas de tres categorías: **atacantes**, **defensores** y **jefes**. Además de esto, se sabe que dentro de las categorías atacantes y defensores existen jerarquías de especialización.

Para desarrollar los personajes, considera que todas las piezas poseen una **habilidad** y **movilidad** diferente, que depende de su rol dentro del juego. Esto deberá implementarse con métodos **especializados** `usar_habilidad()` y `moverse()` para cada personaje.

A continuación, se definen las habilidades y movimientos de los personajes que han logrado ser identificados.

#### a) Habilidades

Las siguientes habilidades se ejecutan al usar el método `usar_habilidad()` de los personajes. Recuerda que debes nombrar estos roles como tú quieras.

- **atacante 1:** puede atacar a una pieza enemiga sólo si se encuentra a una celda de distancia en diagonal, o a dos celdas de distancia en línea recta. La pieza atacada es destruida y retirada del tablero.
- **atacante 2:** puede transformar a un atacante enemigo en piedra durante 2 turnos, lo que evita que este último pueda moverse o que pueda ser destruido. Esta habilidad sólo se puede utilizar si el atacante enemigo está a una celda de distancia, en cualquier dirección.
- **defensa 1:** puede transformarse en piedra por un turno (correspondiente al turno del oponente), lo que evita que sea destruido si es atacado. Esta habilidad no puede utilizarse en dos turnos seguidos del mismo jugador.
- **defensa 2:** puede hacer retroceder a un enemigo en línea recta, hasta que este llegue al borde del tablero, o se encuentre con otra pieza. Esta habilidad sólo se puede utilizar si el atacante enemigo está al frente, a una celda de distancia.
- **jefe:** tiene todas las habilidades de los defensas y atacantes. La misma habilidad no puede utilizarse en dos turnos seguidos del mismo jugador.

#### b) Movimientos

Los movimientos que pueden hacer las distintas piezas son los siguientes:

- **atacantes y defensores:** pueden moverse una celda por turno en cualquier dirección, o dos celdas por turno en línea recta (sin considerar las diagonales).
- **jefes:** pueden moverse una celda por turno en todas direcciones.

## Dinámica del juego

El juego debe ser jugado de manera automática por el computador, **el usuario no participa del juego de ninguna manera**. No es necesario que el sistema muestre inteligencia o estrategia, basta que realice movimientos aleatorios, pero que cumplan con las reglas.

Cada equipo está conformado por 5 piezas: un jefe, dos atacantes y dos defensas. El juego se desenvuelve en turnos intercalados entre ambos jugadores. Un turno esta compuesto por un movimiento, y opcionalmente seguido de una habilidad, para cada una de las piezas del jugador. El juego acaba si todas las piezas del enemigo son eliminadas.

La posición inicial de las piezas queda a criterio de ustedes. Además, en cada turno deberás usar una función para mostrar texto (descrita más abajo), para indicar lo que va acontecer en el turno.

## Librería game

Tendrán a disposición una librería llamada game (*game.py*). Para que esta funcione, deberán realizar previamente la instalación del módulo **arcade**, utilizando el siguiente comando:

```
pip3 install arcade (si el anterior comando no funciona, intentar con pip install arcade o con
python3 -m pip install arcade)
```

Una vez instalado el módulo, la librería *game.py* va a funcionar correctamente. Lo primero que debe hacer es instanciar la clase del juego y luego inicializar una partida con el comando **setup()**. En este caso sería:

```
miJuego = game.LostGame()
miJuego.setup()
```

Luego deberá crear sus personajes con el comando que se describe a continuación.

```
miJuego.poner_piece(nombreArchivo, celda_x, celda_y, escala)
```

- **nombreArchivo** corresponde al nombre del *sprite* en formato **png** (que debe estar en la carpeta *sprites*), por ejemplo *personaje.png*.
- **celda\_x** corresponde a la posición de la celda en el eje x donde será creada la pieza.
- **celda\_y** corresponde a la posición de la celda en el eje y donde será creada la pieza.

- **escala** corresponde a un multiplicador al tamaño del *sprite* (archivo .png) para que puedas ajustarlo al juego

Una vez definidos tus personajes, deberás realizar todas las jugadas posibles. Para ello debieras tener un *loop* como el que se describe a continuación, y utilizar los métodos `mover_pieza()`, `quitar_pieza()` y `mostrar_texto()` que se encargan de mover, quitar y mostrar información en forma de texto, respectivamente.

```
while condition:
    #logica
    miJuego.mover_pieza(nPieza, celda_x, celda_y)
    miJuego.quitar_pieza(nPieza)
    miJuego.mostrar_texto(texto)
```

Donde:

- **nPieza** corresponde al identificador de la pieza que es el orden en que fue creada tal pieza con `poner_pieza()`, comenzando en **0**. Es decir, si quieres mover o quitar la primera pieza creada,  $nPieza = 0$ . Las piezas nunca cambian de identificados, incluso si otras piezas fueron quitadas.
- **celda\_x** corresponde a la posición de la celda en el eje x donde quiere desplazarse.
- **celda\_y** corresponde a la posición de la celda en el eje y donde quiere desplazarse.
- **texto** Muestra texto en pantalla para indicar qué ocurre en el juego.

Una vez que termine de ejecutar todas las instrucciones deberá utilizar el comando `miJuego.fin_del_juego()`. Finalmente el comando `miJuego.run()` iniciará la visualización.

```
miJuego.run()
```

## Bonus

Además del lo indicado anteriormente, la compañía se comprometió a entregar un succulento bonus de 1.0 punto, por cada uno de los siguientes elementos:

- **Personajes no identificados:** Además de los personajes anteriormente mencionados, existe un atacante y un defensa cuyos movimientos y habilidades no han logrado ser identificados. Deberás crear los sprites, habilidades y movimientos para ellos de modo tal que sean razonables respecto a las demás

piezas. Es decir, las piezas que diseños no deben ser inmortales o que permitan ganar el juego en un turno.

- **Comportamiento inteligente:** Deberás proveer a los personajes de un comportamiento razonablemente inteligente, que permita analizar el entorno y tomar una decisión en base al estado actual de este.

## Corrección

Para la corrección de este laboratorio, se revisarán las clases definidas, la relación entre estas y los métodos implementados, con énfasis especial en la especialización de su comportamiento mediante polimorfismo. Además, se evaluará el correcto comportamiento del juego y de las funcionalidades entregadas por la librería `game`.

### Avance parcial (final del la clase)

- Implementación completa de al menos un personaje, que permita **ver** su comportamiento en el tablero de juego. No es necesario que sea en el contexto de una partida, basta que se mueva y ejecute su habilidad.
- En caso de usar **colab**, cambiar el requerimiento anterior por la implementación del modelo de clases completo para los personajes y tablero. No es necesario que todos los métodos estén implementados, pero sí que estén definidos.

### Entrega final

- Modelo de clases completo del juego, incluyendo personajes, tablero y dinámica del juego. Se evaluará la calidad y completitud del modelo desde el punto de vista de la orientación a objetos, con énfasis en polimorfismo. **(3 ptos.)**
- Presentación gráfica de los personajes, sus movimientos y sus habilidades. **(1.5 ptos.)**
- Dinámica del juego correcta, respetando las reglas. **(1.5 ptos.)**

## Política de Integridad Académica

*“Como miembro de la comunidad de la Pontificia Universidad Católica de Chile me comprometo a respetar los principios y normativas que la rigen. Asimismo, prometo actuar con rectitud y honestidad en las relaciones con los*

*demás integrantes de la comunidad y en la realización de todo trabajo, particularmente en aquellas actividades vinculadas a la docencia, el aprendizaje y la creación, difusión y transferencia del conocimiento. Además, velaré por la integridad de las personas y cuidaré los bienes de la Universidad.”*

En particular, se espera que mantengan altos estándares de honestidad académica. Cualquier acto deshonesto o fraude académico está prohibido; los alumnos que incurran en este tipo de acciones se exponen a un procedimiento sumario. Ejemplos de actos deshonestos son la copia, el uso de material o equipos no permitidos en las evaluaciones, el plagio, o la falsificación de identidad, entre otros. Específicamente, para los cursos del Departamento de Ciencia de la Computación, rige obligatoriamente la siguiente política de integridad académica en relación a copia y plagio: Todo trabajo presentado por un alumno (grupo) para los efectos de la evaluación de un curso debe ser hecho individualmente por el alumno (grupo), sin apoyo en material de terceros. Si un alumno (grupo) copia un trabajo, se le calificará con nota 1.0 en dicha evaluación y dependiendo de la gravedad de sus acciones podrá tener un 1.0 en todo ese ítem de evaluaciones o un 1.1 en el curso. Además, los antecedentes serán enviados a la Dirección de Docencia de la Escuela de Ingeniería para evaluar posteriores sanciones en conjunto con la Universidad, las que pueden incluir un procedimiento sumario. Por “copia” o “plagio” se entiende incluir en el trabajo presentado como propio, partes desarrolladas por otra persona. Está permitido usar material disponible públicamente, por ejemplo, libros o contenidos tomados de Internet, siempre y cuando se incluya la cita correspondiente.