



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC2233 Programación Avanzada (2018-2)

Tarea 02

Entrega

- Diagrama de clases
 - **Entregable preliminar:** 29 de septiembre de 2018, 23:59.
 - **Lugar:** [Formulario de Google](#)
 - **Entregable final:** 7 de octubre de 2018, 23:59.
 - **Lugar:** GitHub — Carpeta: Tareas/T02/
- Tarea
 - **Fecha y hora:** 7 de octubre de 2018, 23:59.
 - **Lugar:** GitHub — Carpeta: Tareas/T02/
- README.md
 - **Fecha y hora:** 8 de octubre de 2018, 23:59.
 - **Lugar:** GitHub — Carpeta: Tareas/T02/

Objetivos

- Desarrollar algoritmos para la resolución de problemas complejos.
- Aplicar conceptos de programación orientada a objetos (OOP) para modelar y resolver un problema.
- Utilizar correctamente los *properties* cuando corresponda.
- Comunicar diseños orientados a objetos a través de documentación externa.
- Resolver un problema a través de una simulación de tipo *tick-by-tick*.
- Resolver un problema altamente parametrizado, para ofrecer flexibilidad en la creación y exploración de distintos escenarios.
- Definir, utilizar y documentar supuestos que no contradigan el enunciado, para manejar casos que no estén especificados de forma exhaustiva.
- Familiarizarse con el paquete de interfaz gráfica PyQt5.

Índice

1. Introducción	3
1.1. Distribuciones de probabilidad	3
2. <i>DCCasino</i>	3
3. Entidades	4
3.1. Clientes	4
3.1.1. Características	4
3.1.2. Personalidades	4
3.1.3. Decisiones	5
3.2. Personal del casino	6
3.2.1. Características	6
3.2.2. Tipos de personal	6
3.3. Juegos	7
3.3.1. Tipos de juegos	7
3.4. Instalaciones	8
3.4.1. Características	8
3.4.2. Tipos de instalaciones	8
4. Actividades	9
5. Interfaz gráfica	9
5.1. Clase <code>Entity</code>	10
5.1.1. Clase <code>Human</code>	10
5.1.2. Otras entidades	11
5.1.3. <i>Debugging</i>	12
5.2. Mapa	12
6. Simulación	13
6.1. Estadísticas	13
6.2. Parámetros	13
7. Diagrama de clases	14
8. Notas	14
9. Restricciones y alcances	14

1. Introducción

Lamentablemente, el malvado Dr. H⁴ consiguió lanzar por los aires los correos electrónicos del DCC. Es por ello que el departamento decide cambiar de rubro, por lo que le piden al conocido líder de la mafia *Enzini Tini Tamburini* que instaure un casino. Sin embargo, él no conoce verdaderamente el negocio; por ende, no se sabe qué tan viable sería realizar este proyecto. Es por esto que te pide a ti, maestro de la modelación OOP, que **simules lo que sucede cada segundo en este casino**, y saques posteriores estadísticas de lo sucedido. Así, él podrá ajustar los distintos parámetros asociados y obtener resultados convenientes. Por otro lado, a él no le interesan sólo los números, sino que además quiere **ver** cómo se va desarrollando el casino en cada momento del día. Debido a esto, te pide conectar tu programa a una interfaz gráfica proporcionada por el mismo *Tini*.

1.1. Distribuciones de probabilidad

Para prevenir que el malvado Dr. H⁴ sea capaz de predecir el comportamiento del *DCCasino* y dejarlo en la ruina, *Tini* te ha pedido que generes números usando distribuciones de probabilidad. Muy coloquialmente, una distribución de probabilidad asigna a un número una cierta probabilidad de salir elegido. Por ejemplo, en la distribución uniforme, todos los números (dentro de cierto intervalo) tienen exactamente la misma probabilidad de ser elegidos. Puedes ver más explicación respecto a esto en este [notebook](#).

En Python, para generar un número a partir de una distribución, podemos utilizar el conveniente módulo *built-in* llamado [random](#). A continuación, te presentamos las distribuciones más usadas.

- Distribución uniforme entre 0 y 1: `random()`
- Distribución uniforme entre a y b : `uniform(a, b)`
- Distribución $\mathcal{N}(\mu, \sigma)$ ¹: `normalvariate(mu, sigma)`
- Distribución Triangular(a, b, c): `triangular(a, b, c)`

2. DCCasino

El gran *DCCasino* pretende estar abierto **siempre**, ya que los apostadores llegan día y noche, como ocurre en Las Vegas. Aún así, asume que el casino abre un día cualquiera al mediodía. En otras palabras, el casino no cierra nunca, sólo cuando la simulación termina. La cantidad de días de duración de la simulación debe ser solicitada al usuario al iniciar tu programa.

En este casino habrá varias entidades relacionándose entre sí, por lo que será vital implementar una buena modelación de clases para tener todo bajo control. Es importante destacar que en este enunciado **no se te dirá explícitamente** cómo modelar las distintas clases y aspectos del programa. Ante esto, deberás modelarlo correctamente acorde a la información que se te proporciona. Hay pistas a lo largo de todo el enunciado.

¹Normal con parámetros μ y σ .

3. Entidades

3.1. Clientes

Los clientes del *DCCasino* son aquellas personas que pueden apostar. Por lo tanto, interactuarán con los distintos juegos e instalaciones de este. Además, los clientes se podrán relacionar e influenciar entre sí.

3.1.1. Características

Los clientes de *DCCasino* tienen un **identificador único**, un **nombre**, y una **edad** mayor o igual a 18 años. En cada instante de la simulación, existe una probabilidad fija p de que llegue un nuevo cliente. Además de las características ya mencionadas, existen otras más que variarán dependiendo de la personalidad del cliente. Estas son:

- **Dinero:** Atributo que representa el nivel de riqueza del jugador.
- **Lucidez:** Atributo que influye en la capacidad de tomar buenas decisiones.
- **Ansiedad:** Atributo que influye en la frecuencia con que se visitan los juegos y la cantidad de dinero que se apuesta en ellos. Aumenta en un 25 % si el jugador posee más del doble o menos de la quinta parte del dinero inicial. La cantidad que un cliente apuesta en los juegos será igual a $(1 + \theta * \text{ansiedad}) * \text{apuesta_base_juego}$, donde θ es un parámetro externo a la simulación.
- **Suerte:** Atributo que influye en la probabilidad (individual) de ganar en los juegos.
- **Sociabilidad:** Atributo que influye en la probabilidad de relacionarse con otros (realizar alguna actividad) luego de participar en un juego.
- **Stamina:** Atributo que influye en la probabilidad de quedarse en el casino después de cualquier acción. Esta quedará en exactamente 0 si el jugador se queda sin dinero.
- **Deshonestidad:** Atributo que influye en la probabilidad de coludirse con otros clientes para contar cartas en los juegos.

También existe la posibilidad de que estos atributos varíen durante la visita al casino, dependiendo de las instalaciones que el cliente use.

3.1.2. Personalidades

Al igual que cualquier otro establecimiento de este tipo, el connotado *DCCasino* admite distintos tipos de personas y, con ello, distintos tipos de personalidades. La personalidad determina si cierto atributo se presenta en un nivel alto, medio o bajo. A continuación, se presentan los valores posibles para cada atributo según su nivel:

- **Alto:** Distribuye *uniforme* en el intervalo $[0,70; 1,00]$
- **Medio:** Distribuye *uniforme* en el intervalo $[0,30; 0,70]$
- **Bajo:** Distribuye *uniforme* en el intervalo $[0,00; 0,30]$

Por ejemplo, si en cierta personalidad el atributo *dinero* se presenta en nivel alto, entonces las personas con esa personalidad tienen en *dinero* un número al azar (con distribución *uniforme*) entre 0,7 y 1.

A continuación, se señalan las personalidades, junto a sus respectivas descripciones:

- **Ludópata:** Adicto al juego. Sus probabilidades de salir del casino antes de varios juegos o quedar sin dinero son bajas. Suele apostar grandes cantidades de dinero, lo que se debe principalmente a su alto nivel de ansiedad. El nivel de cada una de sus características es:
 - **Alto:** Ansiedad, *stamina*.
 - **Medio:** Dinero, lucidez, suerte, sociabilidad, deshonestidad.
- **Kibitzer:** Le gusta mirar como otros juegan, pero duda al momento de apostar su dinero. Suele pasar una gran cantidad de tiempo dentro del establecimiento sin apostar un solo peso, prefiriendo conversar con el resto de los clientes. El nivel de cada una de sus características es:
 - **Alto:** Sociabilidad.
 - **Medio:** Lucidez, suerte, deshonestidad
 - **Bajo:** Dinero, ansiedad, *stamina*.
- **Dieciochero:** Comienza siendo precavido para meterse a un juego, pero no duda en aceptar un vaso de un trago de la temporada. A medida que pasa el tiempo, apuesta cada vez más dinero (y probablemente lo pierde). El nivel de cada una de sus características es:
 - **Alto:** Sociabilidad, ansiedad.
 - **Medio:** *Stamina*, suerte, dinero.
 - **Bajo:** Lucidez, deshonestidad.
- **El ganador:** Nadie sabe cómo lo hace. Quizás cuenta cartas o simplemente tiene suerte pero, a final de cuentas, casi siempre termina ganando más que los demás. El nivel de cada una de sus características es:
 - **Alto:** Sociabilidad, suerte, deshonestidad, *stamina*.
 - **Medio:** Ansiedad, dinero, lucidez.
- **Millonario:** Si bien no es adicto al juego, le agrada apostar de vez en cuando. No duda en poner una gran suma de dinero a disposición del azar por pura diversión. El nivel de cada una de sus características es:
 - **Alto:** Dinero, *stamina*.
 - **Medio:** Sociabilidad, suerte, ansiedad, lucidez, deshonestidad.

Es necesario notar que los valores de los parámetros **no** pueden ser menores a 0, ni mayores a 1. En caso de que, por algún motivo, alguno de los parámetros se salga del rango, debes asegurarte de colocarlo en el extremo más cercano.

La cantidad de dinero inicial de un jugador depende directamente del parámetro aleatorio *dinero*. Un cliente entra al casino con:

$$\text{dinero de entrada} = \text{dinero} \times 200$$

3.1.3. Decisiones

Los clientes toman decisiones al llegar al casino, cada vez que terminan de participar en alguna partida, luego estar en una instalación o después de participar en alguna actividad. En cada uno de estos momentos, el cliente escoge con cierta probabilidad alguna de las posibilidades que tiene: jugar, ir a una instalación, participar en actividad, o retirarse del casino. Las probabilidades de ejecutar cada acción son:

- **Retirarse** = $1 - \text{stamina}$
- **Jugar** = $\min(\text{ansiedad}, 1 - \text{retirarse})$
- **Participar en actividad** = $\min(\text{sociabilidad}, 1 - \text{retirarse} - \text{jugar})$
- **Ir a instalación** = $1 - (\text{retirarse} + \text{participar en actividad} + \text{jugar})$

Cuando el cliente toma la decisión de jugar, escoge un juego de forma que cada opción tiene la misma probabilidad de ser la elegida. Lo mismo ocurre al momento de participar en una actividad, o ir a alguna instalación.

3.2. Personal del casino

El personal del casino comprende a las personas que realizan algún tipo de trabajo para este. El casino tiene contratadas a 120 personas como personal: hay 55 *bartenders*, 62 *dealers*, y 3 Mr. T.'s. Todas deben asistir diariamente al casino en un horario *escogido* por ellas (se detalla a continuación).

3.2.1. Características

Además de tener un **identificador único**, un **nombre**, y una **edad** mayor o igual a 21 años, cuentan con las siguientes características:

- **Juego/instalación:** Corresponde a su lugar de trabajo. Cada uno tiene asignado un juego o instalación en la cual trabajan permanentemente.
- **Tiempo de trabajo:** Duración en **minutos** de su jornada laboral diaria. Esta se calcula mediante una distribución triangular de probabilidad, dependiendo del tipo de personal. A pesar de que recién al comenzar el turno la persona toma la decisión de cuánto tiempo trabajará, el casino exige que deben trabajar **al menos 6 horas, y a lo más 9 horas**, diariamente.
- **Tiempo de descanso:** Es el tiempo en **horas** en que la persona pasa fuera del casino luego de terminar su turno. Para todo el personal, este tiempo **no debe ser menor a 8 horas, ni mayor a 20 horas**. Distribuye $\mathcal{N}(14, 5)$, y debe tomar valores enteros. Se calcula al momento en que cada persona termina su turno.
- **Horario próximo turno:** Es la hora en la que comienza el próximo turno. Depende del tiempo de descanso y la hora del turno que acaba de terminar. El casino impone la regla de que los turnos parten en horas completas, aproximando por exceso. Por ejemplo, si una persona termina su turno a las 20:17, y luego calcula un tiempo de descanso de 13 horas, esta persona debe comenzar su nuevo turno a las 10:00 (del día siguiente), en vez de a las 9:17.

3.2.2. Tipos de personal

- **Bartender:** Se encarga de atender y preparar los pedidos que hacen los clientes en el restobar.
 - **Juego/instalación:** Restobar
 - **Tiempo de trabajo:** distribuye Triangular(360, 540, 490)
- **Dealer:** Los *dealers* son los encargados de entregar las cartas en los juegos del casino, o de supervisar los juegos automáticos. Existen algunos *dealers* que están comprados por la mafia, por lo que ellos aumentan la probabilidad de ganar de los mafiosos. Los *dealers* tienen la facultad de descubrir a

las personas que cuenten cartas, con una probabilidad ω (un valor real que va entre 0 y 1, el cual es un parámetro externo a la simulación) en cada ronda. Las personas descubiertas son expulsadas del casino.

- **Juego/instalación:** Todos los juegos del casino
- **Tiempo de trabajo:** distribuye Triangular(360, 540, 540)
- **Mr. T.:** Lee el *tarot* a las personas y les “dice” el futuro.
 - **Juego/instalación:** Tarot
 - **Tiempo de trabajo:** distribuye Triangular(360, 500, 420)

3.3. Juegos

Para ganar dinero con el casino, se deben poner juegos en él, a modo de atraer gente y servir como ingreso extra para el casino. Cada juego tiene su propia probabilidad de ganancia, la cual determina si el jugador gana o pierde y es afectada por la suerte del jugador, de manera que:

$$\text{probabilidad de ganar} = \text{probabilidad juego} + 0,2 \times \text{suerte} - 0,1$$

Debido a como funcionan las probabilidades², el resultado **siempre** debe ser un número entre 0 y 1.

3.3.1. Tipos de juegos

Debido a las distintas naturalezas de cada juego, estos tienen propiedades específicas.

- **Tragamonedas:** Este juego consiste en sacar tres íconos iguales en una pantalla electrónica. La apuesta mínima del juego es de uno. La probabilidad de ganar este juego es de α , donde α es un parámetro externo a la simulación. Cuando una persona apuesta, 10 % del dinero es ganancia del casino, mientras que el resto queda guardado en un **pozo**, el cual se vacía cuando una persona gana, entregándole la totalidad de ese dinero al ganador.
- **Ruleta:** Este famoso juego consiste en intentar adivinar el número o el color en el cual caerá una pelota dentro de una ruleta. Esta cuenta con números de 0 a γ , con $\gamma \in \mathbb{N}^+$ y par, donde γ es un parámetro externo a la simulación. Cada uno de estos números tiene la misma probabilidad de ser elegido por la pelota y tiene asignado un color dependiendo si este es par (rojo), impar (negro) o cero (verde). La apuesta mínima del juego es de uno. Los jugadores elegirán al azar si quieren apostar a un color o a un número. La ganancia de cada apuesta está dada por:

Tipo de apuesta	Número / Color	Ganancia	Probabilidad Juego
Color	Verde	$\times 5$	$1/(\gamma + 1)$
Color	Negro	$\times 1,5$	$\gamma/2(\gamma + 1)$
Color	Rojo	$\times 1,5$	$\gamma/2(\gamma + 1)$
Número	0 hasta γ	$\times 5$	$1/(\gamma + 1)$

El dinero perdido por los jugadores es ganancia directa del casino, así como las ganancias de los jugadores son pérdidas de este.

²Gracias EYP1113 y Andréi Kolmogórov.

3.4. Instalaciones

Las instalaciones son lugares en donde las personas pueden interactuar con el casino sin la necesidad de apostar. Sin embargo, estas sí influirán en sus estados de ánimo y posteriores decisiones.

3.4.1. Características

Además de tener un **identificador único**, una **ubicación** y un **nombre**, cada instalación cuenta con las siguientes características:

- **Duración por persona:** Tiempo en que una persona se demora en interactuar con la instalación.
- **Gente en espera:** Personas que están esperando a que se desocupe la instalación para poder ocuparla o interactuar con ella. Es básicamente una cola común y corriente. Debe entrar primero el que lleve más tiempo en ella. Sólo habrá cola cuando la instalación esté funcionando, y esta se disolverá si se cierra inesperadamente.
- **Capacidad máxima:** Cantidad límite de gente que la instalación puede soportar al mismo tiempo.
- **Costo:** Precio que una persona debe pagar para utilizar/interactuar con la instalación.
- **Personal de la instalación:** Encargados de ciertas actividades desarrolladas dentro de la instalación. Los requerimientos mínimos de este personal dependerá de cada tipo de instalación.
- **“Funcionando”:** Indica si la instalación está (o no) funcionando. Dependerá si la instalación cuenta con el personal mínimo para operar. En caso de que la instalación deje de operar con personas ya utilizándola/interactuando con ella, estas deben abandonar el lugar de inmediato. Para estas personas será como si nunca hubiesen estado allí.

Por otro lado, cada instalación provoca aumentos o disminuciones en los diferentes parámetros que definen el estado de ánimo de las personas, lo que se describe en detalle a continuación.

3.4.2. Tipos de instalaciones

- **Restobar:** Aquí es donde se vende comida y bebidas *mágicas*. Estas últimas disminuyen en 0,2 la lucidez de los clientes, en 0,15 su ansiedad y en 0,3 la probabilidad de que se retiren del casino. Por otro lado, la comida aumenta en 0,1 la lucidez, y disminuye en 0,2 la ansiedad. Si la lucidez es mayor que la ansiedad, la persona elegirá comprar una bebida, y en el caso contrario, una comida. En caso de ser iguales, será una elección aleatoria.
 - **Duración por persona:** Inversamente proporcional a la cantidad de *bartenders* trabajando en el momento. Queda a libre disposición tuya cómo calcular este número, sin embargo, debe ser mínimo de 10 minutos, y máximo 50 minutos.
 - **Capacidad máxima:** 20 personas
 - **Costo:** 2 por persona
 - **Requerimientos de personal:** mínimo dos *bartenders*
- **Tarot:** En este lugar se leen las cartas para que cada persona pueda *ver su futuro*, por ende, puede aumentar la probabilidad de irse de cada persona, o bien, aumentar su suerte. Aquí es donde trabaja Mr. T.
 - **Duración por persona:** $\mathcal{N}(3, 5)$ minutos

- **Capacidad máxima:** una persona
 - **Costo:** 10 por persona
 - **Requerimientos de personal:** exactamente un Mr. T.
- **Baño:** Acá es donde la gente acude para hacer sus necesidades. El ir al baño repercute en una disminución de 0,1 en la ansiedad.
- **Duración por persona:** $\mathcal{N}(3 \times (1 - \text{lucidez}), 2)$ minutos
 - **Capacidad máxima:** 20 personas
 - **Costo:** 0,2 por persona
 - **Requerimientos de personal:** no tiene

4. Actividades

Aparte de apostar en los diferentes juegos, o interactuar con las instalaciones del casino, existen las *actividades*. Estas dependen directamente de la característica *sociabilidad* de los clientes y tienen una duración variable. Cuando una persona decide realizar alguna actividad, todas tienen la misma probabilidad de ser escogidas. Luego, cada persona tiene que ver si puede (o no) realizar la actividad.

Duración actividad en minutos:

$$\text{máx}(\text{lucidez} + \text{sociabilidad} - \text{ansiedad}; 0,1) \times \pi^2$$

- **Conversar:** La persona espera como máximo δ minutos a que algún otro cliente también decida realizar esta actividad. Si esto sucede, ambos clientes conversan por el tiempo máximo entre los dos definido por la función “duración actividad”. Si es que no hay clientes para conversar, la persona continua con su vida de forma normal. Esta actividad disminuye la ansiedad en $\varepsilon\%$ y aumenta la deshonestidad en χ . Siempre se conversa en **parejas**. Por último, δ , ε y χ son parámetros externos a la simulación.
- **Hablar con *Tini*, *il padrino*:** Tiene un costo de 20. Aumenta la probabilidad de irse en η y aumenta un $\kappa\%$ la probabilidad de ganar **solo** en las mesas donde el *dealer* esté coludido con la mafia. Por último, η y κ son parámetros externos a la simulación.
- **Contar cartas:** Tiene como requisito haber hablado con alguna persona y ser de personalidad *kibitzer*. Luego de estudiar, la persona puede decidir ir a una mesa a contar cartas con probabilidad igual a su parámetro de *deshonestidad*. Si es que decide contar cartas, lo realiza en las siguiente ν rondas que juegue en alguno de los juegos donde se puede. Contar cartas aumenta la probabilidad de ganar en $\psi\%$ sólo en la ruleta. Si la persona es descubierta por algún *dealer*, es expulsada inmediatamente del casino. Por último, ν y ψ son parámetros externos a la simulación.

5. Interfaz gráfica

Adicionalmente, tiene a su disposición el módulo **gui** con el que podría ver a través de una interfaz gráfica el comportamiento de su simulación. En el modulo **gui** se encuentran los siguientes métodos:

- **gui.init():** Este método inicializa la interfaz y debe ser llamado antes de todos los demás métodos.
- **gui.add_entity(entity):** Este método agrega una entidad a la interfaz.

- `gui.run(function, delay=25)`: Este método recibe una función y un número entero que representa un tiempo en milisegundos. Una vez llamado, se mostrará la interfaz y se repetirá la función constantemente. Esta función puede corresponder a un *tick*³ de la simulación.

5.1. Clase **Entity**

Clase base que representa los objetos en la interfaz gráfica y se encuentra en el *package* `gui.entities`. Posee los siguientes métodos:

- `def __init__(self, path)`: En el inicializador se recibe el *path* de la imagen PNG que va a corresponder a la entidad.
- `def add_decoration(self, path)`: Agrega una decoración sobre la imagen de la entidad. Esta decoración rota junto con la entidad. Si el *path* es `None`, quita la decoración.
- `def height(self)`: Retorna la altura en pixeles que tiene la entidad.
- `def width(self)`: Retorna el ancho en pixeles que tiene la entidad.
- `def deleteLater(self)`: Elimina la entidad de la interfaz.

Además, posee los siguientes *properties*:

- `angle`: Ángulo en grados en los que está rotada la imagen y decoración de la entidad.
- `x`: Representa la coordenada horizontal en la que se encuentra la entidad.
- `y`: Representa la coordenada vertical en la que se encuentra la entidad.






Es importante notar que el origen cartesiano —el punto (0,0)— de la ventana parte en la **esquina superior izquierda**. De forma análoga, las coordenadas de las entidades representan la esquina superior izquierda de ellas.

5.1.1. Clase **Human**

La clase **Human** hereda de **Entidad**, por lo que posee los mismos métodos y *properties* que esta, pero tiene ciertos cambios:

- `def __init__(self, personality, x=0, y=0)`: El inicializador de la clase **Human**, en vez de recibir un *path*, recibe el tipo de personalidad. La entidad será representada con una imagen diferente dependiendo de la personalidad.
- `personality`: Es la personalidad que tiene la entidad. Las personalidades posibles son:

³Un *tick* corresponde a un ciclo de la simulación, que como decía en la introducción era de un segundo.


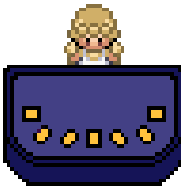
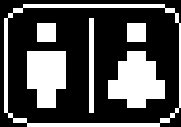
Personalidad	Imagen	Nombre de la imagen
"ludopata"		human_ludopata.png
"kibitzer"		human_kibitzer.png
"dieciochero"		human_dieciochero.png
"ganador"		human_ganador.png
"millonario"		human_millonario.png

Opcionalmente, recibe las coordenadas iniciales de la entidad. En caso de que no se reciba, la entidad se encontrará en la esquina superior izquierda.




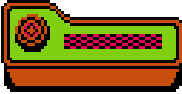
5.1.2. Otras entidades

Adicionalmente, se tienen las entidades **Building**, para las instalaciones del punto 3.4.2, y **Game** para los juegos explicados en la sección 3.3.1. Estas entidades son idénticas a la clase **Entity**, excepto por su inicializador. Los inicializadores reciben un tipo en vez de un *path*, y pueden recibir coordenadas de posición para que puedas ubicarlas en el mapa. En caso de que no se reciba una posición específica, la entidad se encontrará en la esquina superior izquierda.

- Inicializador de **Building**: `def __init__(self, type_, x=0, y=0).`

Instalación	Imagen	Nombre de la imagen
"restobar"		bulding_restobar.png
"tarot"		building_tarot.png
"baños"		building_baños.png

- Inicializador de Game: `def __init__(self, type_, x=0, y=0).`

Juego	Imagen	Nombre de la imagen
		
		
"tragamonedas"		game_tragamonedas.png
"ruleta"		game_ruleta.png

5.1.3. Debugging

En el módulo `gui.entities` hay una variable llamada `DEBUGGING`. Cuando se le cambia el valor a `True`, se pueden ver los contornos de las entidades, que por lo general, son más grandes que la entidad para que no desaparezcan al rotar. Siéntete libre de modificar **este, sólo este y nada más que este** atributo de la interfaz para poder depurar mejor tu programa.

5.2. Mapa

En la carpeta `gui/assets/mapa` se te proporciona el siguiente mapa:

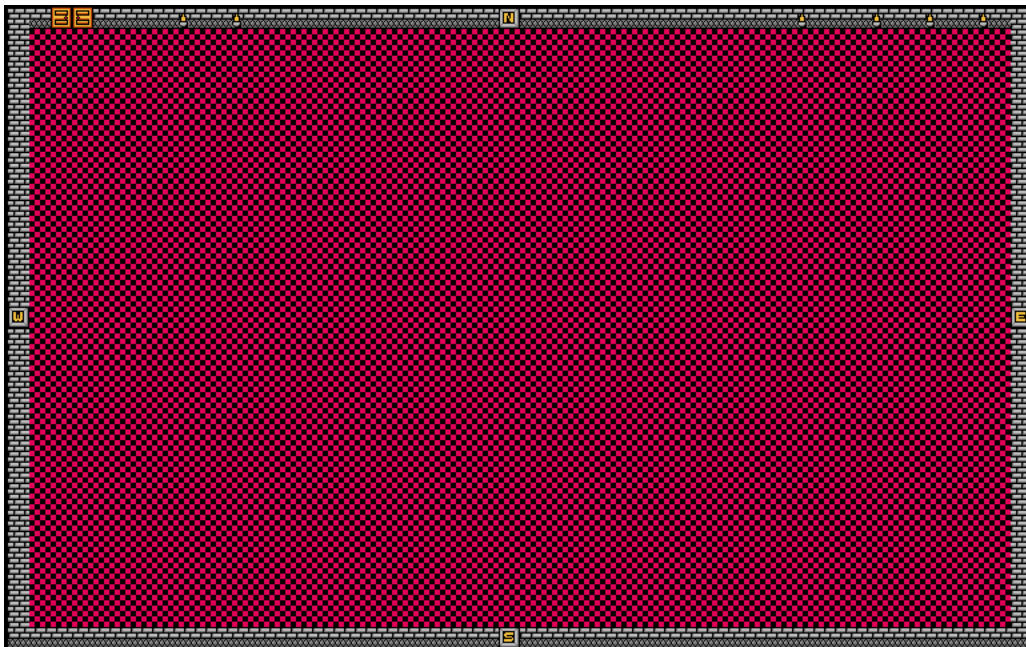


Figura 1: Mapa

En este los clientes de *DCCasino* deben poder moverse para participar en los juegos y las diversas actividades que el casino ofrece.

El movimiento de los clientes debe verse representado en el mapa. Para esto, únicamente debes preocuparte de las colisiones del personaje con los juegos y las instalaciones del casino. Por ejemplo, dos instalaciones

no pueden compartir una misma superficie del mapa, ni una instalación/juego con algún personaje. Sin embargo, no existe restricción entre personajes.

6. Simulación

La simulación consiste en el proceso mediante el cual se modela un sistema y se realizan experimentos sobre el modelo diseñado. El objetivo principal es comprender el comportamiento del sistema o evaluar el funcionamiento del modelo.

6.1. Estadísticas

El resultado final de esta simulación consiste en un conjunto de estadísticas que resumen y cuantifican el funcionamiento y aspectos de interés en el sistema modelado. Para este caso, *Enzini* quiere calcular algunas estadísticas que le entreguen información acerca del rendimiento de *DCCasino*. Se debe definir un tiempo de duración para la simulación; una vez terminada, se deben mostrar las siguientes estadísticas:

1. Dinero promedio final, ganado o perdido, por los clientes del casino. El monto puede ser positivo o negativo, según haya habido ganancias o pérdidas respectivamente.
2. Dinero promedio final, ganado o perdido, por tipo de personalidad. El monto puede ser positivo o negativo, según haya habido ganancias o pérdidas respectivamente.
3. Tiempo promedio de estadía en el casino, por parte de los clientes.
4. Tiempo promedio de estadía en el casino, por tipo de personalidad.
5. Ganancias del casino en promedio por día.
6. Juego que generó una mayor ganancia al casino, en comparación al dinero que entregó en premios.
7. Porcentaje de personas que contó cartas.
8. Razones de salidas del casino. Entregar el porcentaje de cada razón, respecto al total.
9. Tiempo total sin funcionar de cada instalación.
10. Número de personas que visitó cada juego en promedio por día.

6.2. Parámetros

Como podrás haber leído, este enunciado tiene varios parámetros cuyo valor no está definido, y por ende, son externos a la simulación. Por ejemplo, la probabilidad de que entre un cliente (p), o la probabilidad de ganar en las máquinas tragamonedas (α).

Para esta tarea, todos estos parámetros de la simulación deben ser definidos en un archivo separado, que tenga por nombre `parameters.py`. De esta manera, podrán ser modificados con facilidad, permitiendo evaluar la simulación bajo distintos escenarios. Además, el programa debería recibir la duración total de la simulación. Luego de que esta termine, debe imprimir las estadísticas en consola y guardarlas en un archivo de texto.

7. Diagrama de clases

En conjunto con el programa, se tendrá que entregar un diagrama de clases en dos ocasiones distintas. Uno preliminar en la primera semana de la tarea, que permita entregarles *feedback* de cómo va el desarrollo de la modelación de su programa; y uno final junto a la entrega final, que represente fielmente la modelación del problema de su programa.

En ambos casos, esto incluye las clases junto con sus atributos, métodos y *properties*, y todas las relaciones existentes entre estas (agregación, composición y herencia). No es necesario indicar la cardinalidad ni la visibilidad de los métodos o atributos.

8. Notas

En este enunciado, existen varios aspectos que no están especificados. Por ejemplo, cómo elegir la personalidad de las personas que entran al casino, o sus nombres y edades. La idea es que este tipo de elecciones quede a tu criterio.

Además, como podrás haber notado, la descripción de las habilidades de las personas es muy general. Esto es así ya que queda a **tu criterio** definir el nivel de efecto que estas tienen sobre la simulación. Idealmente, mantente dentro de efectos razonables. Por ejemplo, una persona ludópata no debería apostar menos que una persona normal.

9. Restricciones y alcances

- Esta tarea es **estrictamente individual**, y está regida por el [Código de honor de Ingeniería](#).
- Tu programa debe ser desarrollado en Python 3.6.
- Si no se encuentra especificado en el enunciado, asume que el uso de cualquier librería Python está prohibido. Pregunta en la *issue* especial del foro si es que es posible utilizar alguna librería en particular.
- Debes adjuntar un archivo `README.md` **conciso y claro**, donde describas los alcances de tu programa, cómo correrlo, las librerías usadas, los supuestos hechos, y las referencias a código externo. **Tendrás hasta 24 horas después del plazo de entrega** de la tarea para subir el *readme* a tu repositorio.
- Tu tarea podría sufrir los descuentos descritos en la [guía de descuentos](#).
- Entregas con atraso de más de 24 horas tendrán calificación mínima (1,0).
- Cualquier aspecto no especificado queda a tu criterio, siempre que no pase por sobre otro.
- Queda **estrictamente prohibido** el uso de materia que no se ha visto en clases hasta la fecha de entrega del enunciado. Esto incluye conceptos como *threading*, *exceptions*, etc.

Las tareas que no cumplan con las restricciones del enunciado obtendrán la calificación mínima (1,0).