




IIC2233 - Programación Avanzada (II/2016)
Tarea 4

1. Objetivos

- Aplicar conocimientos de simulación síncrona.
- Aplicar conocimientos de testing para verificar funcionalidades en un programa.
- Crear un programa con un buen manejo de excepciones.

2. Introducción

Un día te encontrabas escribiendo muchas formulas en Maplemathica para lograr graficar un coseno hiperbólico cuando miras la ventana de tu progra-cueva y  descubres que los Dioses aparecieron en tu mundo y obligaron a los humanos a luchar entre ellos. Estando en tu progra-cueva, los Dioses no se percataron de tu existencia y quedaste libre de esta lucha. Con la tecnología destruida solo quedaba luchar con espadas y arcos, pero tu computador se logro salvar así que decides simular las peleas para ver a que raza te unirías si es que te pillan.

3. Problema

Para esta tarea, deberás implementar una simulación síncrona de una eventual lucha entre DOS Dioses y sus ejércitos, este duelo consistirá en una batalla por derrumbar el templo enemigo, basándose en la estrategia y personalidades de los Dioses y las características de cada raza. Para ello, deberás tomar en cuenta muchos sucesos que podrían ocurrir en un evento de este tipo y cómo las razas reaccionarían a él. En particular, interesa observar cómo va oscilando la población de soldados por ejército, recursos, cómo se mueven por el mundo y cómo son las interacciones entre ellos y los Dioses (aldeanos recolectando recursos, se efectúan poderes, soldados atacan cuando corresponde).

Al iniciar la simulación, debes preguntar por consola el tiempo de simulación, el Dios principal de cada ejército, la raza que serán y el intervalo de creación de `guerreros:arqueros:mascota` (Ej: creación 2:1:2 equivale a que cada dos guerreros aparecerá un arquero y 2 mascotas)¹

4. Raza

A los Dioses les gusta divertirse así que no solo esclavizaron a la humanidad, sino que transformaron a una gran cantidad de ellos en seres verdes y feos y revivieron a los muertos con el fin de ver a tres razas diferentes

¹ Revisar el punto de Parámetros Iniciales

luchar entre sí.

Se podrá escoger entre tres razas para los ejércitos de la simulación, las cuales tendrán distintas características e interacciones únicas entre sí. Dentro de las razas se encuentran los Muertos Vivientes, Humanos y Orcos. Cada ejército solo puede estar compuesto **por una sola raza**.

Existen características en común con las razas, en particular, todas empezarán con una cantidad definida de aldeanos, una mina de oro, un Dios (se especificará más adelante) y un set de edificaciones disponibles para construir. Ahora, cada raza posee preferencias para desarrollar su estrategia en el campo de batalla. Es deber de ustedes modelar de forma conveniente la estrategia, siempre y cuando cumplan las siguientes restricciones:

4.1. Humanos

Los Humanos son los mas racionales y equilibrados, por lo que su ideal siempre será tener un área sin enemigos. Solo cuando sus tropas correspondan a $3/4$ del ejercito enemigo irán a atacar, manteniendo la mitad de su infantería en la base para defender². En términos genéricos, esta raza preferirá armar una base, desarrollarse con cuarteles y torretas y cuando llegue a cierta cantidad de soldados, ademas del requisito de que no estén siendo atacados, irán a atacar al enemigo. Esta raza solo soporta un máximo de 25 unidades en el campo sin contar

4.2. Orcos

Su naturaleza agresiva y verde los hará tomar decisiones sin medir las consecuencias. Son la raza más agresiva, lo que implica que siempre tomarán la iniciativa para atacar con todo lo que tienen. Preferirán matar a su objetivo más cercano antes que devolverse a defender su base. Además, su gran tamaño y forma los hace ser la raza mas fuerte de las 3, pero la más lenta.

4.3. Muertos Vivientes

Esta raza tiene la particularidad de ser la mas rápida de todas. Su naturaleza de muerto les hizo perder toda fe en los Dioses, por lo que querrán terminar la pelea lo antes posible para poder volver a descansar. Su objetivo es ir a destruir el templo enemigo a toda costa, tratando de esquivar como sea al enemigo. Sin embargo, si llega a ser confrontado, no dudará en atacar de vuelta para luego continuar con su misión. Por otro lado, el tiempo ya los tiene desgastados y por lo tanto poseen menos resistencia ante los ataques de otros (perderán un encuentro más rápidamente).

5. Edificaciones

Todas las edificaciones poseen una cantidad determinada de Puntos de Vida, que determinaran su duración. Una vez que estos puntos de vida lleguen a 0, la edificación desaparecerá (Considere que el Templo al ser el objetivo final deberá ser la estructura con mayor cantidad de Vidas). Notar que los edificios que pueden ser atacados son los templos, torretas y cuarteles. Además, no se puede atacar a edificios en construcción. Todas las razas tienen como prioridad **tener al menos un edificio de cada uno**, es por ello que ante la destrucción de uno, siempre se priorizará construir el faltante antes de continuar con lo que estén haciendo. Al momento de crear una nueva edificación, se deberá asignar **a lo menos un aldeano para construir**. Este deberá dejar de recolectar para poder construir el edificio. También, es necesario diferenciar un edificio que se encuentra en proceso de construcción, de uno que ya este listo. Para esto, puede hacer uso de los distintos assets que se entregan junto con la interfaz.

² En caso de ser un numero impar de soldados, usted decide si trunca o aproxima el número

5.1. Cuartel

En este edificio se crearán las unidades de combate, en particular se podrán crear unidades de combate cuerpo a cuerpo (guerreros) y de combate a distancia (arqueros). Las unidades tendrán un tiempo determinado de creación [5 segundos demora crear un guerrero y 7 un arquero] y se van creando de a 1. Si la proporción entregada es 2:1:2, empiezo a crear un guerrero pasan 5 segundos, este sale y vuelvo a crear un guerrero por 5 segundos hasta que salga. Luego entra a la creación el arquero y después de otros 7 segundos este sale. Después, se pone en pausa la creación de unidades en el cuartel y, en el templo se construye una mascota por 6 segundos y esta sale. Luego se crea otra y finalmente sale del templo luego (después de otros 6 segundos). Notemos que aquí recién se completó un ciclo de la proporción entregada en los parámetros iniciales. Notar que para crear una nueva unidad es necesario tener una cantidad de oro disponible para poder pagarlo (20 de oro para un guerrero y 30 de oro para un arquero).

Se debe repetir esto de forma continua a través de la simulación a no ser de que pase lo siguiente:

- Se termine la cantidad de oro. En ese caso, se pone en pausa la creación hasta contar con el oro suficiente (si la unidad que venía por crear era un arquero, se deberá tener al menos 30 de oro para continuar).
- Si se destruye un edificio, la reconstrucción pasa a ser lo más importante por lo que habrá que esperar hasta que este termine de construirse para continuar con la creación de unidades.
- Si se llega al máximo de unidades por raza (sin incluir a los aldeanos). En este caso, se deberá esperar que alguno muera para continuar. En este punto, el límite máximo se verifica cuando el soldado está listo para salir. Si en ese momento ya se llegó o superó el máximo, el soldado se queda dentro del cuartel hasta que se reduzca la población lo suficiente para que esté dentro de los márgenes y de ahí sale permitiendo continuar el ciclo.

Para construir este edificio se requerirá una cantidad de [100] oro y tomará una cantidad definida por ustedes en construirse (similar a la creación de unidades). Para la construcción de cuarteles y torretas los aldeanos deberán construirlas, es decir, deben mostrar que un aldeano efectivamente está trabajando en el edificio por la duración que tome en construirse (basta con que este se mantenga por el tiempo que dura la construcción en el borde del edificio).

5.2. Torre

Es un edificio defensivo con el que se podrá proteger al Dios de los ataques enemigos. Se puede considerar que tiene un alcance considerable en comparación con los arqueros. Este edificio cuesta [150] de oro en construirse y tiene una duración (definida por ustedes) en construirse. Este edificio deberá tener características como:

- Velocidad de Ataque
- Puntos de vida del edificio
- Puntos de Daño

5.3. Templo

Es el edificio principal de cada raza, en este se deposita el oro recolectado por los aldeanos y se sumará al total de recursos disponibles. Todas las razas empiezan la simulación con un templo donde se crean los aldeanos. El Templo tiene la particularidad de poder crear mascotas, aldeanos y al héroe.

La creación de estas unidades sigue la misma lógica del cuartel, se necesita una cierta cantidad de oro para poder crear una unidad y que no exceda el límite de unidades. Para la creación de un aldeano se necesita [10 de oro y se demora 5 segundos en crearse], toda raza siempre buscará tener 6 aldeanos en el campo, en caso de morir uno dejara su producción de tropas para solicitar la creación de un aldeano para luego continuar

con lo anterior. Para la creación de las mascotas se necesita de [6 segundos y 50 de oro]. El templo tambien posee las mismas restricciones que el cuarter para pausar Considere que el Templo es único por raza, así que **no se puede** construir uno nuevo.

6. Recursos

6.1. Mina de Oro

Es el edificio necesario para poder extraer recursos. Con este, los aldeanos podrán entrar y extraer el oro necesario para crear nuevas unidades, edificios, etc. Cada ejército iniciara con una Mina de Oro dada por las coordenadas entregadas (más adelante) y los aldeanos deberán entrar (puede utilizar el comando `entity.hide()` y `entity.show()` para hacer desaparecer y aparecer a la entidad cuando entre y salga de la mina), recoger oro durante un periodo de tiempo y finalmente ir a depositarlo al templo de su raza. Cada aldeano puede extraer 10 de oro por viaje (sin considerar fuerza), por lo que se necesitan de varios aldeanos trabajando la mina para optimizar los recursos. Sin embargo, no pueden haber más de 5 aldeanos al mismo tiempo sacando oro. La Mina de Oro **no se puede** destruir por los ejércitos y puede considerar que tiene una cantidad ilimitada de oro.

7. Ejército

No puede haber guerra sin soldados y los Dioses lo saben, así que les otorgaron las armas más primitivas que encontraron y los obligaron a luchar por sus vidas. Además, para darle un poco más de entretención, los Dioses dispusieron de sus mascotas para que apoyaran a cada raza.

Para poder sobrevivir, estas personas se mueven por el mundo evitando o eliminando a los otros enemigos que se encuentran, mientras otros buscan oro para ofrecerle a su Dios.

Las distintas unidades tendrán características y atributos que determinarán su eficacia en el campo de batalla, las cuales se detallaran a continuación:

7.1. Guerrero

- Velocidad de Movimiento: Determina qué tan rápido se mueve una persona por el campo de batalla.
- Puntos de vida: Es una abstracción de lo que dura la vida de unidad. Cuando los puntos de vida de una persona llegan a cero esta muere. Estos puntos se pueden ver afectados por golpes de otras unidades, dependiendo del daño de estas.
- Puntos de Daño: Representa el daño infligido por esta unidad a otras unidades/edificios.

7.2. Arquero

Posee los mismos tipos de atributos que el guerrero, pero además:

- Rango: Representa el rango de disparo del arquero a otras unidades.

7.3. Mascota

Las mascotas se consideran unidades con rango medio, es decir, no necesitan estar a corta distancia como los guerreros pero tampoco tendrán un rango como el de los arqueros (queda a su criterio el rango que les pongan, mientras sea coherente).

. Estas contarán con atributos similares:

- Puntos de Vida, Velocidad de Movimiento, Puntos de Daño, Rango

- Rabia: Característica única de las mascotas. Cuando estas vean que una unidad aliada está siendo atacada dentro de su rango de batalla (ustedes definen el rango), esta dejará de lado todo lo demás e irá a protegerla.

7.4. Aldeano

Es la unidad encargada de recoger oro para los fines del Dios, es por ello que **no se considera** como parte del ejército al momento de contabilizar a tus tropas. Este posee:

- Velocidad de movimiento, Puntos de vida al igual que cualquier otro soldado.
- Velocidad de recolección: Esta atributo es característico de los aldeanos, representa la velocidad con la que recogen material en la mina y salen de esta.
- Velocidad de Construcción: Esta habilidad determinara que tan rápido se construye un edificio, esta habilidad determinara que tanto mas (o menos) se demora en realizarse el edificio a partir de su tiempo default de construcción.³.
- Fuerza: Este atributo determinara cuanto más (o menos) puede acarrear un aldeano desde la mina de oro (la base es 10, considere que lo máximo que puede llevar un aldeano es 15).

7.5. Héroe

Todo ejercito necesita un líder y los Dioses te ofrecieron uno. Estas unidades son superiores a cualquier soldado, algunos tienen más vidas y más daño, otros tienen poderes. Dentro de los héroes a optar están:

- Cristian: Es el héroe de raza humana, como mago su daño es muy reducido, pero éste tiene la capacidad de convertir unidades enemigas en copias exactas de él. Para esto, necesita realizar un ritual y permanecer a cierto rango mínimo de la unidad objetivo a convertir. Una vez completado el ritual (tiempo y distancia definido por ustedes) la unidad objetivo pasará a formar parte de las tropas de los humanos, independiente de la raza que era. Estas copias tendrán los mismos atributos que Cristian, pero no podrán ejercer su poder de transformar.
- Manu: Es el héroe de la raza de los orcos, este tiene la capacidad de crear dos mascotas que lo ayudarán en la batalla (una vez invocadas no deberá poder invocar de nuevo sino pasado un lapso largo de tiempo definido por ustedes). Estas no siguen el flujo regular de creación de unidades (no intervienen en ese proceso). Al invocar la creación, estas aparecen automáticamente una a cada lado del héroe y no se deben pagar. Estas mascotas tenderán a protegerlo a él sobre al resto de las unidades, dado que es su maestro.
- Hernan: Es el héroe de la raza de los muertos, este tiene una particularidad, posee 3 vidas, y por lo tanto después de morir, tras esperar un tiempo (determinado por ustedes) reaparece en su misma posición. Luego de perder todas las vidas, vuelve al infierno a descansar.

Todos los héroes serán invocados en su templo respectivo, ellos serán invocados por los Dioses cada cierto tiempo (Parámetro inicial - Tasa de Invocación héroe) y si mueren en batalla se demoraran un tiempo, misma tasa de invocación, en ser revividos por los Dioses. Por lo tanto, no es necesario tener oro para poder tener un héroe, los Dioses son benevolentes en entregarte uno gratis.

³ Si el edificio se demora 10 segundos y el aldeano tiene mucha velocidad de construcción el edificio se hará en 8-9 seg, por ejemplo.

8. Dioses

Los Dioses se aburrieron de solo observar cómo se mataban entre razas, así que decidieron ser partícipe de la masacre que se estaba llevando al cabo. Pero no contaban que algunos se terminarían encariñando con ciertas razas y pelearían para ayudarlos, mientras que otros simplemente se aliarían con una solo para dejar en ridículo a otro Dios (sí... existía envidia entre Dioses).

Los Dioses encargados de esta abominable hazaña⁴ tienen nombre y apellido, pero además tienen personalidades que definirán el comportamiento del ejército dados por sus gustos particulares. Es por esto que deberán modelar los efectos de los distintos Dioses para cada raza en particular, que de por si tienen sus preferencias.

- ~~El Dios del debugging~~ GodPezoa: Es un Dios un tanto bipolar, en general, prefiere quedarse durmiendo pacíficamente al interior de la base sin intenciones de atacar a sus enemigos, pero de momento que lo atacan.... se desata su furia incontenible. Este Dios afectará a la raza reduciendo el nivel de agresividad de esta mientras no sean atacados. Cuando GodPezoa entra en modo furia utilizará su poder apenas pueda y todas las unidades de la raza atacarán para satisfacer a su Dios que solo busca venganza por haber sido despertado.
- Jundead: Es el Dios más malo de los cielos. Lo único que quiere es ver a los ejércitos destruidos y ser un agradable espectáculo para él. Cuando se este bajo el mando de este Dios, los ataques que se ejecuten tendrán como objetivo adicional romper los edificios que encuentren a su paso (cuarteles, torretas) para tratar de dejar a la raza enemiga en las ruinas.
- GodessFlo: Es una Diosa amante de los animales, por esta razón, la raza se verá en la obligación de cambiar la proporción de las mascotas creadas y que esta tome el valor de la proporción más grande y se le sume dos (si la relación era 3:1:2, deberá cambiar y ser 3:1:5), para mantenerla feliz y que esté dispuesta a ocupar su poder. En cuanto a su personalidad es de carácter neutral, sin embargo en peleas, al ser una amante de los animales, privilegiará defender a sus mascotas de los enemigos (Incluye mascotas enemigas) antes que atacar otras unidades.
- Godolfo: Este Dios goza viendo a sus enemigos morir. Cuando Godolfo está al poder, el ejercito tendrá como objetivo matar a la mayor cantidad de soldados enemigos. Una vez que el ejército de Godolfo duplica a la del enemigo, este comenzará a atacar el Templo.

Los Dioses vienen representados por los distintos tipos de templos (ver assets). Estos se tendrán que implementar dependiendo del Dios escogido en la fase de ingreso de parámetros para la raza correspondiente.

Deben especificar en su Readme como enfrentarán la personalidad del Dios con la naturaleza de la raza. Si una raza es agresiva pero el Dios es pasivo, deben decidir (y especificar) si alguna predominará o si será una mezcla de ellas.

8.1. Poderes

Los Dioses no son Dioses por que sí, estos tienen poderes increíbles que amenazan con la integridad de todas las unidades en el campo de batalla. Sin embargo, dado el nivel de energía requerido para usarlos, estos solo pueden ocupar a lo más 1 poder. La elección del poder es aleatorio para cada simulación y **no se podrán repetir** entre dioses enemigos. Además, tendrán requisitos para usarse.

- Plaga: Este poder hará que todas las unidades enemigas se contagien de la plaga. Esta reducirá la velocidad de movimiento y los puntos de vida de las unidades de manera constante durante un periodo de tiempo determinado⁵. Una vez transcurrido el tiempo de la plaga, las unidades que sobrevivan al ataque volverán a su velocidad normal.

⁴ Al igual que de todas sus tareas

⁵ La duración la determinan uds.

- Berserker: Este poder hará que las unidades hagan el doble del daño normal, con el costo de que su hp disminuya a 1. Como los Dioses son tan generosos, permitirán que las unidades tengan un tiempo en que son inmortales (Especificar la Duración README). Una vez transcurrido el tiempo, las unidades afectadas perderán la bendición de la inmortalidad y podrán morir (seguirán teniendo 1hp y doble daño)
- Terremoto: Los Dioses utilizarán su poder sobre la Tierra, provocando un terremoto en tierras enemigas. Este poder se ejecutará por un periodo de tiempo al azar⁶. Mientras se encuentre funcionando, las construcciones del enemigo sufrirán daño de manera constante, siendo capaz de destruirlos si sus puntos de vida llegan a 0. Solo existe una excepción, el Templo enemigo **puede recibir daño por el terremoto, pero como es un edificación chilena, cuando llega a 10 de hp ya no es afectado por tal catástrofe, es decir, ya no se reduce más su hp por el terremoto.**
- Invocar a los muertos: Con este poder, los Dioses demostrarán su infinita generosidad y traerán de vuelta a la vida a los últimos 7 soldados muertos en combate que sean de su propia raza. Estos contarán con los mismos atributos que tenían antes de su muerte, pero con la mitad de sus puntos de vida.
- Glaciar: Como su nombre lo indica, se generará una masa de hielo bajo los pies de la raza enemiga y estos serán congelados. El estado congelado previene a las unidades de moverse y no podrán atacar por la duración del poder. La duración del glaciar será definida por ustedes, siempre y cuando tenga un orden similar al de los otros poderes.

8.2. Activación Poderes

Los poderes de los Dioses tienen requisitos y a estos se les llamará objetivos. Al inicio de la simulación, se deberá escoger de manera aleatoria entre los distintos objetivos que se muestran más adelante. Este objetivo se debe mostrar en la parte superior de la interfaz y será el mismo para ambos ejércitos (la manera de realizar esto se especifica en Interfaz Gráfica). El primer ejército en lograrlo será bendecido con la ayuda de su Dios y desde ese momento podrá ocupar su poder cuando la situación lo amerite. Una vez completado el objetivo, se volverá a elegir uno de forma aleatoria y se repetirá el proceso. Los objetivos son:

- Recolectar recursos: Se indicará una cantidad al azar entre 200 a 500 unidades de oro que los ejércitos deberán recolectar. No es necesario acumular el oro, sino solo llevar un registro de cuanto se ha recolectado desde que se declaró el objetivo. Se debe mostrar un mensaje similar a 'Recolectar 300 de oro'.
- Matar unidades: Se indicará una cantidad aleatoria entre 5 a 10 de un tipo de unidad que deberán matar de la raza enemiga. El contador de muertes solo comienza a contar desde el momento que se anuncia el objetivo. Se debe mostrar un mensaje como 'Eliminar 10 arqueros'.
- Creación de 'x' unidades: Para los Dioses una vez que se hayan creado una cantidad de 'x' unidades, podrán activar el poder. Las unidades por default son Guerreros, excepto para el caso de GoddessFlo que es amante de los animales, en este caso deberán ser "x" mascotas. El mensaje debe ser como 'Crear 5 unidades'.

9. Assets

A continuación se mostrará todos los assets utilizados para esta simulación:

⁶ Mínimo de 3 segundos, máximo de 15 segundos



(a) Human - distance



(b) Human - pet



(c) Human - warrior



(d) Human - villager



(e) Human - villager2



(f) Orc - distance



(g) Orc - pet



(h) Orc - warrior



(i) Orc - villager



(j) Skull - distance



(k) Skull - pet



(l) Skull - warrior



(m) Skull - villager



(n) Heroe Humano



(ñ) Heroe Orco



(o) Heroe Muertos vivientes



(p) Templo - Flo



(q) Templo June



(r) Templo Pezoa



(s) Templo Rodolfo



(t) Cuartel en construcción



(u) Torreta en construcción



(v) Cuartel



(w) Mina



(x) Toreta

10. Interfaz Gráfica

Adicionalmente, tiene a su disposición el módulo gui con el que podrá ver a través de una interfaz gráfica el comportamiento de su simulación.

En el modulo gui se encuentran los siguientes métodos:

gui.set_size(largo, ancho) Este metodo recibe dos int, largo y ancho para configurar el tamaño de la pantalla. Por defecto se solicita minimo 1024 x 680.

gui.set_objective(string) Este metodo recibe un string y lo imprime en la pantalla de la interfaz como objetivo.

gui.run(función) Este método recibe una función. Una vez llamado, se mostrará la gui y se repetirá la función constantemente. Esta función puede corresponder a un tick.

gui.add_entity(Entidad) Este método agrega una entidad a la interfaz.

set_gold_t1(int) Este metodo recibe un int y lo imprime en pantalla como el oro del equipo 1(izquierda).

set_gold_t2(int) Este metodo recibe un int y lo imprime en pantalla como el oro del equipo 2(derecha).

10.1. Entity

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

__init__(path, size, hp, pos) En el init se recibe el path de la imagen png que va a corresponder a la entidad en forma de lista, de modo que si existe una carpeta dentro de assets, el path de este método debería ser una lista de la forma [carpeta", carpeta", ..., imagen.png"].

add_decoration(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.

height() Retorna la altura en pixeles que tiene la entidad.

width() Retorna el ancho en pixeles que tiene la entidad.

hide_hp_bar() Recibe un booleano que en caso de ser True, esconde la barra de vida ya en caso contrario la muestra.

y las siguientes properties:

angle Ángulo en grados en los que va a estar rotada la imagen y decoración de la entidad.

cord_x Representa la cordenada horizontal en la que se encuentra la entidad (las coordenadas parten desde la esquina superior izquierda de las entidades).

cord_y Representa la coordenada vertical en la que se encuentra la entidad.

health Representa la cantidad de hp (vida) que posee la entidad (de 0 a lo especificado en el init, por defecto el maximo es 100).

10.1.1. Human, Orc, Skull

Se encuentran en el modulo gui.kinds. Las clases Human, Orc y Skull heredan de Entidad, por lo que posee los mismos métodos y properties que esta. Lo que cambia es su método **init**. El init de estas clases reciben el tipo de unidad a la que corresponde ("villager", "pet", "warrior", "heroz", "distance") y, opcionalmente, la posición en la que se encuentran (en forma de tupla (x, y)) y la vida máxima si no se quiere por defecto 100.

10.1.2. Building

La clase Building hereda de Entidad, por lo que posee los mismos métodos y properties que esta. Lo que cambia es su método **init**. El init de la clase Building recibe el tipo de edificio a la que corresponde ("mine", "tower", "barracks") y, opcionalmente, la posición en la que se encuentran (en forma de tupla (x, y)).

Esta clase posee una property llamada in_construction, que si se le asigna un True cambia el asset del edificio a uno en construcción y en caso contrario lo pone como cualquier otro.

10.1.3. Temple

La clase Temple hereda de Entidad, por lo que posee los mismos métodos y properties que esta. Lo que cambia es su método **init**. El init de la clase Temple recibe el dios al que corresponde ("flo", "june", "pezoa", "rodolfo") y, opcionalmente, la posición en la que se encuentran (en forma de tupla (x, y)).

10.1.4. Debugging

En el módulo entities hay una variable llamada _debugging. Cuando se le cambia el valor a True, se pueden ver los contornos de las entidades (por lo general, son más grandes que la entidad para que no desaparezcan al rotar). Siéntanse libres de modificar **este, solo este y nada más que este** atributo de la gui para poder debuggear mejor sus programas.

10.2. Cómo usar estas clases

La idea de estas clases es que ustedes **hereden** de ellas y las extiendan, de forma que puedan darles el comportamiento deseado. Adicionalmente, pueden crear sus propias entidades si así lo estiman conveniente (heredando de la clase Entidad), sin embargo, no es requisito que agreguen ninguna entidad adicional.

11. Archivo Mapa

Se les entregará un archivo mapa.csv que contendrá las especificaciones de donde se deben posicionar las distintas edificaciones. La primera columna corresponde a la coordenada x, la segunda a la y, la tercera al tipo de edificio (si es cuartel, mina, templo o torreta) y la cuarta el número de ejército al que pertenece (1 o 2).

Por ejemplo:

Archivo	X	Y	TIPO	EJERCITO
Linea 1	700	500	Cuartel	2
Linea 2	100	190	Cuartel	1
Linea 3	850	520	Templo	2
Linea 4	800	50	Mina	2
Linea 5	200	40	Torreta	1

Nota: Puede asumir que los edificios nunca se sobrepondrán y que el nombre no cambia respecto a los mostrados en el ejemplo. Por otro lado el archivo final presentará 8 edificaciones, dos de cada tipo.

12. Parámetros Iniciales

Al iniciar la simulación, el programa deberá preguntar el tiempo de simulación, el Dios principal de cada raza, la raza que serán y la razón de creación de "guerreros : arqueros : Mascotas" (Ej: creación "2:1:1- cada dos guerreros aparecerá un arquero y luego una mascota). El Dios y la raza **deben ser distintos** entre cada ejército. Además de lo anterior, deberá entregar la tasa de invocación del héroe, esto determinará cuándo

arribará el héroe al campo de batalla y cuánto se demorará en revivir en caso de morir. (Asumir arribo exponencial)

Al iniciar la simulación deberá partir con 5 aldeanos y una cantidad de oro igual a 800. Luego de que esta termine, debe imprimir las estadísticas en consola y guardarlas en un archivo de texto.

13. Excepciones

Se espera que se manejen todas las excepciones correctamente (i.e: indicando el tipo de excepción a capturar y lo que se hará con ella. Una excepción genérica no se considerará como 100 % correcta, a menos que sea la única solución posible - explicar en el README.md por qué es la única solución posible).

14. Testing

Al menos un 50 % de todas las funciones que implemente deben estar testeadas (su programa debe ser modular, i.e: no tener una función que realice más tareas de las necesarias cuando se puede separar en dos funciones - el objetivo general es la re utilización de estas).

15. Estadísticas

Se debe definir un tiempo de duración de la simulación, con una cantidad de aldeanos iniciales y de oro. Una vez terminada la simulación, se deben mostrar las siguientes estadísticas:

- Número de Soldados Creados/Muertos por raza
- Numero de Arqueros Creados/Muertos por raza
- Numero de Mascotas Creadas/Muertas por raza
- Numero de Poderes Utilizados por el Dios
- Efectividad del poder utilizado (Cantidad de victimas en caso de Plaga o Cuantas unidades eliminaron a otras en el caso de Furia)
- Tasas de Creación de Unidades Y Tasa de Muerte (por minuto) por raza
- Cantidad de oro gastada en Unidades/Edificios
- Tasa de Extracción de oro por minuto
- Resultado Final... ¿Qué Dios y raza fueron victoriosos... o hubo empate?

16. Notas

Como podrás haber notado, la descripción de los soldados, héroes y Dioses es muy general. Esto es así ya que queda a tu criterio definir el nivel de efecto que estas tienen sobre la simulación. Idealmente, manténgase dentro de efectos racionales. Por ejemplo, una soldado con poca vida no debería poder sobrevivir a un encuentro con un enemigo y debería tener pocas probabilidades de escapar.

17. Bonus

Se otorgará una bonificación a criterio del ayudante según la calidad de la simulación y representación fiel de lo que ocurre en la batalla.

Hint: Si ocurre algún daño físico, se ve sangre, si alguien muere, aparece una tumba, etc. Para esto, cuentan con el método:

```
entidad.agregar_decoracion(path)
```

18. Restricciones y alcances

- Tu programa debe ser desarrollado en Python 3.5
- Esta tarea es estrictamente individual, y está regida por el Código de Honor de la Escuela: Clickear para Leer.
- Su código debe seguir la guía de estilos PEP8
- Si no se encuentra especificado en el enunciado, asuma que el uso de cualquier librería Python está prohibido. Pregunte por foro si se pueden usar librerías específicas.
- El ayudante puede castigar el puntaje⁷ de tu tarea, si le parece adecuado. Se recomienda ordenar el código y ser lo más claro y eficiente posible en la creación algoritmos.
- Debe adjuntar un archivo `README.md` donde comente sus alcances y el funcionamiento de su sistema (*i.e.* manual de usuario) de forma *concisa* y *clara*.
- Cree un módulo para cada conjunto de clases. Divídalas por las relaciones y los tipos que poseen en común. **Se descontará hasta un punto si se entrega la tarea en un solo módulo**⁸.
- Cualquier aspecto no especificado queda a su criterio, siempre que no pase por sobre otro.

19. Entrega

- **Fecha/hora:** 21 de Octubre - 23:59 hrs
- **Lugar:** GIT - Carpeta: Tareas/T04

Tareas que no cumplan con las restricciones señaladas en este enunciado tendrán la calificación mínima (1.0).

⁷ Hasta -5 décimas.

⁸ No agarre su código de un solo y lo divida en dos módulos, module su código de forma inteligente