



Pontificia Universidad Católica de Chile
Escuela de ingeniería
Departamento de ciencias de la computación
Profesor: Hernán Cabrera

IIC2513 Tecnologías y aplicaciones WEB

Segundo semestre 2020

Introducción

Las tareas tienen como objetivo que enfrenten un desafío de diseño y programación que les permita practicar los conocimientos adquiridos durante este semestre.

Las tareas están vinculadas entre sí y la última de las tareas tendrá como resultado una aplicación totalmente funcional con un lado cliente y otro servidor. Es decir, ustedes deberán cumplir con objetivos intermedios de la tarea de manera de llevar un diseño e implementación inicial a un “producto” totalmente funcional. Lo anterior revela la importancia de no atrasarse ni saltarse una entrega y procurar mantenerse al día.

Los enunciados dan la línea general de la funcionalidad que deben implementar, pero sin entrar en mayores detalles ni puntos específicos de tal manera que ustedes demuestren su capacidad de trabajo en equipo y resolución de problemas. Salvo excepciones, tendrán total libertad en cuanto a la interfaz y la implementación de su trabajo.

Para las tareas deberán formar un grupo (que se mantendrá para todas las entregas) de 2 o 3 personas como máximo.

Para inscribir los grupos se creará un formulario en google forms donde podrán registrarlos.

Marco general de las tareas

El resumen de una línea:

Deberán construir un juego basado en turnos e implementado en WEB.

Reglas y características generales del juego

1. Deberá permitir de 2 a 5 jugadores
2. Todos los jugadores son “humanos”
3. El juego será en base a turnos

4. Los turnos son enviados a un servidor una única vez por jugador por turno, en un formato de texto, diseñado por ustedes
5. El servidor recibe, hasta una cierta fecha límite, por turno, todas las jugadas de todos los jugadores inscritos
6. Los turnos son todos entregados hasta una fecha límite y periódica, Por ejemplo, una vez a la semana
7. Los jugadores humanos tendrán entradas diferenciadas con usuario y password, cada uno desde su browser
8. Las jugadas se arman y realizan desde el browser
9. La página WEB debe tener información de las reglas del juego
10. La página WEB sólo permitirá realizar jugadas del jugador conectado
11. La página WEB debe mostrar:
 - a. El estado del juego en general
 - b. Ranking de los jugadores (lugares)
 - c. Bitácora con las jugadas realizadas
12. La entrega de turnos al servidor se realizará desde una función en la página WEB donde el jugador, una vez satisfecho de sus movimientos y jugadas, decide enviar el turno hacia el servidor
13. El periodo de la entrega de los turnos es configurable
14. Si un jugador no realiza la entrega de su turno, se considera que “pasa” pero el computador sigue trabajando el juego con el resto de las acciones entregadas
15. El “computador” será quien procese los turnos y actuará de árbitro
16. Terminado de procesar los turnos, el computador entregará el resultado a cada jugador
17. Las jugadas las almacena el servidor
18. El juego tendrá un tablero donde ocurre la acción (piensen en un juego de turnos estilo Ataque, Catan u otros por el estilo con la diferencia que los turnos se entregan todos de una sola vez)

Tipos de juegos permitidos

Los siguientes son tres tipos de juegos que pueden utilizar.

Las reglas que se describen a continuación, son las mínimas por tipo de juego. Hay mucho en el diseño que dilucidar y también en la forma del juego, los turnos, etc. ¡Eso es parte de lo que ustedes harán!

A continuación se presentan los 3 tipos de juegos propuestos.

Tipo 1: Dominio de territorio y recursos

El objetivo típico de estos juegos es conquistar todos los territorios, las “capitales” ciertas zonas específicas, etc. e implica enfrentamientos entre los participantes

Este tipo de juego debe tener al menos:

1. Territorios que se enfrentan (puede ser la tierra, Star Trek, imperio romano, fantasía que ustedes quieran)
2. Cada territorio posee un propietario, límites y recursos que entrega (hay territorios con más recursos, otros con menos)
3. Adicionalmente un territorio puede recibir mejoras en defensa los cuales servirán de “bono” a la defensa de una agresión desde otro territorio
4. Los territorios pueden tener o no una unidad de defensa
5. Puede haber distintos tipos de unidades que defiendan un territorio (por ejemplo, legiones, manipulas, super legiones, ejércitos, etc) Las unidades tienen cierta cantidad de puntos de ataque, cierta cantidad de puntos de defensa y cierta “resistencia”
6. Cada turno, se calcula la cantidad de recursos y se le entregan al jugador específico el cual deberá gestionar sus recursos para mejorar defensa de un territorio, mejorar la capacidad de ataque o defensa de una unidad en específico
7. Las unidades solo se pueden mover a un territorio adyacente.
8. Si el territorio le pertenece al mismo jugador, simplemente la unidad se mueve
9. Si el territorio le pertenece a otro jugador, se considerará un ataque
10. Por turno, el computador deberá calcular ataques, defensas y mejoras de territorios según las reglas y coherencia que ustedes determinen. Algo incoherente, por ejemplo, sería que un territorio conquistado, pueda mejorar en el mismo turno) su defensa luego de ser conquistado. Por eso la regla sería, hacer todas las mejoras a territorios y unidades y luego procesar los movimientos.
11. El computador en caso de empate debe determinar por algún método el ganador (un random? simular tirada de dados?)

Tipo 2: Gestión de organización o país u otro

El objetivo de este juego es por lo general obtener cierta cantidad de puntos, donde el ganador es el que sobrevive o el que tiene mayor puntaje enfrentado a las dificultades aleatorias que el servidor arroja y que ustedes deben resolver con sus recursos y eventual colaboración entre participantes.

Este juego debe tener al menos:

1. Repartir aleatoriamente recursos escasos a la organización, por ejemplo “ciudades” o “países” y pueden recibir presupuesto, alimentos, profesionales u otros
2. Los recursos sirven para producir ciertos resultados ya sean finales o intermedios
3. Los recursos intermedios son producidos por alguna cadena de producción que ustedes determinen. Por ejemplo, se requiere carbón y hierro para producir acero y acero para producir arados y arados para producir comida (las cadenas las definen ustedes)
4. Deben tener al menos 2 producciones de tipo “cadena de producción”, es decir, recursos que se convierten en otro final
5. Puede haber solicitud de comercio al servidor el cual intercambia alimento por dinero y viceversa
6. Al final todos los recursos producen alguna unidad de intercambio (monetaria u otra) que sirve para la gestión
7. Para poder producir lo que sea, se deben tener especialistas que requerirán cierta cantidad de recursos para mantenerlo.
8. Los recursos de un especialista son de dos tipos: monetarios y de alimentación
9. Hay especialistas en producir comida (por ejemplo, agricultores), otros en producir bienes (que al final es recurso monetario)
10. Usted deberá equilibrar sus recursos
11. Debe haber algunas restricciones, por ejemplo, el terreno para producir alimentos es limitado
12. Deben tener la posibilidad de mejorar (vía recursos monetarios) a los especialistas para que logren más producción
13. Los especialistas requieren diversos usos de recursos monetarios y de alimentación
14. Aleatoriamente el servidor puede arrojar algún tipo de catástrofe a todos los jugadores a la vez que afecte la generación de alimentos o de recursos monetarios (por ejemplo, una inundación hace que la producción de alimentos caiga un 20%)
15. Si la producción no alcanza para mantener a los especialistas, estos se van del lugar
16. El juego comienza con una cierta cantidad de recursos, terreno y especialistas fija para cada jugador.

Tipo 3: Aventura/descubrimiento

El objetivo de este juego es por lo general obtener cierta cantidad de elementos de recompensa (por ejemplo, tesoros, banderas, etc). antes que el resto de los jugadores. Los jugadores poseen un equipo especialista y van avanzando por un

mapa “oculto” que a medida que van saltando de casilla en casilla, descubren, con sus recompensas o sus dificultades (o a veces, nada)

1. Hay un tablero y diversos equipos descubriendo el territorio común (mapa desconocido)
2. Los jugadores poseen un grupo de personajes (o razas) particulares
3. Típicamente un personaje “fuerte” o guerrero, otro mágico, otro que sana, etc.
4. Los jugadores se desplazan por el tablero, descubriendo la zona y obteniendo las recompensas ocultas en el mapa
5. El mapa es generado por ustedes, pero los recursos son colocados en el mapa de forma aleatoria. por el servidor
6. Hay recursos de tipo renovable (que los puede obtener más de un jugador) y hay recursos que se extinguen, es decir, lo obtiene el que primero lo recupera.
7. El “grupo” de personajes se mueven todos en bloque
8. En las casillas también hay amenazas que deben sortear (un dragón, un mago malvado, etc)
9. Si dos o más equipos van a una misma zona (casilla) del mapa, se enfrentan y luchan por la zona.
10. Ustedes determinan las reglas del enfrentamiento
11. Los personajes pueden “mejorar” de acuerdo a los recursos que hayan obtenido
12. Hay una casilla “zona cero” que es exclusiva de un jugador (inconquistable) y que solo sirve para curar al equipo. Un equipo herido es enviado por el servidor automáticamente a esa casilla para reponer fuerzas cuando uno o más de los personajes tiene una salud menor a cierto parámetro.
13. Los personajes consumen energía cada vez que tienen enfrentamientos o pueden usar cierta cantidad de unidades de su habilidad por turno (por ejemplo, el que sana puede entregar cierta cantidad de puntos de sanación por turno, al igual que el mago)

Postulación de otro tipo de juego

Si postulan un tipo de juego diferente, este puede ser aceptado, aceptado con reparos (que deben mejorar algún aspecto) o rechazado.

Lo importante es que deben tener:

1. Un mapa donde jugar
2. Características de las casillas del mapa
3. personajes, unidades u otro que requieren recursos
4. Recursos limitados
5. Posibilidad de mejora de unidades, territorio, ambos, etc.
6. Turnos que se envían todos a un servidor

7. Un servidor que dirime y comunica el resultado del turno

Indicaciones

1. Para alojar sus aplicaciones en Internet, la plataforma soportada y recomendada será [Heroku](#). Si desean utilizar otro proveedor, pueden hacerlo, pero deberán justificar y documentar su elección.
2. El código deberá entregarse en Github. Les indicaremos, una vez oficializados los grupos, cuáles son los repositorios asignados por grupo
3. El resultado final esperado es que ustedes tengan que realizar un despliegue real de su aplicación en modo producción y que ésta quede disponible para cualquier usuario de Internet. No cumplir con ello implica un castigo en la evaluación.
4. Para evaluar cada entrega, todos los miembros del equipo tendrán una reunión de corrección con el ayudante que se les haya asignado. La no asistencia (salvo justificación de peso) de alguno de los integrantes implica un castigo en esa entrega.
5. El avance logrado en cada entrega se evaluará en base a lo mostrado en Heroku en el momento de la reunión con el ayudante y a lo subido al repositorio Github en la fecha límite de la entrega.
6. Las entregas se evaluarán según la siguiente escala:
 - 7.0: Muy buen trabajo. Avance mayor del solicitado; excede expectativas.
 - 6.0: Buen trabajo. En línea con la exigencia.
 - 4.0: Suficiente. Cumple en forma mínima con la exigencia.
 - 3.0: Insuficiente. Muy por debajo de lo solicitado.
7. En el caso de que un grupo no entregue nada, la nota será de un 1.0.
8. Para asegurar una participación equitativa y justa de los integrantes del grupo se implementará una evaluación de pares. Los detalles y funcionamiento de la evaluación de pares se entregarán en documento aparte.

Condiciones y restricciones

1. La aplicación debe desarrollarse (obligatoriamente) usando: Node.js, koa, HTML, CSS, Javascript
2. Para la construcción de la aplicación pueden usar jquery y React
3. Para base de datos deberán usar Postgres y para conexión a la base de datos pueden utilizar pg, knex o un ORM como sequelize
4. Pueden usar también EJS, SACC/SCSS, less, stylus u otros preprocesadores de CSS

5. **NO** está permitido el uso de lenguajes que compilen a JavaScript, como CoffeeScript o TypeScript. Tampoco podrán usar bibliotecas de componentes HTML/CSS/JS como Twitter Bootstrap o Zurb Foundation.
6. Cualquier package o biblioteca adicional que quieran usar debe ser aprobado por el equipo docente.

Recomendaciones

1. Diseñen muy bien las reglas, el entorno, los turnos, la resolución de empates y conflictos
2. Planifiquen el trabajo para que les permita la colaboración entre los integrantes del equipo
3. Pregunten y consulten, usen foro, colaboren entre ustedes (**NO COPIEN**) los ayudantes están para apoyarlos
4. Trabajen con tiempo, no esperen a último momento para despejar dudas
5. Comiencen con una interfaz de usuario (UI) simple; no es necesario que su aplicación se vea "bonita" desde el comienzo
6. Para la primera entrega deberán crear maquetas o prototipos (*mockups*) de su aplicación. La idea es que las utilicen en las entregas siguientes así que preocúpense de diseñarlos adecuadamente. Los detalles de la primera tarea se entregarán en su enunciado específico
7. Siempre podrán, justificadamente, cambiar alguna regla que ustedes hayan definido para su juego o bien mejorar algún aspecto del juego, de la interfaz, etc.

Dudas

Para que todo el curso se vea beneficiado, hagan sus preguntas sobre las tareas, a través del foro del curso que se habilitará para tal efecto. **No se responderá ninguna duda de las tareas por e-mail.**