

75.42 Taller de programación I

Documentación de proyecto

Integrantes:

- Alvarez Windey, Juan - 95242
 - Aparicio, Rodrigo - 98967
 - Pinto, Tomás - 98757

Ayudante:

Ezequiel Werner

1. Organización	2
2. Tareas	2
3. Bugs conocidos	5

1. Organización

Utilizamos Trello como herramienta para el trackeo de tareas. Tenemos varios status para cada tarea: To Do, Doing, Review y Done. Cada uno de los integrantes se asigna una tarea y toma ownership para que la misma se complete. La manera en la que nos dividimos fue que Tomás se encargue del server, Juan del cliente, y Rodrigo de las tareas externas al server o cliente más específicas.

Para el versionado del código utilizamos Git con el repo hosteado en Github. El branching model elegido fue git flow, saliendo desde master para cada nuevo feature y realizando los pull requests desde dichos branches contra master. Para los pull requests configuramos que se requiera al menos una aprobación para poder mergear, y también al crear el PR se corre un chequeo de integración continua en Circle CI que verifica que el programa compile sin ningún error. Si falla este chequeo entonces no se puede mergear.

2. Tareas

A continuación mostraremos el estado de todas las tareas del proyecto. En particular nos quedaron dos estados: finalizadas y nice to have (salvo el deploy a AWS que continuamos intentando lograrlo). En Nice to have se encuentran algunos features extra los cuales no llegamos a implementar por falta de tiempo. En la columna de Members se pueden observar las iniciales de quien desarrolló dicha tarea.

List	Title	Members	Card #
Nice to have	(Cliente) Mostrar proyectiles		100
Nice to have	(Server) Vencimiento de drops		61
Nice to have	(Server) Balanceo del juego		53
Nice to have	(Server) distintos monstruos por mapa		63
Nice to have	(Server) Restringir cambio de sala según si el jugador fue atacado etc		70
Nice to have	(Server - Cliente) Monstruos suban de nivel y se hagan más fuertes		72

Nice to have	(Server - Protocol) Mover ids de notificaciones a su clase en vez del vector con datos		67
Nice to have	(Cliente - Server) Que el jugador tenga una password		73
Nice to have	Refactor para set direction		79
Nice to have	(Server) Refactor ArgentumGame		101
Nice to have	(Server - cliente) Validar que el nombre de usuario no se haya usado ya		103
Doing	Deploy en aws	RA	25
Done	(Server) Arreglo de crasheos en el server	TP	104
Done	(Cliente) Elección de mapa en el lobby	JW	102
Done	(Cliente) Lobby	JW	34
Done	Investigando crasheos cliente	TP	93
Done	(Cliente) Añadir la opción full-screen	JW	56
Done	(Tiled) Creación de nuevos mapas	JW	45
Done	(Server) Restringir cambio de sala por tiempo	TP	71
Done	Mapa hielo	TP	99
Done	(Documentacion) Modelo de objetos server	TP	95
Done	(Server) Perder experiencia al morir	TP	98
Done	(Documentacion) Diagrama de flujo - proyectiles: vida	TP	96
Done	(Documentacion) Diagrama de flujo - proyectiles: impacto a otro héroe	TP	97
Done	(Server) Inicializar personajes nuevos con ítems según su raza/clase	TP	92
Done	(Correccion - server) Movimiento por velocidad en vez de comandos	TP	83
Done	(Documentacion) Diagrama de flujo - movimiento de monstruos	TP	94
Done	(Correccion - server) Correr cppcheck	TP	84
Done	Suppresion file cliente	RA	24
Done	Configurar CI	RA	1
Done	Investigar msgpack	RA	31
Done	Establecer comunicación cliente/servidor	RA	10
Done	Protocolo binario	RA	17
Done	(Correccion - server) Inicialización de sockets	TP	88
Done	Server - varios	TP	26
Done	(Correccion - socket) Recv	TP	86
Done	(Cliente / Server) Cambio de la lógica del movimiento del hero. (corrección eze)	JW,TP	91
Done	(Correccion - server) Corregir warnings server	TP	82
Done	(Cliente) Chequear bug sonido. Múltiples sonidos por un mismo canal	JW	57

Done	(Cliente) Arreglar el deep copy de double buffer	JW	75
Done	(Cliente) Animación de meditar	JW	90
Done	(Correccion - server) Soportar crasheos de clientes	TP	87
Done	(Cliente) Cambiar la orientación del personaje cuando se quiere mover y algo lo bloquea	JW	59
Done	(Server) Lógica de fair play	TP	51
Done	(Server) Lógica y comando para que se pueda curar con flauta élfica	TP	50
Done	(Server) Lógica y comandos para resucitar	TP	48
Done	(Server) Comando meditar	TP	49
Done	(Server) Carga de mapas	RA	65
Done	(Server - Cliente) Envío y recepción de nombre de jugador, clase y raza a jugar	RA	55
Done	(Server) Mover constantes a archivo de configuración	RA	60
Done	(Server) Cambiar orientación aunque no se pueda mover	TP	78
Done	(Cliente) Añadiendo un banco y mercado interactivo y visible	JW	77
Done	(Cliente) Chat	JW	44
Done	(Server) Lógica y comandos de compra-venta	TP	47
Done	(Cliente) Renderizado de stats y de los distintos mensajes en el mini chat	JW	74
Done	(Server) Lógica y comando de banqueo	TP	52
Done	Persistencia archivo	RA	16
Done	(Server) Envío de mensajes privados o notificaciones como daño	TP	54
Done	(Cliente) Manejo de distintos mapas	JW	46
Done	(Cliente) Renderizar personaje muerto	JW	58
Done	Integración cambio de sala server-cliente	TP	66
Done	(Server) Refactor parseo mapas	TP	68
Done	(Server - Cliente) Comando para cambiar de sala	TP	62
Done	(Server - cliente) drop items	TP	64
Done	(Cliente) Mostrar spells mandados por el server	JW	42
Done	(Server) Drops	TP	37
Done	(Cliente) Render de drops	JW	41
Done	(Server) Proyectiles	TP	35
Done	(Cliente) Renderizado de equipamiento	JW	43
Done	(Cliente) Agregado de selector de items y envío al server de la selección	JW	39
Done	(Cliente) Agregado de hechizos	JW	36
Done	(Cliente) Agregando UI y items al inventario	JW	38
Done	(Cliente) Música	JW	20
Done	(Cliente - Protocol) Recepción del estado del juego	JW	29

Done	(Server - Protocol) Comandos sin ID del jugador que los manda	TP	33
Done	Analizando enunciado (Server)	TP	3
Done	(Cliente) Mejora threads y comunicación interna en el cliente	JW	32
Done	(Server) Lógica de movimiento de personajes y NPCs	TP	6
Done	(Server - Protocol) Envío de estado de juego	TP	27
Done	(Server) Lógica de ataque defensa. Razas y clases.	TP	14
Done	(Cliente) Aplicando double buffer y cambiando los hilos GameUpdater y GameRenderer	JW	15
Done	(Cliente) Agregado de Armas y armaduras	JW	28
Done	(Server) Estado del juego en JSON - posiciones	TP	13
Done	(Server) Salas	TP	4
Done	(Server) Carga de mapas	TP	5
Done	Agregar imagen del lobby	JW	2
Done	(Cliente) Mostrar personajes y su animación moviéndose en el mapa	JW	12
Done	Refactor texture, aplicando singleton para Paths	JW	18
Done	Configuración de cmake	JW	9
Done	(Cliente) Creando pantalla de log in en qt	JW	7
Done	(Cliente) Crear animaciones en SDL	JW	8
Done	Documentación primera entrega	RA	19
Done	Dockerfile server	RA	21
Done	Instalador y levantador server y client	RA	22
Done	Documentación segunda entrega	RA,TP	40
Done	(Correccion - socket) Corregir connect y bind	TP	85
Done	(Server) Envio de mas mensajes de eventos del juego	TP	89
Done	(Cliente) Implementación de los comandos restantes	JW	69
Done	Hacer video del juego corriendo	TP	81

3. Bugs conocidos

Dos cosas que detectamos son:

- Los monstruos siguen/atacan al héroe más cercano, pero si hay héroe que se conectó antes y esta un poco más lejos (dentro del rango del monstruo) va a targetear a ese antes que al más cercano.
- Si se ejecuta el server con un ups alto o bajo (< 40 / > 50), a veces ocurre que el cliente crashea.