

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

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, tanto las que están finalizadas, en progreso, y pendientes. En la columna de Members se pueden observar las iniciales de quien desarrolló o quien tiene a su cargo dicha tarea.

List	Title	Members
TODO	(Server) Mover constantes a archivo de configuracion	
TODO	(Tiled) Creacion de nuevos mapas ?	
TODO	(Server - Cliente) Manejo de distintos mapas	
TODO	(Server) Logica y comandos de compra-venta	
TODO	(Server) Logica y comandos para resucitar	
TODO	(Server) Comando meditar	
TODO	(Server) Logica y comando para que se pueda curar con flauta elfica	
TODO	(Server) Logica de fair play	
TODO	(Server) Logica y comando de banqueo	
TODO	Instalador	RA

TODO	Persistencia archivo	RA
TODO	Lobby	RA
TODO	Deploy en aws	RA
TODO	Documentación segunda entrega	RA
TODO	(Cliente) Chat	JW
TODO	(Server) Balanceo del juego	
TODO	(Server) Envio de mensajes privados o notificaciones como daño	
TODO	(Server) Recepcion de nombre de jugador, clase y raza a jugar	
TODO	(Cliente) Listado de opciones y todo el texto de la UI	JW
TODO	(Cliente) Renderizar personaje muerto	JW
TODO	(Cliente) Cambiar la orientación del personaje cuando se quiere mover y algo lo bloquea	JW
Doing	Documentación primera entrega	RA
Doing	Dockerfiles server y client	RA
Doing	(Server) Drops	TP
Review	(Cliente) Chequear bug sonido. Multiples sonidos por un mismo canal	JW
Done	Server - varios	TP
Done	(Cliente) Mostrar spells mandados por el server	JW
Done	(Cliente) Render de drops	JW
Done	(Server) Projectiles	TP
Done	(Cliente) Renderizado de equipamiento	JW
Done	(Cliente) Agregado de selector de items y envio al server de la selección	JW
Done	(Cliente) Agregado de hechizos	JW
Done	(Cliente) Agregando UI y items al inventario	JW
Done	(Cliente) Música	JW
Done	(Cliente - Protocol) Recepción del estado del juego	JW
Done	(Server - Protocol) Comandos sin ID del jugador que los manda	TP
Done	Analizando enunciado (Server)	TP
Done	(Cliente) Mejora threads y comunicación interna en el cliente	JW
Done	(Server) Logica de movimiento de personajes y NPCs	TP
Done	(Server - Protocol) Envio de estado de juego	TP
Done	(Server) Lógica de ataque defensa. Razas y clases.	TP
Done	(Cliente) Aplicando double buffer y cambiando los hilos GameUpdater y GameRenderer	JW
Done	(Cliente) Agregado de Armas y armaduras	JW
Done	(Server) Estado del juego en JSON - posiciones	TP

Done	(Server) Salas	TP
Done	(Server) Carga de mapas	TP
Done	Agregar imagen del lobby	JW
Done	(Cliente) Mostrar personajes y su animación moviendose en el mapa	JW
Done	Refactor texture, aplciando singleton para Paths	JW
Done	Configuración de cmake	JW
Done	(Cliente) Creando pantalla de log in en qt	JW
Done	(Cliente) Crear animaciones en SDL	JW
Done	Suppresion file cliente	RA
Done	Configurar CI	RA
Done	Investigar msgpack	RA
Done	Establecer comunicación cliente/servidor	RA
Done	Protocolo binario	RA