

Relazione sul progetto didattico del corso
“Tecnologie open-source”

Enrico Sartorello

17 giugno 2010

Indice

1	Introduzione	3
2	Visione	4
3	Mercato	5
4	Storia	6
5	Licenza	7
6	Business Model	8
7	Processo di sviluppo	10
8	Community	11
9	Information Management Tools	13
10	Development Management Tools	15
11	Bibliografia	16

1 Introduzione

Il digital entertainment è una realtà con esperienza oramai più che trentennale nel mondo informatico moderno.

Negli anni, corroborata da un mercato potenziale in continua espansione, abbiamo assistito ad una vera e propria evoluzione nel mondo dei videogiochi, sia dal punto di vista tecnologico/contenutistico sia considerando i business models adottati.

Una branca importante dell'intrattenimento digitale è rappresentata da quell'insieme di videogiochi denominati "First Person Shooter" (o FPS), caratterizzati dall'immergere il giocatore in un mondo virtuale nel quale, in prima persona, farsi largo a suon di spari e combattimenti. Gli FPS per loro stessa natura si affacciano ad un pubblico per la maggior parte maschile compreso in una fascia d'età abbastanza ampia.

Nel 1999 Id Software rilascia un videogioco sparatutto destinato a diventare una pietra miliare del genere: si tratta di *Quake 3 Arena* (noto anche come Quake 3, Q3 o Q3A). La terza versione dello sparatutto dell'azienda texana si discosta dalle release precedenti: si concentra su una modalità di gioco multiplayer e sul fornire un motore di gioco semplice ma snello e altamente configurabile.

Il successo di Q3 è notevole, e tutt'ora vanta migliaia di fan in tutto il mondo. A sei anni di distanza dal rilascio, il 19 Agosto 2005, Id Software decide di rilasciare pubblicamente il codice completo di Quake 3 Arena con licenza GPL ([1]). Di fatto il rilascio pubblico comprende *solo* il motore di gioco, mentre contenuti quali texture e modelli non vengono rilasciati (forzando quindi chi volesse giocare al gioco originale ad acquistarne una copia).

Il rilascio ha un notevole impatto nella comunità open-source: nascono *ioquake*, progetto che prende in carico il motore di Q3 con l'intento di migliorarlo, fare bugfixing e aggiungere funzionalità, ed *OpenArena*, che aggiunge a ioquake tutti i contenuti necessari a rendere fruibile il gioco.

2 Visione

Fino ai primi anni 2000, l'approccio classico dei videogiochi First Person Shooter era il fornire uno scenario umani-contro-umani, dove sostanzialmente i vari partecipanti al gioco esibivano caratteristiche pressoché identiche (eccezion fatta per il set di armamenti che poteva variare di partecipante in partecipante). Nel 2000 Tim Angus (noto alla comunità open-source con il soprannome "Timbo") decide di sfruttare la possibilità di estendere Quake 3 (il gioco infatti era fornito di un sistema per estendere le funzionalità mediante plugin/extension packs) per uscire dagli schemi tradizionali dell'epoca: la sua idea è di realizzare uno sparatutto nel quale si affrontano due razze totalmente differenti.

Angus immagina un gioco immerso in un ambiente futuristico dove si fronteggiano gli umani (con caratteristiche simili a quelle degli sparatutto tradizionali) e gli alieni. Questi ultimi, in particolare, sono la vera novità che renderà celebre *Tremulous*, il gioco nato dalla sua idea: per gli alieni infatti cambia completamente il *gameplay*, non si tratta più di un'interazione del tipo "mira-e-spara", ma piuttosto di un vero e proprio governare i loro movimenti (anche del tutto atipici, come il potersi arrampicare sui muri) per poter infliggere danni agli umani mediante graffi, scosse, iniezione di tossine.

In aggiunta a tutto questo, Angus arricchisce Tremulous con un aspetto strategico (sulle quali sfaccettature non vale la pena soffermarsi) che lo rende molto più complesso e interessante di un semplice sparatutto, ed introduce innovazioni grafiche di notevole rilievo.

Gli elementi di innovazione e distinzione rispetto ai prodotti preesistenti dunque sussistono e contribuiranno (come spiegato nel capitolo "Storia") al successo del prodotto.

Ciò a cui si assisterà successivamente, utilizzando il termine con il quale è titolato questo capitolo, è che il *core team* di Tremulous si trovi privo di una visione ad ampio raggio che ne consenta un'ulteriore espansione. Questo aspetto sarà analizzato con maggior ricchezza di dettagli nei capitoli che seguono.

Nel 2005 (quasi 5 anni dopo l'idea originale e 2 anni dopo l'inizio ufficiale dei lavori), pochi giorni prima del rilascio di Quake 3 sotto licenza GPL, Tremulous viene rilasciato come estensione del software originale, ed inizia la sua "vita pubblica" nel mondo open-source.

3 Mercato

E' difficile immaginare che il software analizzato, sin dalle origini, ambisse all'annessione di specifiche quote di mercato: Tremulous nasce come progetto realizzato, senza fini di lucro, da un gruppo di appassionati che intendono migliorare la loro esperienza di gioco su Q3.

Ciò che si è verificato a posteriori è che il prodotto ha riscosso un notevole successo tra i giocatori che già ben conoscevano Q3 (un pubblico quindi orientato all'intrattenimento digitale mediante sparatutto), ed altresì ha avuto notevoli apprezzamenti, più in generale, pure tra gli appassionati di GNU/Linux e del mondo open-source.

Come sarà spiegato più precisamente nel capitolo seguente, Tremulous, insieme ad altri validi videogiochi open-source quali Tux Racer, Frozen Bubble, Cube, SuperTuxKart, ha dato un impulso molto importante al mondo open-source, dimostrando che GNU/Linux può essere un sistema operativo che offre videogiochi di buon livello e fornendo una notevole attrattiva al mondo del digital entertainment.

4 Storia

Come anticipato nei capitoli precedenti, Tremulous viene ideato nell'anno 2000. Nei primi tre anni di incubazione dell'idea, il suo creatore Tim Angus organizza un piccolo team di sviluppo composto da circa una decina di sviluppatori, e nel 2003 inizia la realizzazione vera e propria del software.

Nell'Agosto 2005 Angus annuncia il rilascio della prima versione ufficiale (1.0.0) del videogame, come *mod* di Quake 3 Arena: la reazione del pubblico è tiepida, Quake 3 è un prodotto relativamente vecchio e molte *mod* di successo sono state già pubblicate negli anni precedenti.

Qualche settimana dopo però, il rilascio open-source di Q3 apre una nuova prospettiva: il team di sviluppo decide di concentrare i propri sforzi nell'inglobare ioquake e fornire una versione standalone completamente free e open-source.

Il rilascio della versione standalone 1.1.0 è datato fine Marzo 2006 ed è un vero successo: è il primo derivato di Q3 totalmente open-source e standalone e la comunità di videogiocatori lo accoglie con notevole interesse.

In pochi mesi la release 1.1.0 di Tremulous raggiunge la quota di 1'000'000 di download, con una media di 400 giocatori attivi in ogni istante. In pochissimo tempo fioriscono una serie di comunità satellite (spesso distinguibili dal paese d'origine) con un numero crescente di membri.

Sempre nel 2006 il videogioco riceve una serie di importanti riconoscimenti nel mondo dei videogiochi open-souce e soprattutto, nasce una web-radio in suo onore e centinaia di utenti si sbizzarriscono ad arricchirlo di nuove mappe.

Per due anni circa (fino a metà 2008), Tremulous continua ad espandersi, seppur rallentando rispetto ai ritmi frenetici successivi al rilascio della versione standalone. Il 2009 rappresenta l'anno nel quale questa tendenza si inverte: la maggior parte degli appassionati comincia a sentire la necessità di nuove release ed aggiornamenti che tardano ad arrivare.

Inoltre, il rapporto tra core team e community (come sarà spiegato nel dettaglio nel capitolo relativo) favorisce la diffusione di un sentimento di disaffezione nei confronti degli sviluppatori: nascono quindi una serie di progetti paralleli a Tremulous che ne ereditano il codice e tentano di assecondare le necessità espresse dai giocatori. Nonostante idee molto interessanti ed uno sforzo di sviluppo apprezzabile, nessuno di questi progetti riesce a diffondersi e buona parte di essi muore nell'arco di un anno. Il sottoscritto ha partecipato ad una di queste iniziative, denominata "Duck'n Cat" [2], come sviluppatore in linguaggio C e beta-tester.

A fine 2009 (dopo circa 4 anni dalla prima e unica release) gli sviluppatori ufficiali rilasciano una nuova versione beta, che però tradisce le aspettative della comunità concentrandosi su aspetti relativi al bilanciamento del gioco tralasciando innovazioni dal punto di vista grafico o strutturale.

Allo stato attuale non paiono esserci segnali di ripresa di interesse del videogioco da parte del pubblico; la comunità, anche nei suoi adepti di maggior spicco, appare in netto decadimento.

5 Licenza

Il primo rilascio di Tremulous avviene sottoforma di mod/espansione per un prodotto closed-source (Q3). Nella fattispecie, Q3 fornisce una virtual machine alla quale è possibile fornire dei moduli da interpretare.

Considerando *linking* l'interazione descritta, con mutua interazione tramite metodi e condivisione di strutture dati, Angus si vede impossibilitato nel pubblicare il codice sotto licenza GPL e lo rilascia sotto una licenza ad-hoc che denomina "Open-Source Mod Licence" (OSML). La licenza riceve alcune critiche in quanto la sua stesura appare poco curata e per certi versi poco chiara, ma sino ad allora non esistevano mod rilasciate open-source e quindi il problema non era mai stato affrontato.

Accettando i termini d'uso di OSML 1.0, l'utente è libero di ispezionare, modificare e redistribuire il codice della mod; la distribuzione è però possibile solo senza scopi commerciali, il che non rende il prodotto open-source in senso stretto. Inoltre, una particolare enfasi viene posta nel sollevare gli autori della mod da eventuali danni che possono essere causati al software o eventuali problemi nell'uso della stessa in un contesto di gioco online.

Piuttosto che impiegare tempo e sforzi nella stesura di una licenza che risulta poco chiara e non esaustiva, sarebbe stato molto più semplice adottare una licenza GPL-compatibile come LGPL (che nella fattispecie permette il linking con moduli non-free) oppure aggiungere un'eccezione alla licenza GPL affinché consentisse di linkare il codice ad un programma principale non-free.

Nel 2005, il rilascio del motore di Quake 3 con licenza GPL semplifica la situazione: per quanto riguarda il codice, Tremulous diventa un prodotto standalone rilasciato sotto licenza GPL, e al suo interno incorpora pacchetti esterni rilasciati con licenze GPL-compatibili.

Per quanto riguarda i *media* che compongono il gioco (sostanzialmente texture, suoni e modelli 3D), essi hanno una licenza Creative Commons Attribution-ShareAlike 2.5 che basilamente consente di condividere e modificare le opere considerate con l'obbligo però di specificare gli autori delle opere originali e di mantenere lo stesso schema di licenza.

Questa doppia licenza ha spinto un buon numero di sviluppatori, per lo più appassionati, a realizzare delle versioni alternative e non ufficiali di Tremulous o a fornire delle piccole modifiche concentrate su determinati aspetti (ad esempio lato server, nell'interfaccia utente, nei modelli 3D).

L'adozione di GPL, seppur scontata, ha consentito di diffondere il gioco in modo capillare all'interno di svariati canali di comunicazione del mondo open-source, quali distribuzioni, repository di pacchetti, siti e riviste di interesse.

6 Business Model

Tremulous nasce come sforzo di un piccolo gruppo di appassionati interessati unicamente a migliorare la propria esperienza di gioco e a consentire ad altri di fare altrettanto.

In conseguenza di questo, il progetto nasce e permane privo di un vero business model, ad eccezione del fatto che gli sviluppatori vengono supportati mediante un sistema di donazioni. Recentemente, inoltre, è stato introdotto un piccolo negozio on-line di gadget (magliette, adesivi) con l'obiettivo di raccogliere ulteriori fondi.

Secondo chi scrive, il mancato tentativo di instaurare un business model serio è una delle cause della decadenza del progetto: riuscire a trovare dei fondi significa poter incentivare chi sviluppa e quindi riuscire a rilasciare nuove versioni in un lasso temporale accettabile.

Dal canto suo, intorno al 2007 Tremulous aveva tutte le carte in regola per fare il salto di qualità: una comunità ampia e leale, colma di iniziative e di interesse verso il progetto.

Il bacino di utenza era notevole; l'unico sforzo da porre era la necessità di trovare dei meccanismi che vincessero l'inerzia dell'utente medio di software open-source abituato a non spendere di tasca propria.

Probabilmente, il rendere più chiara e più pubblicizzata la necessità di supportare il gioco tramite le donazioni avrebbe portato, di per sé, ad un aumento dei fondi raccolti. Inoltre, è possibile immaginare una serie di misure che avrebbero potuto migliorare ulteriormente la situazione:

- avviando una politica di sviluppo incentrata in rilasci frequenti, è ipotizzabile l'istituzione di una forma di abbonamento a pagamento che consenta agli abbonati di ottenere le release più recenti in anteprima rispetto al pubblico.
Questo modello di business è già noto e diffuso sia nell'ambito dei videogiochi sia nell'ambito del mondo open-source.
- istituire un sistema di merchandising più robusto e pubblicizzato può portare ad entrate cospicue. Un esempio da seguire in questo senso è Apache Foundation, che ha nel merchandising una delle fonti primarie di sostentamento economico.
- ipotizzare la richiesta di una piccola fee da pagare annualmente per poter giocare nei server ufficiali;
- alcuni server di Tremulous gestiti autonomamente sono sponsorizzati o utilizzati per introdurre delle piccole pubblicità (generalmente relative a siti di e-commerce per videogiocatori): il fenomeno può essere sfruttato introducendo una seconda licenza che obblighi il versamento di una tassa agli sviluppatori qualora utilizzino i server di Tremulous a fini pubblicitari.
- l'organizzazione di tornei ufficiali, dove l'iscrizione di squadre preveda il versamento di una piccola quota di partecipazione con il fine esplicito di finanziare lo sviluppo del gioco.

I margini per tentare un qualche tipo di approccio sono presenti, ma nulla del genere è mai stato concretamente analizzato; in realtà, il mancato interesse verso

la fioritura di un lato business del progetto è da considerarsi in un contesto più ampio, ovvero nell'approccio che gli sviluppatori ufficiali hanno avuto, sin dall'inizio, nei confronti della natura del progetto e nei rapporti con la comunità. Questo aspetto sarà approfondito nel seguito, nel capitolo "Community".

7 Processo di sviluppo

Descrivere con correttezza e precisione quale sia la politica che regola lo sviluppo di Tremulous è pressoché impossibile, poiché lo sviluppo è condotto senza trasparenza ed è necessario ricondursi ai report forniti dagli strumenti utilizzati dal team di sviluppo per dedurre il *modus operandi*.

La natura del progetto è stata ormai ampiamente descritta, ma per comprendere le modalità di sviluppo è necessario ricordare che si tratta sostanzialmente di volontari che contribuiscono per passione.

Attualmente non è possibile asserire che lo sviluppo di Tremulous aderisca ad un qualche modello di sviluppo; inoltre, gli sviluppatori non hanno vincoli temporali formali da rispettare e ciò rende impossibile ipotizzare una qualche forma di pianificazione concreta.

Ciò che si sa è che lo sviluppo è "politicamente" governato dal fondatore Tim Angus, il quale decide verso quale direzione debba andare lo sviluppo e come ripartire agli sviluppatori i vari task individuati.

Il team di sviluppo si regola mediante una mailing-list privata; la codifica non segue code-convention specifiche, ma semplicemente delle regole (non obbligatorie) che specificano i criteri di spaziatura del codice e le modalità di scrittura dei commenti.

Il commit è consentito solo per codice considerato privo di anomalie da parte dell'autore. Il testing del software è per lo più manuale, non regolamentato, e prevede cicli di uso del software che ricreino gli scenari d'uso tipici.

Il rilascio pubblico di una nuova versione avviene solo quando una serie di requisiti (tipicamente molto ampia) sono stati soddisfatti totalmente, ed il software è stato testato per un lasso di tempo sufficiente da un gruppo di videogiocatori-tester senza riportare anomalie.

L'analisi stilata descrive perfettamente i vari motivi per i quali i rilasci di Tremulous nel tempo siano stati pochi e distanziati di anni l'uno dall'altro.

8 Community

L'approccio alla community rappresenta probabilmente il punto più controverso dell'intero progetto, e a detta di chi scrive probabilmente anche il motivo per il quale esso non ha avuto un seguito nonostante le sfavillanti premesse.

Sintetizzando, si può tranquillamente affermare che il core team di Tremulous abbia sempre considerato la community come un corpo estraneo privo di valore e non meritevole di particolare interesse. Il motivo di questo atteggiamento è da ricercarsi probabilmente in aspetti caratteriali dei principali promotori del progetto e in una mancanza di interesse, visione, prospettiva ed ambizione.

Poiché sarebbe fuorviante descrivere le caratteristiche delle attività attuali del core team, in quanto come anticipato negli ultimi anni il progetto ha subito un certo grado di involuzione, tale descrizione farà riferimento perlopiù agli anni 2005-2008, intervallo temporale nel quale il progetto può essere considerato vivo a tutti gli effetti.

La leadership di Tremulous può vantare un notevole numero di qualità che hanno valore aggiunto all'interno di un progetto open-source: in primis, Tim Angus ed i due collaboratori più stretti che "comandano" il progetto hanno spiccate caratteristiche da leader, hanno forza, personalità ed autorità. Inoltre, il tempo che dedicano al progetto è senza dubbio notevole e superiore a qualsiasi contributor esterno.

Altro aspetto importante è che si tratta di programmatori di grande esperienza per quanto concerne le specificità del progetto: conoscono molto bene il linguaggio C, la programmazione mediante OpenGL, la programmazione e il funzionamento del motore di Quake 3 Arena (Angus ha dei precedenti come collaboratore in altri progetti che hanno utilizzato/esteso il motore di Q3A).

Negli anni, la presenza del core team nei forum e mailing list di riferimento è sempre stata costante; ciononostante, molti sono i limiti del team per quanto concerne il rapporto con il mondo esterno.

Entrando nel dettaglio, si è notato nel tempo che il core team non ha mai avuto grandi attitudini nell'aiutare i nuovi arrivati o nel prestare particolare pazienza nei loro confronti. Una considerazione analoga vale per quanto riguarda il grado di apertura del team nei confronti delle idee proposte da persone esterne al nucleo; come detto, Tremulous ha avuto un grande riscontro, e ciò si è tradotto in un notevole numero di interventi da parte di persone, più o meno qualificate, nel suggerire miglioramenti, correzioni, bilanciamenti, estensioni al software.

Spesso le argomentazioni portate erano supportate da studi preliminari documentati o addirittura espressione di interesse sub-comunità locali che raccoglievano migliaia di appassionati. Nonostante il valore degli interventi, la politica del core team è sempre stata quella di rigettare la stragrande maggioranza dei suggerimenti a priori, con un atteggiamento spesso ostile. Di certo questo tipo di comportamento non ha aiutato la formazione di una comunità fedele e produttiva.

Il codice sorgente iniziale è stato certamente un artefatto importante e di notevole qualità: lo sforzo iniziale del team di sviluppo è stato grande nel fornire del codice con un buon grado di modularità, un tool di installazione adeguato ad un pubblico generalista, delle specifiche chiare sulle piattaforme supportate. Un impegno non indifferente è stata pure la promozione del progetto nei canali di comunicazione dedicati agli appassionati di videogames e a quelli relativi al mondo open-source.

Negli anni considerati molte persone hanno espresso la loro volontà a contribuire allo sviluppo del codice sorgente del progetto, ma anche in questo caso l'atteggiamento del core team è stato abbastanza discutibile: non sono mai stati accettati contributi esterni, neppure previa moderazione. L'ingresso di nuovi componenti nel core team non segue una politica chiara, è stato raro ed ha interessato solo persone che hanno dato contributi molto forti al progetto.

Questa sostanziale chiusura nei confronti di potenziali contributi ha fatto nascere una serie di iniziative che hanno cercato di catalizzare nuove forze per sopprimere all'assenza e alla chiusura degli sviluppatori ufficiali: tali iniziative, snobbate dal core team, per quanto valide e meritevoli hanno finito col disperdere gli sforzi e con lo spaccare parzialmente la comunità, anche se buona parte di essa è rimasta fedele al ramo ufficiale.

Caratteristica principale di queste iniziative era quella di accentuare il proprio grado di trasparenza e partecipazione nel progetto, ad ulteriore conferma che l'eccessiva chiusura da parte del team ufficiale è stata vissuta come una scelta difficilmente comprensibile e fortemente disagiata.

Vedendosi vanificare i propri sforzi molti individui hanno infine deciso di dedicarsi ad altri progetti.

Concludendo lo studio relativo alla comunità, è possibile quindi asserire che questo aspetto rappresenta probabilmente il punto più negativo dell'intero progetto: senza incorrere in un impegno troppo eccessivo per il core team, un atteggiamento più benevolo nei confronti dei contributi esterni avrebbe certamente giovato al progetto e probabilmente garantito ad esso una maggiore longevità.

9 Information Management Tools

Tremulous ha un proprio sito ufficiale [4] nel quale è presente un numero notevole di risorse relative al progetto.

In primo luogo, è possibile scaricare il software sia in formato binario (per Windows, GNU/Linux e MacOSX) sia nel suo codice sorgente; il sito fornisce inoltre un primo riferimento dove sono riportate le notizie più importanti del progetto, un manuale d'installazione e d'uso e ulteriori informazioni utili quali una pagina di F.A.Q., specifiche della licenza, una sezione che elenca i contributor del progetto.

Sono presenti inoltre due mailing list:

- una mailing list privata, utilizzata dagli sviluppatori accreditati per coordinare lo sviluppo e scambiarsi idee, opinioni, aiuto. Tale mailing list non risulta accessibile né consultabile pubblicamente.
- una mailing list pubblica, aperta a chiunque voglia parteciparvi, per discutere idee e opinioni sul gioco e soprattutto incentrata sul testing funzionale del software e l'eventuale reporting di anomalie.

Entrambe le mailing list utilizzano una piattaforma diffusa per la loro gestione, che consente di usufruire delle più comuni funzionalità: l'iscrizione via email o via web, la possibilità di ricevere le email in digest, attività di moderazione e di archiviazione dei contenuti.

Il progetto adotta un sistema di versionamento mediante SVN. Il repository SVN viene utilizzato unicamente per il versionamento del codice sorgente, mentre la documentazione tramite Wiki adotta un sistema di versionamento autonomo (ma tuttavia molto semplice e intuitivo).

L'uso di SVN appare adeguato e corretto; non sono presenti nel repository file generati automaticamente. Il repository è navigabile nel Web [5], dove è possibile visualizzare i singoli commit e la cronologia di modifica dei vari file con un livello di dettaglio molto buono.

Ogniquale è stato necessario, si è provveduto a creare dei branch all'interno del repository che sono stati poi reinseriti nel trunk a tempo debito. L'uso di branch, ad esempio, è stato adottato prima di ogni release con l'intento di stabilizzarla prima del rilascio ufficiale.

Tremulous adotta un sistema di bug tracking molto noto e diffuso: Bugzilla, incorporato nella Mozilla Foundation. L'adozione di questo strumento per il bug tracking web-based [6] consente agli sviluppatori e agli utenti esterni di segnalare bug o richiedere feature, e fornisce un punto vero di raccordo tra gli sviluppatori e la community.

Lo strumento è integrato con il repository SVN di modo che sia tracciabile la corrispondenza tra issue e modifiche al codice relative. Ogni attività relativa alle issue inoltre risulta loggata e visualizzabile.

Il progetto dispone di un canale IRC (#tremulous su FreeNode) nel quale sono spesso disponibili in tempo reale i membri del core team; questo sicuramente è uno strumento che avvicina gli utenti al team di sviluppo, anche se talvolta la disponibilità a rispondere e la qualità delle risposte nel canale dipende da quali membri siano disponibili in quel momento. Il canale resta comunque un punto di riferimento per lo scambio di opinioni tra gli utilizzatori del software e quindi come punto virtuale di aggregazione.

Un'altra forma di comunicazione valida è rappresentata dal forum, che raccoglie interventi su tanti diversi aspetti del prodotto e vanta un buon grado di partecipazione da parte del core team.

A partire dalla homepage di Tremulous è possibile accedere ad un'area Wiki che rappresenta una documentazione approfondita del gioco in molte sue parti: gli argomenti trattati variano da una descrizione di carattere generale dello scenario di gioco, alle tattiche dei vari personaggi, a argomenti di carattere tecnico quali tutorial per lo sviluppo di nuove mappe o per la personalizzazione dell'interfaccia utente mediante script.

Seppur tardivo rispetto al rilascio della versione 1.0.0, lo sforzo di documentazione in Wiki è stato notevole ed è figlio soprattutto di iniziative esterne al core team. Notevole è l'incentivo a partecipare alla stesura di pagine della Wiki in quanto mancano tutt'ora molte parti importanti.

Per quanto concerne questo strumento, è necessario sollevare una critica: le pagine non aderiscono a standard editoriali condivisi, ma soprattutto a livello di interfaccia manca una minimappa o un semplice breadcrumb che contestualizzi la pagina correntemente visualizzata all'interno della struttura della Wiki.

Nonostante l'area Wiki, bisogna affermare che lo sforzo documentativo da parte del core team è sempre stato notevole e, da questo punto di vista, la loro attività è stata lodevole: sin dalla prima release è stato fornito un manuale ben scritto ed esaustivo per chiunque volesse utilizzare il software. Lo stile di scrittura appare buono ed adeguato all'audience (generalista) del prodotto.

10 Development Management Tools

Il progetto presenta sicuramente dei limiti per quanto riguarda l'adozione di strumenti che consentano una gestione agevole dello sviluppo.

Per quanto riguarda la *build automation*, si utilizza Make per l'automatizzazione della compilazione del prodotto. La scelta è sicuramente adeguata poiché la procedura di compilazione di Tremulous è alquanto complicata dal fatto che il codice sorgente è presente in quantità considerevole, ed inoltre in fase di compilazione devono essere incluse molte librerie esterne (quali ad esempio LibCurl, ZLib, LibSpeex). La compilazione manuale in questo caso risulterebbe tediosa e scoraggerebbe potenziali sviluppatori a lavorare al progetto.

Non sono presenti tool per la progettazione e per la generazione automatica di codice, scelta corretta poiché non sono individuabili use-case per i quali sia necessario dotarsi di simili tecnologie.

Il vero punto di rilievo è legata alla totale mancanza di strumenti per la *quality assurance*. Questo aspetto è certamente negativo ed ha conseguenze ben note nell'ambito del ciclo di vita del software, ma può forse essere giustificato tenendo a mente alcune cose:

- Il team di sviluppo è molto ridotto, il software è relativamente "piccolo" ed estremamente modularizzato. Queste tre precondizioni rendono l'inserimento di un'anomalia più difficile e la scoperta e risoluzione molto più semplici e immediate.
Inoltre, c'è coscienza del fatto che l'inserimento di bug nel codice può creare notevoli problematiche, motivo per il quale gli sviluppatori sono chiamati a fare delle modifiche che siano fortemente ponderate anche da questo punto di vista.
- Sin dai primi mesi di sviluppo, attorno al core team si è creato un piccolo (non più di circa 30 persone) ma fedele gruppo di beta tester che testano le modifiche che via via vengono apportate al programma. Le sessioni di testing (che si traducono in sessioni di gioco) sono abbastanza frequenti e consentono di individuare tempestivamente eventuali anomalie.
Al fine di allargare ulteriormente il fronte di testing, è stato istituito il sistema di bug tracking descritto nel capitolo precedente.
- Il prodotto in questione esce dai canoni tradizionali del software: si tratta di un videogioco, ed in quanto tale ha numerosi aspetti che difficilmente possono essere testati senza l'interazione umana. Si pensi al rendering dei modelli tridimensionali, a aspetti di gameplay, a fattori di bilanciamento tra i vari giocatori, alla gestione dell'economia di gioco in presenza di problemi di connettività: sono solo alcuni esempi di problematiche per le quali è quasi impossibile concepire metodi di analisi e testing automatico poiché hanno a che fare con le aspettative e la percezione degli utenti.

D'altro canto, è noto che l'uso di adeguati strumenti per la quality assurance favorisce la pratica di rilasci frequenti, dunque un'adozione in questo senso avrebbe probabilmente giovato ad uno degli aspetti più critici del progetto, la lentezza dei rilasci ufficiali.

11 Bibliografia

1. Slashdot annuncia il rilascio sotto licenza GPL del codice di Quake3, <http://games.slashdot.org/article.pl?sid=05/08/20/1329236&tid=112>
2. Sito ufficiale del progetto "Duck'n Cat", <https://dnc.tremforges.net>
3. Creative Commons Attribution-ShareAlike 2.5 License, <http://creativecommons.org/licenses/by-sa/2.5/>
4. Sito ufficiale di Tremulous, <http://tremulous.net>
5. Sito web del repository SVN ufficiale di Tremulous, <http://svn.icculus.org/tremulous>
6. Bug tracking system di Tremulous, https://bugzilla.icculus.org/buglist.cgi?query_format=advanced&product=Tremulous