

# Protokoll des Orxonox Think Tanks vom 9. Juni 2012

## Anwesende:

- Benjamin De Capitani
- Benjamin Knecht
- Damian Frick
- Fabian Landau
- Mauro Frischherz
- Johannes Ritz
- Sandro Merkli

Der Think Tank fand am Samstag 9. Juni 2012 in der Wohnung von Johannes statt und dauerte knapp fünf Stunden von 18 bis 23 Uhr, inklusive leckerem Essen mit Pizza und Kuchen (danke Jo).

Die Traktanden lassen sich grob in zwei Themenbereiche gliedern: organisatorische Themen (Server, PPS, Organisation, Release) und inhaltliche Themen (aktueller Stand und künftige Features). Der Fokus lag klar auf der Organisation der Entwicklung und des PPS. Im Folgenden sollen die wichtigsten Gesprächspunkte in thematischer (nicht chronologischer) Reihenfolge aufgeführt werden.

## Server

- Server: Die Organisation des Servers sowie seine Stabilität bzw. Erreichbarkeit wurden kritisiert. Der Server ist häufig offline, ausserdem hat die Wiki lange Ladezeiten. Damian hat beim VSETH einen weiteren Server beantragt, um den Server neu aufzusetzen. Falls strukturelle Probleme mit VSETH Servern bestehen, etwa was die instabile Netzverbindung angeht, wurde ein Wechsel zu einem anderen Anbieter vorgeschlagen. Es besteht auch die Bereitschaft, gemeinsam einen Server zu mieten. Wie der Server im Detail neu aufgesetzt wird, ist noch offen und könnte z.B. im Forum diskutiert werden.
- Backups: Es muss geklärt werden, ob es für unseren Server ein vernünftiges Backup System gibt.
- Wiki: Zusammen mit der Neuorganisation des Servers soll auch der lang geplante Umzug der Wiki auf ein besseres System vollzogen werden. Zur Zeit ist TikiWiki ein heisser Kandidat, nachdem Felix bereits vor knapp einem Jahr eine Testinstallation gemacht hat und wir TikiWiki auch bei Ogre erfolgreich im Einsatz sehen.
- Dokumentation: Die Dokumentation des Codes soll wenn möglich ausschliesslich in DoxyGen geschehen und nicht in der Wiki.
- Git: Es wurde einmal mehr der Vorschlag gemacht, von SVN auf Git umzusteigen. Bisher überwiegen Bedenken, dass Git für die PPS-Studenten zu komplex sei und dass die Grösse des Data-Repositories mit seinen grossen Binärfiles zu Unannehmlichkeiten führt. Laut Mauro gibt es aber, zumindest was das Data-Repository angeht, verschiedene Lösungsansätze, etwa dass auf den Tardis Rechnern nur ein kleiner Teil des Repositories ausgecheckt werden muss, oder dass unnötige grosse Commits nachträglich gelöscht werden können. Man hat sich darauf geeinigt, einen Testversuch zu wagen, bei dem zwar weiterhin SVN verwendet wird, man aber bereits auf Git diverse Use-Cases durchspielen kann. Sollte sich Git bewähren, würde man definitiv umsteigen. Allerdings wurde diesem Vorhaben eine geringe Priorität zugeteilt.

## PPS

- Im Rückblick auf das vergangene Semester wurde festgestellt, dass die Betreuung von 16 Studenten mit drei Assistenten stark grenzwertig ist und dies auch nur möglich war, weil ein Teil der Studenten mit Musik, Forum, Content oder Gruppenarbeiten beschäftigt waren, was den Betreuungsaufwand reduziert hat.
- Gleichzeitig wurde aber auch bemängelt, dass die Qualität einzelner Arbeiten teilweise etwas unter den Erwartungen liegt (nicht nur in diesem Semester, sondern allgemein). Im Folgenden wurden verschiedene Massnahmen besprochen, die dem entgegenwirken sollen.
  - Es wurde mehr Disziplin von den Studenten verlangt, etwa was das regelmässige Committed oder das Befolgen von Style-Guides anbelangt. Es wurde aber auch bezweifelt, dass dies überhaupt eine Wirkung zeigt.
  - Anwesenheit im PPS sollte Pflicht sein. In Ausnahmefällen kann zu Hause gearbeitet werden, aber dann muss Anwesenheit bei Reviews und der Präsentation verlangt werden.
  - Es wurde auch diskutiert, ob es klare Bedingungen für das Erhalten der PPS-Punkte geben soll (Anwesenheit, Qualität, Präsentation, etc).
  - Es wurde vorgeschlagen, dass die Studenten in den ersten 1-2 Wochen nach dem Verteilen der Projekte ein grobes Design und einen Zeitplan erstellen. Dies soll verhindern, dass Studenten den Aufwand unterschätzen und man eine klare Abfolge der einzelnen Punkte hat, zwischen denen auch Reviews stattfinden können.
  - Teamprojekte wurden unterschiedlich bewertet: Einerseits fördern sie Teamwork und verursachen weniger Betreuungsaufwand, andererseits ist der Output deutlich kleiner als wenn die Studenten alleine an einem Projekt arbeiten. Teamprojekte, bei denen ein Student ein Model macht und der andere den Code schreibt, wurden jedoch einvernehmlich positiv bewertet.
  - Es wurde diskutiert, ob die PPS-Zeit von drei auf vier Stunden erhöht werden soll, was bei drei Stunden am Nachmittag noch Raum für eine Stunde Hausaufgaben bieten würde. Es wurde aber bezweifelt, dass dies bei unmotivierten Studenten etwas bringen würde, während motivierte Studenten ohnehin mehr als das Minimum leisten.
  - Es wurde stattdessen vorgeschlagen, Studenten dadurch zu motivieren, indem man ihnen klar sagt, dass ihre Arbeit nur weiterverwendet werden kann, wenn sie in einem vernünftigen Zustand hinterlassen wird (d.h. entweder fertig oder aber zumindest sauber aufgeräumt und gut dokumentiert).
  - Am Anfang des PPS könnte man im Tutorial einen kleinen Test einbauen (analog zu kleinen Programmieraufgaben bei Bewerbungsgesprächen), um ein Gefühl für das Talent der Studenten zu erhalten. Dadurch soll verhindert werden, dass Studenten, die sich selber überschätzen, komplizierte Projekte auswählen.
- Da es trotz allen Massnahmen immer Studenten geben wird, die schlechte oder unvollständige Arbeit hinterlassen, wurde beschlossen, das PPS neu zu interpretieren: Das PPS soll eine Art „Lab“ sein, in dem Studenten einen Prototypen für ein neues Feature basteln können. Es wird nicht mehr garantiert, dass die Arbeit in den Trunk übernommen wird (allenfalls in einen temporären Branch für die Präsentation). Stattdessen wird nach dem Semester ein „Gremium“ von Stammentwicklern die Arbeiten reviewen und das weitere Vorgehen bestimmen:
  - Abgeschlossene Arbeiten können 1:1 in den Trunk gemerged werden.
  - Beinahe abgeschlossene Arbeiten können aufgeräumt und dann gemerged werden.
  - Unvollständige, aber sauber hinterlassene Arbeiten können einem neuen Studenten zugewiesen werden.
  - Alle übrigen Arbeiten, die noch einen nennenswerten Aufwand benötigen, werden

entweder verworfen (bzw. ins Archiv verschoben) oder aber von einem Stammentwickler übernommen, der es dann zu Ende führt (oder zumindest in einen Zustand bringt, in dem es von einem neuen Studenten übernommen werden kann).

Durch diese Massnahme soll verhindert werden, dass der Trunk mit unvollständigen Arbeiten, Hacks und verwaistem Code zugemüllt wird. Gleichzeitig soll dadurch den Studenten auch klar werden, dass ein gewisser Aufwand nötig ist, wenn ihr Projekt irgendwann in das endgültige Spiel übernommen werden soll.

- Für die Assistenten wurde vorgeschlagen, die Kommunikation mit den Studenten zu verbessern (z.B. regelmässig compillieren und committen) und auch untereinander eine Art Vorbesprechung oder Nachbesprechung des PPS durchzuführen, damit man immer ungefähr weiss, wie die Studenten vorankommen.
- Für das nächste Semester werden voraussichtlich Jo, Sandro und Mauro die Betreuung des PPS übernehmen. Von ihnen kann aber keiner die Betreuung von Modeling Arbeiten übernehmen. Beni (D.C.) wäre am Anfang des Semesters noch für eine Einleitung verfügbar. Über Kevin ist nichts Genaues bekannt. Es wurden Vorschläge für neue Assistenten im Bereich Content Creation gemacht (Namen vertraulich), es ist aber unklar, ob für das Herbstsemester überhaupt vier Assistenten benötigt werden.
- Fabian will während dem PPS über Skype (oder einen anderen Kommunikationskanal) erreichbar sein, um Fragen beantworten zu können.

## Organisation

- Die Kommunikation zwischen den Entwicklern soll verbessert werden. Dazu sollen einerseits wieder häufiger Think Tanks durchgeführt werden, andererseits wurde vorgeschlagen, eine Art Workshop zu machen, bei denen jeder Entwickler seine Arbeit präsentiert und z.B. auch eine Einführung in neue Features oder Entwicklungswerkzeuge geben kann.
- Es gibt viele kleinere und grössere Refactorings im Code, etwa den Umbau der XML-Anbindung oder den NewHumanController, die noch anstehen und die im Konflikt mit dem Semester-Rhythmus des PPS stehen (da die Studenten auf diesen Teilen des Codes aufbauen, kann man sie schlecht während einem Semester ändern, die Ferien sind aber meistens zu kurz um wesentliche Änderungen durchzuführen). Diese Situation wird etwas entschärft, indem nicht mehr alle PPS-Projekte in den Trunk gemerged werden. Andererseits kann es aber auch bedeuten, dass wir verwaisten Code (den keiner von uns kennt) löschen müssen, da der Aufwand zu gross wäre, diesen Code immer wieder anzupassen, wenn sich z.B. etwas im Framework ändert. Das genaue Vorgehen bei grösseren Änderungen muss von Fall zu Fall besprochen werden.
- Die Organisation des Data Repositories und des Data Pools muss verbessert (bzw. zu Ende geführt) werden. Damian hat sich bereit erklärt, diese Arbeit zu übernehmen.

## Release

- Wie wir festgestellt haben, sind eigentlich alle wesentlichen Punkte für ein Release erfüllt, daher wurde beschlossen, den Release endlich anzupacken.
- Für den Release wird sowohl für Code als auch Data ein Branch erstellt (bevorzugt nach dem Aufräumen und Mergen des aktuellen presentation Branches) und alle Levels und Features gelöscht, die wir nicht offiziell unterstützen wollen.
- Das Data Repository wird auf die notwendigen Files für die ausgewählten Levels reduziert.
- Für diese Files muss die Lizenzfrage geklärt sein.
- Ein Thread im Forum soll die Diskussion über das weitere Vorgehen für den Release

ermöglichen, z.B. welche Levels und Features wir behalten und welche wir entfernen wollen.

## Technisches

- Der Support von Mac OSX verläuft weiterhin schwerfällig. Beni besitzt als einziger einen Mac, konnte Orxonox aber nicht erfolgreich zum Laufen kriegen. Kevin war bisher nicht zu erreichen. Es wäre wünschenswert, diesen Punkt möglichst bald abschliessen zu können. Noch ist aber völlig unklar, wer dafür verantwortlich ist.
- KDevelop führt auf den Tardis Rechnern zu einem Problem beim Öffnen des Projekts. Als Alternative hat sich Eclipse bewährt. Gegebenenfalls könnte man versuchen, KDevelop auf den Tardis Rechnern wieder brauchbar zu machen (Update?).

## Aktueller Stand nach dem PPS

- Es wurden alle Branches in den presentation Branch gemerged mit Ausnahme von spacebomb, captureTheFlag und einem Teil von waypoints (3D Pfeil).
  - Die Änderungen im Code des spacebomb Branches werden verworfen, das Model wird übernommen.
  - Die Änderungen im Code des captureTheFlag Branches werden voraussichtlich verworfen oder archiviert, da das Projekt weder kompiliert noch getestet wurde.
  - Die Änderungen der AI im waypoints Branch müssen noch gereviewt werden. Der 3D Pfeil muss gefixt und überarbeitet werden, könnte dann aber voraussichtlich gemerged werden.
- Das Aufräumen der PPS Projekte findet zur Zeit im presentation2012merge Branch statt.
- Der Branch pCuts kann vermutlich auch (bald?) gemerged werden.

## Geplante neue Features

- Einführung einer Kampagne: Dies setzt voraus, dass man mehrere Levels verbinden kann und möglicherweise auch das Schiff des Spielers von Level zu Level übernimmt. Ein persistentes Speichern ist zwar noch nicht möglich, aber man könnte zumindest mithilfe des PlayerInfo Objekts Daten über Levelgrenzen hinweg erhalten.
- Design einer Ober-Map die der Navigation über mehrere Levels hinweg dient.
- Überarbeitung HUD (Mauro, was ist damit gemeint?)
- Quest-Dialog-Meldungen mit Audio/Video Support
- Cut-Scenes Implementierung: Dazu müsste die Kontrolle der Kamera etwas generischer werden, da sie im Moment rein clientside ist. Es wäre wünschenswert, die Kontrolle der Kamera so zu implementieren, dass Cut-Scenes auch in Multiplayer Matches funktionieren. Um Cut-Scenes zu scripten, müssten die Idee des Scriptable-Controllers wieder aufgegriffen und weiterentwickelt werden. Zur Zeit ist das Abfliegen von Waypoints die einzige Möglichkeit, einen Bot zu scripten.
- Die Dokumentation des Weaponsystems sollte verbessert werden. Gewünscht wird insbesondere eine Übersicht über das ganze Weaponsystem und das Zusammenspiel der einzelnen Komponenten.
- Weapons, WeaponModes und Munition sollten überarbeitet werden und stärker über XML konfiguriert werden, damit man für ein neues Modell (z.B. Space Bomb) nicht gleich zwei neue Klassen schreiben muss.
- Der NewHumanController sollte überarbeitet werden, indem er einerseits mit dem HumanController gemerged wird und andererseits HUD Elemente in die Overlays auslagert.

- Der DistanceTrigger sollte über die Physik-Engine getriggert werden. Ein Zusammenlegen von Trigger und MultiTrigger steht zur Debatte.