

GemuSoft Dokument

Vorgehensweise für Magical Twilight:

- 1) Minispiel mit Hauptfunktionen
 - 2) Ein Level mit drei Gebieten
 - 3) (Veröffentlichen (Wie kommt es an? Was für Kritik?))
 - 4) Story festsetzen
 - 5) Minispiel zum eigentlichen RPG erweitern
 - 6) Modellierung
 - 7) Organisation
 - 8) Sonstiges
-

Minispiel mit Hauptfunktionen

Es wird zuerst ein kleines Minispiel mit den Hauptfunktionen programmiert. Dazu gehören:

- GemuSoft Intro abspielen
- 3rd Person + Kamera
- Levels laden
- Animationen
- Musik
- Springen
- GUI
- Kollisionen
- Gegner
- Spawnen
- Leben verlieren
- Sterben
- Laufen + Angreifen
- Bone Physik
- Kampfsystem
- Skills
- Waffe halten

Es wird die **Irrlicht Engine 1.4** genutzt!

Ein Level

Das Level ist in drei verschiedene Gebiete aufgeteilt:

- 1) Normale, Graslandschaft
- 2) Eislandschaft
- 3) Berge/Wasser

Der Spieler merkt nur vom Aussehen, wann er ein neues Gebiet betritt, d.h. sie sind nicht extra instanziiert.

Veröffentlichung?

Das Minispiel könnte veröffentlicht werden, um Kritik und Wünsche von Außenstehenden zu bekommen. Ansonsten ist es natürlich größtenteils zur Übung der Programmierer und Modellierer. Andererseits wäre es eine Art „Beweis“, dass unser Projekt noch lebt.

Story

Die Story ist ein wichtiger Punkt des Spiels, besser gesagt der wichtigste (neben dem Gameplay). Hier ist mal eine grobe Übersicht der Story:

Der Hauptcharakter bekommt eine Prophezeiung: Er muss die Welt retten, indem er 5 Türme, die über den ganzen Kontinent verteilt sind, aktivieren. Sollte dies nicht geschehen, gelangt durch unzählige Portale das Böse nach und nach in die Welt um die Weltherrschaft zu erlangen. Diese Prophezeiung wird wahr und der Hauptcharakter wird auserwählt, um die Welt zu retten. Auf seiner Reise bekommt er verschiedene Gefährten, er wird jedoch von einer mysteriösen Gruppe daran gehindert, die Türme zu aktivieren. Es stellt sich heraus, dass sie dies nur tun, um ihre Dimension zu schützen; denn sollten die Türme aktiviert werden, öffnen sich die Portale in der anderen Dimension.

Am Anfang könnte man eine Rassenwahl einbauen, allgemein sollte man auch Klassen im Laufe des Spiels erlernen können. Eventuell bauen wir einen an Japan orientierten Stil ein.

Minispiel erweitern

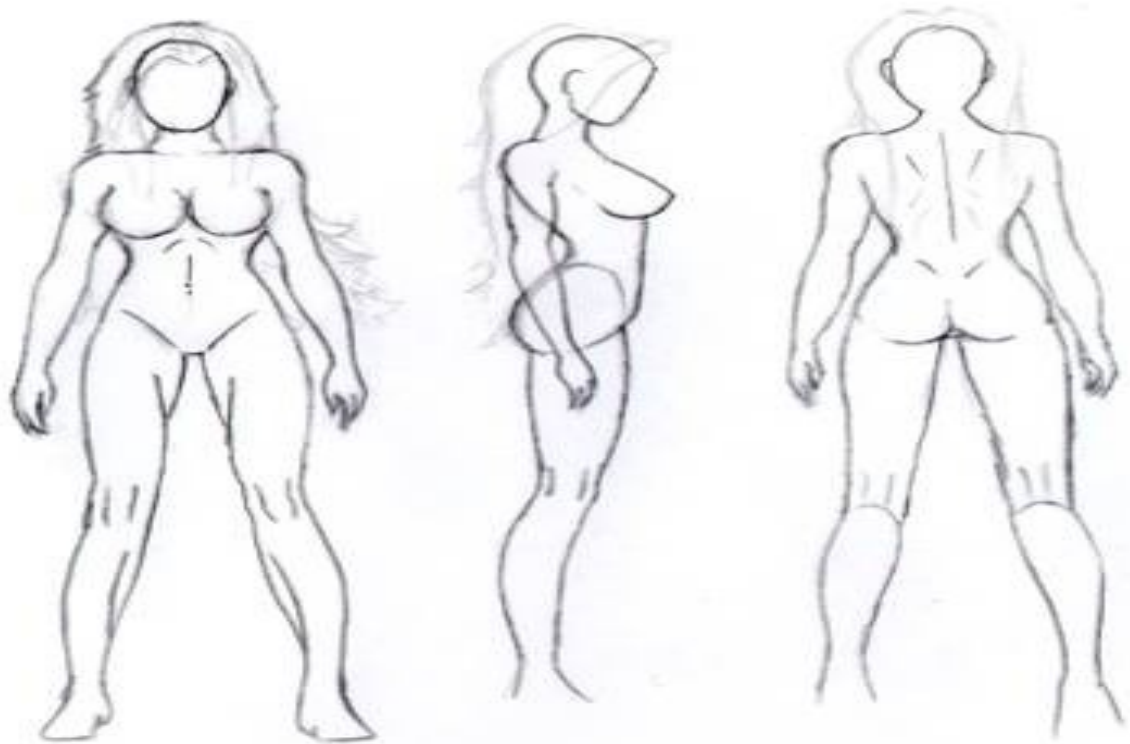
Wenn unser Minispiel fertig ist, sollten wir uns über weitere Funktionen Gedanken machen. Außerdem sollte zu diesem Zeitpunkt die Story feststehen! Wenn alles besprochen und festgelegt ist, können die Programmierer das Spiel weiter programmieren und die Modellierer können die Modelle programmieren.

Modellierung

Die Charaktere sollten aus primitiven Objekten modelliert werden, d.h. aus den Grundobjekten. Hier ist eine kleine Liste:

Kopf	Kugel
Haare	Hair Modul
Oberkörper	Gerundeter Würfel
Hals	Zylinder
Arme	Zylinder
Beine	Zylinder
Füße	?
Hände	Kugel
Finger	Zylinder
Augen	Kugel / Boole-Objekt
Ohren	?
Nase	(Textur)
Mund	Boole

Die Grundobjekte müssen natürlich noch stark abgeändert werden, denn beispielsweise der Kopf ist ja keine Kugel. Hier ist mal ein Blueprint eines weiblichen Charakters:



Blueprints sind sehr wichtig, eine Blueprint Seite ist z.B. [diese hier](#). Man kann sich bei den Blueprints schön die Proportionen beim Menschen ansehen, natürlich können wir auch selbst Blueprints anfertigen (sowohl für Menschen, als auch für Monster). Entscheiden sind dabei folgende Perspektiven:

Vorne, Hinten und Seite.

Geladen können wie Blueprints wie folgt:

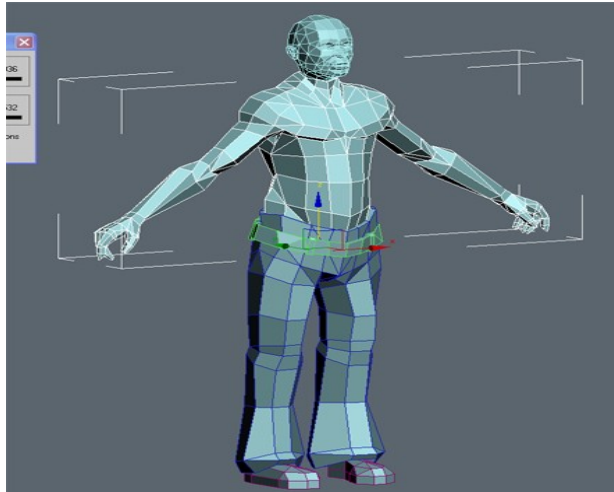
Bearbeiten → Ansichts-Voreinstellungen (Shit+V) → Hintergrund -->Bild: _____

Achtung:

Dies geht jeweils nur in den Perspektiven Oben, Rechts und Vorne!

Wenn das Blueprint im 3D Fenster angezeigt werden soll, dann müsst ihr eine Ebene erstellen, sie um 90° Y drehen und sie anschließend mit dem Blueprint texturieren.

Hier ist noch mal ein Bild, eines aufwendig erstellten Charakters;



Man kann erkennen, dass der Charakter auch aus Grundobjekten erstellt wurde, jedoch wurden aus den Grundobjekten die Polygone herausgezogen und angepasst. Das ist recht aufwendig und schwierig, mit etwas Übung jedoch hinzubekommen. Ein Nachteil wäre, dass das Charakter-Modell hinterher mit großer Wahrscheinlichkeit High-Poly sein wird.

Organisation

Um es zuzugeben: Unsere Organisation ist wirklich ein wenig „hippig“. Ich hab mir einige Gedanken gemacht, wie wir das lösen können.

Als erstes Thema wäre die **Datenübertragung**.

Ich persönlich bin für einen FTP Server. Dort kann man schnell zugreifen und jeder hat die aktuellsten Dateien immer parat. Für einen FTP Geld auszugeben ist jedoch unnötig, daher sollten wir uns entweder ein Free Angebot zulegen oder jemand stellt eine gewisse Kapazität von seiner Festplatte zur Verfügung (Nachteil: Nicht immer online). Falls jemand eine Kapazität zur Verfügung stellen würde, dann sollten wir das über [Hamachi](#) oder [SVN](#) machen. Eventuell kennt auch jemand von euch einen, der uns etwas von seinem Webspace zur Verfügung stellt. Zur Not könnten wir auch Wolf fragen, oder ein Angebot von [hier](#) oder [hier](#) annehmen.

Zum anderen finden die **Chattreffen** meist nur in unregelmäßigen Abständen statt. Das sollte sich ändern, als Vorschläge habe ich folgendes:

Einmal pro Woche – kurz und knapp: Was wurde erreicht, was wird als nächstes gemacht?

Einmal pro Monat – Etwas länger: Sofort die nächsten großen Schritte durchgehen.

Ich wäre für ein Chattreffen pro Woche, da kann man dann eine bis zwei Stunden einplanen und somit können auch kurzfristige Einfälle und Sachen miteinbezogen werden. Da können wir standardmäßig als Termin jeden **Freitag 20-22 Uhr** ausmachen.

Dann wäre da noch der Punkt **Community**. Wir müssen die Community selbstverständlich auf dem aktuellsten Stand halten, als Beweis, dass unser Projekt noch lebt. Wenn die Engine des Spiels halbwegs steht, können wir auch schon woanders Werbung machen, beispielsweise in Foren, Freunden erzählen, Werbe Seiten, als Banner in Signaturen, usw. Außerdem sollten wir regelmäßige Demo Versionen veröffentlichen, um Kritik zu bekommen. Wir sollten sehr auf die Wünsche und Anregungen von Außenstehenden eingehen, sodass das Spiel letztendlich eine große Spieleranzahl bekommt.

