

Jobs in der Spielebranche Programmer

Was genau macht ein Programmierer? Seine Arbeit bleibt für Spieler unsichtbar – aber ohne ihn geht gar nichts.

Jobs in der Spielebranche

Ausgabe	Thema
GS 05/08	Überblick
GS 06/08	Game Designer / Producer
GS 07/08	Programmer Interface Designer
GS 08/08	Level Designer Graphics Designer
GS 09/08	Modeller / Animator
GS 10/08	Writer / Localization Manager
GS 11/08	Sound Designer / Composer
GS 12/08	QA Manager, Tester Community Manager
GS 01/09	Product Manager PR Manager
GS 02/09	Der Weg in die Branche

Ein Autodidakt eignet sich auf eigene Faust Wissen auf hohem Niveau an. Ein solcher ist Martin Mayer, Chef-Programmer und Gründer von Fusionsphere Systems (**Geheimakte Tunguska**). »Programmieren? Das hab ich mir selbst beigebracht«, erzählt Mayer. »Ich habe zu C64-Zeiten zu spielen angefangen und fand es immer interessant, wie so ein komplexes Konstrukt wie ein Spiel aufgebaut ist. Also habe ich mit 13 oder 14 begonnen, mir die Programmiersprache BASIC beizubringen. In der Schule gab es zwar Informatik-Unterricht, aber da war nichts zu holen. Und ein Studium war damals auch noch sehr datenverarbeitungslastig – an grafische Oberflächen oder gar Spiele hat da noch keiner gedacht. 2001 habe ich mich dann einfach bei Ascaron

beworben und an Die Patrizier 2 und dem Addon Aufschwung der Hanse mitgearbeitet. Später bin ich noch kurz bei Spellbound gewesen, bevor ich mich 2004 selbständig gemacht habe.«

Er baut die Motoren

Was genau ein Programmierer macht, ist in wenigen Worten beschrieben: Er erschafft das Grundgerüst des Spiels, den Programmcode. Und zwar in mehreren Bereichen, denn moderne Spiele bestehen aus einem Konglomerat verschiedener Code-Strukturen: der Grafik-Engine, den KI-Routinen, dem Netzwerk-Code, Datenbanken, Tools und Editoren. Tools zum Beispiel bekommt kaum ein Spieler je zu Gesicht; sie werden von den Teams meist selbst entwickelt und sind die Schnittstellen, durch die Leben in ein Spiel fließt. Bei **Geheimakte 2** etwa sieht das folgendermaßen aus: In einem Fenster läuft das Spiel, und mit Hilfe diverser Schaltflächen kann sich der Programmierer Details wie begehbare Flächen, Aktionspunkte, oder bewegliche Objekte anzeigen lassen. Mit solchen Editoren werden dann die Scripts (die Drehbücher), die für die Spiellogik verantwortlich sind, an die Engine (den Motor) angedockt. Die Engine lässt sich am besten mit dem Motor eines Autos vergleichen: Sie ist die Grundlage für jedes Spiel und macht im Grunde nichts anderes, als Daten zu verarbeiten. Das klingt simpel, beinhaltet aber die Grafik, den Sound, die Physik und alle anderen Bestandteile eines Spiels. Der Programmierer baut die Maschinerie, in der dann die Zulieferungen aller anderen Mitentwickler verarbeitet werden. Neben diesen Kernaufgaben gehört vor allem in kleinen Studios noch einiges mehr zum Alltag des Program-



Der eigens für Geheimakte 2 entwickelte Editor zeigt den Entwicklern unter anderem die begehbaren Flächen und Interaktionspunkte der Szene.

mers: Er ist derjenige, der den Grafikern und Designern sagen muss, was die Engine überhaupt leisten kann. Und er ist es auch, der am besten weiß, was wie viel Aufwand bereitet. Deshalb muss sich der Programmierer regelmäßig mit dem Producer und den anderen Projektbereichen eines Entwicklerteams abstimmen.

Er muss motiviert sein

Es gibt keinen Master-Plan, um Spiele-Programmierer zu werden.

»Man muss auf jeden Fall Spiele mögen und verstehen, vor allem, wenn man in kleinen Teams arbeitet, wo der Programmierer auch Entscheidungen trifft und nicht nur seine speziellen Aufgaben abarbeitet«, erklärt Martin Mayer. Um überhaupt in die Branche zu kommen, sollte man Grundkenntnisse der wichtigsten Programmiersprachen und Entwicklungs-Tools mitbringen. »Woher die kommen, ist eigentlich egal«, sagt Mayer. »In jedem Fall sollte man die Programmiersprache C++ beherrschen. C# schadet nicht, ist aber in der Branche erst noch im kommen. Hilfreich, aber nicht zwingend notwendig sind auf jeden Fall noch DirectX, Visual-Studio und 3D-Studio Max.« Offizielle Ausbildungsmöglichkeiten zum Spiele-Programmierer sind rar: »Ausbildungsplätze für diesen Job gibt es, soweit ich weiß, in Deutschland nicht. Aber bei der Games Academy in Frankfurt kann man in die Richtung studieren.«

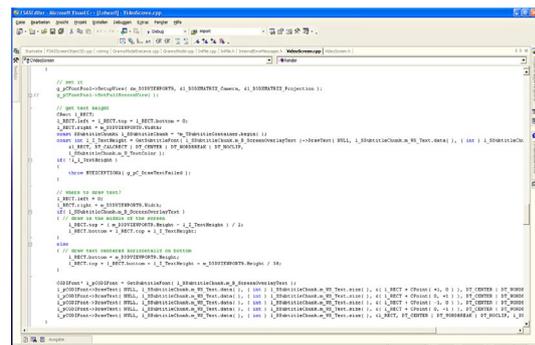
Er ist spezialisiert

Martin Meyers breites Aufgabengebiet ist längst nicht mehr überall Standard. Mit wachsender Teamgröße werden auch die Aufgaben der einzelnen Mitarbeiter spezifischer. Die Programmierer haben oft nur ein Gebiet, auf dem sie vorwiegend tätig sind. So gibt es reine Engine-Programmer, Script-Programmer, KI-Programmer und auch solche, die sich nur mit Editoren auseinandersetzen. Zudem werden nur noch selten eigene Engines programmiert; stattdessen schaffen die Studios existierende Systeme an. Dazu kommt häufig »Middleware« genannte Software, etwa Physik-Engine (Havok, PhysX) oder Grafikroutinen wie Speedtree. Es ist die Aufgabe der Programmierer, sich in solche System einzuarbeiten, sie zu handhaben und zu verändern.

PD

Unser Experte

Martin Meyers (29) erste Spiele waren Die Patrizier 2 und das Addon Aufschwung der Hanse. 2004 gründete er Fusionsphere Systems. Dort ist er Chef-Programmierer und Geschäftsführer.



In der Spielebranche wird viel mit dem Programm Visual Studio gearbeitet. Dieser Code beschreibt, dass in einer Zwischensequenz Untertitel eingeblendet werden.

Jobs in der Spielebranche

Interface Designer

Er baut die Schnittstelle zwischen Spiel und Mensch: Ohne den Interface Designer tut sich nichts in virtuellen Welten.

Was nützt einem die tollste Spielidee, wenn das Programm am Ende nicht reagiert, wenn man keine Einheiten oder Gebäude bauen und keine Waffen wählen kann oder sich in endlosen Menü-Irrgärten verliert? Um solche Probleme zu lösen, dafür gibt es spezialisierte Experten: Interface Designer. Sie bauen auf Basis der vom Game Designer entworfenen Inhalte die Benutzeroberfläche für ein Spiel. Diese wird auch kurz GUI (Graphical User Interface) genannt.

Genre ungleich Interface

Während in großen amerikanischen Entwicklerstudios die Berufsbezeichnung des Interface Designers längst Standard ist und man gezielt Stellenanzeigen nach solchen sucht, beschäftigen sich in den zuletzt kleineren deutschen Teams ein oder mehrere Game Designer und Grafiker mit dieser Aufgabe – so wie Sebastian Bombera und Jost Schweinfurth.

Die beiden sind dafür verantwortlich, dass man im Aufbauspiel **Anno 1701** effizient agieren kann. Sie haben gemeinsam das Interface dafür entworfen. Kann das so schwierig sein? Immerhin haben sich im Laufe der Jahre etliche Bedienungsmuster für die verschiedenen Genres etabliert. Da müsste das

passende Interface doch naheliegen. Jost Schweinfurth verneint solche Kurzschlüsse: »Allein aus dem Game Design generiert sich kein brauchbares Interface. Da können die Anforderungen noch so klar formuliert sein.« Stattdessen müssen Interface Designer immer neu in die Mechanismen jedes Spiels eintauchen und logische Lösungen suchen.

Keine ultimative Lösung

Eine Forderung, die der Interface Designer heute laufend zu hören bekommt, ist die nach einfachen Strukturen. Selbst die komplexesten Vorgänge sollen so überschaubar und intuitiv wie möglich umgesetzt werden – zum einen, um den Spielspaß zu steigern, zum anderen, um die mögliche Zielgruppe zu vergrößern. Auch in **Anno 1701** sollte sich einiges im Vergleich zu den Vorgängern ändern. Bombera erklärt den Ansatz: »Im Vorgänger Anno 1503 war es noch so, dass der Spieler durch ein kleines Fenster in sein Reich schaute, eingerahmt vom riesigen Interface. Für Anno 1701 wollten wir nicht nur einfachere und vor allem schlankere Bedienelemente, sondern den Spieler auch mehr in die Welt ziehen. Viele Informationen, die man früher übers Interface erhielt, kommen in 1701 direkt von den Einwohnern der Städte.«

Jost Schweinfurth relativiert die hehren Ziele: »Ich habe noch nie erlebt, dass ein Interface von jemandem, der keine Ahnung von Spielen hat, sofort in Gänge verstanden wird. Außer natürlich, es ist eine sehr simple Variante. Doch bei einem so komplexen Programm wie Anno 1701 geht ohne Erklärungen, ohne Hilfe-Einblendungen nichts.« Außerdem kann man es niemals allen recht machen, weiß der Grafiker: »Die

Wahrnehmung unterscheidet sich von Mensch zu Mensch. Gewisse Schaltflächen und Farben bemerkt der eine schnell, der andere weniger schnell, muss immer Kompromisse machen. Unter diesen Voraussetzungen ist ein ultimatives Interface eigentlich nicht möglich.«

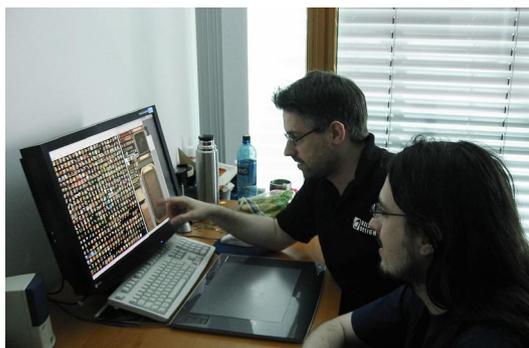
Ein Spiel simulieren

Methodik ist einer der Schlüssel zum Erfolg bei einem guten Interface. Das haben Bombera und Schweinfurth am Projekt **Anno 1701** gelernt: »Wir hatten eine schlechte Herangehensweise. Wir wollten zu schnell zu viel und arbeiteten schon recht früh an Nebenschauplätzen wie Ornamentik und anderen grafischen Elementen«, gibt Schweinfurth zu. »Man darf Bedienelemente nicht einzeln betrachten, sondern muss sie immer im Kontext sehen«, fügt Bombera hinzu. »Der eine arbeitet womöglich an den Schaltflächen, der andere an den Symbolen, und am Ende merkt man, dass das alles nicht zusammenpasst. Wir benutzen deshalb inzwischen ein System, mit dem wir ein Interface spielrelevant simulieren.« Damit vermeiden die Entwickler schon früh mögliche Fehler oder Stolpersteine. »Kleinste Farb- und Größenunterschiede bei Icons können extrem viel im Spielfluss ausmachen«, sagt Schweinfurth. »Es gehört zu unseren Zielen, Aufmerksamkeiten richtig zu lenken und möglicherweise Mausclicks einzusparen.«

Was ist also die bestmögliche Benutzeroberfläche? Sebastian Bombera und Jost Schweinfurth sind sich einig: »Wenn man sich bei einem Spiel auf die eigentliche Aufgabe konzentrieren kann und nicht mehr ununterbrochen daran denken muss, diesen oder jenen Schalter zu drücken, dann ist das Interface gelungen.«

Unsere Experten

Der 29-jährige Game Designer **Sebastian Bombera** (oben) und der 37-jährige Grafiker **Jost Schweinfurth** (unten) haben zusammen bei Related Designs die Benutzeroberfläche für das Aufbauspiel Anno 1701 erstellt. Während Jost schon einige Zeit in der Branche ist und zuvor bei Ascaron beschäftigt war, stieg Sebastian gleich nach seiner Ausbildung an der Games Academy beim Mainzer Entwickler ein.



Sebastian Bombera und Jost Schweinfurth beim Betrachten der zahlreichen verschiedenen Icons, die sie für Anno 1701 entworfen haben.



Die Benutzeroberfläche von Anno 1701 sollte intuitiver als in den Vorgängern werden.