# Game Production Process
## A Preliminary Study

LudoCraft | ELIAS-project
Tony Manninen, Tomi Kujanpää, Laura Vallius, Tuomo Korva, Pekko Koskinen

# Table of Contents

# Executive Summary

The aim of this project in providing a model of game production process is somewhat ambitious. There seem to be no single game production process model. The industry is still very heterogeneous in terms of game design and development. Therefore, this report illustrates the generic production model based on the games literature and industry survey. Furthermore, based on the preliminary study, the process model is refined in order to improve its usefulness in terms of out-sourcing and production distribution.

The point-of-view selected for this report is that of the game developer (game studio, production company, etc.). The relationships between game developer and other industry bodies will be briefly discussed.

The report offers two approaches into the topic: "game production process at glance" and "game production process in detail". The first part of the report describes the basic components of games industry and outlines the main phases in the overall game production process. The level of detail in this part is relatively low in order to build easy-to-access holistic description of the process. The second part of the report goes into detail in describing the tasks and outcomes of each of these main production phases.

In order to increase the usefulness of the game production model, the report outlines the main roles and expertise areas required by game design and development. While the titles and scopes of individuals vary greatly, the basic categorisation is intended to help in pinpointing each production phase into the corresponding expertise.

Furthermore, the report offers practical implications for out-sourcing projects in the form of checklist and an out-sourcing case study. The special emphasis is on cross-border production between Finland and Russia.

# Contributing to the Report

This report is a result of a preliminary study. The contents are constantly evaluated and revisions will be made according to the further research. You can contribute to this work-in-progress by providing feedback, suggestions and/or empirical research material in relation to the content, structure or methods. All correspondence welcome – both from game industry and from academia.

Contact: Tony Manninen (tony.manninen@oulu.fi)

# Introduction to Game Development

Game development can be seen as a specific form of software development where certain product and/or service is designed and developed. The outcome of the development, i.e., digital game, comprises of assets - audiovisual material and software, which generally exist only in electronic format. Due to the heterogeneous nature of game assets, the development requires multi-talented teams consisting of skilled individuals working in seamless collaboration.

The game design and development process is a tedious effort to merge multiple disciplines together. Today, the trend has been towards more specialisation, whereby each business unit becomes defined as a separate entity. Figure 1 illustrates the field by offering a game developer – publisher – consumer -model. This report focuses on game developer and especially on the game production process, which mostly undertaken by the developer.

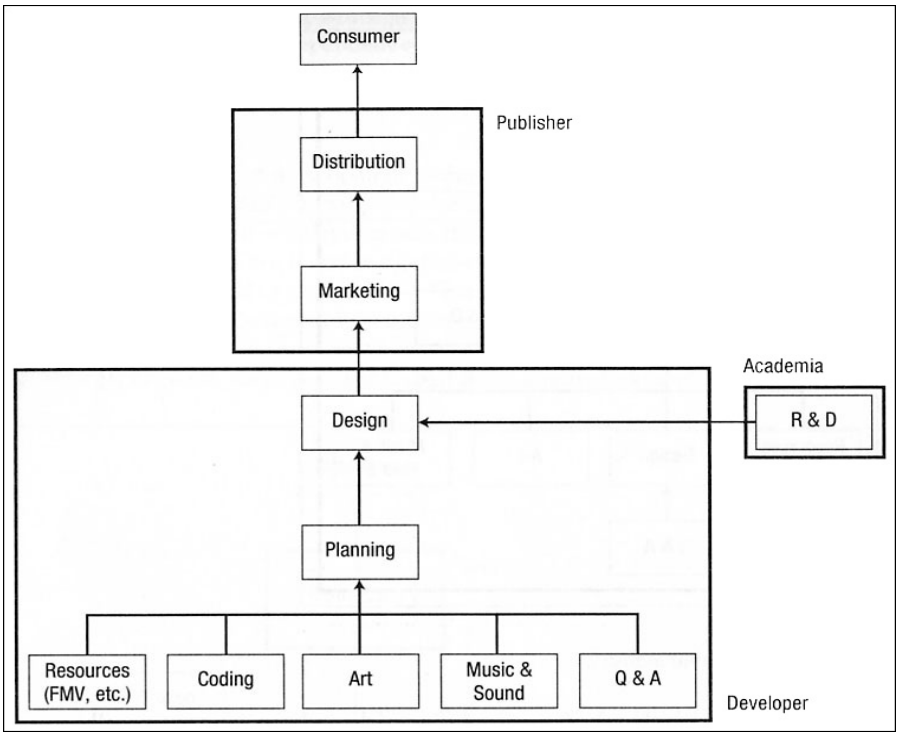


Figure 1. A current representation of the game industry. (Grassioulet 2003).

Producing electronic games is a complex and expensive process. A developer's goal is to produce the highest quality game within the limits of the resources and budget. The publisher's goal is to produce a best-selling game while limiting their risk by keeping costs low. There is a common interest in producing a successful product between these two

parties, and also a conflict of interest in how much money and time that product should require. (Fullerton et al. 2004, p. 347)

In relation to the aforementioned Figure 1, the main actors in game development and game business are the game developer, publisher and consumer. Figure 2 outlines the basic relationships between these parties. It should be noted though, that the business model illustrated in Figure 2 is highly abstracted and should be considered as simple guideline only.

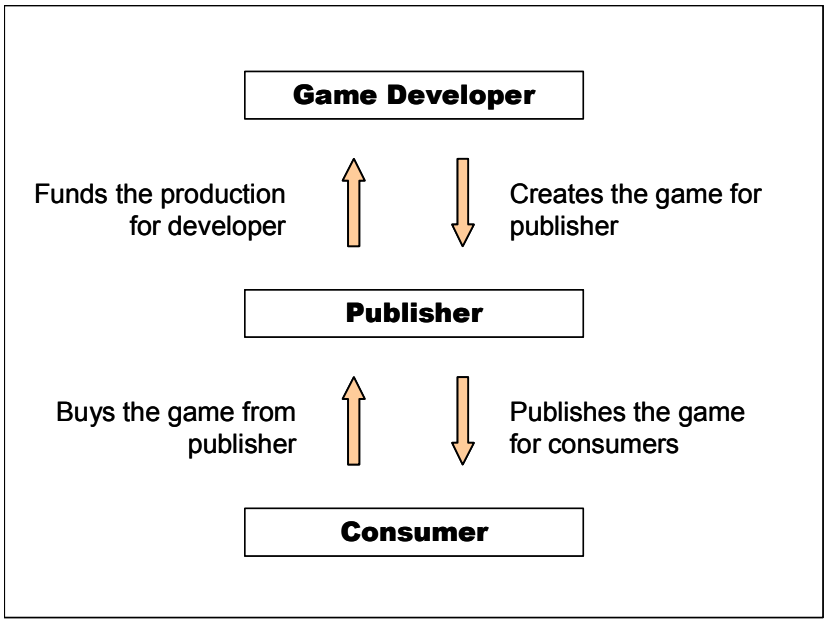


Figure 2. Abstracted point-of-view into the relationships of game business.

***Game Developer*** is the company, studio, or team of individuals who design and develop the game. Main part of game production process is managed by the developer, while the consumer business is generally left for others.

***Publisher*** acts as a source for partial funding in the form of development budget and/or royalty advances. The role of publisher is generally specified in the publishing contract made between the developer and the publisher. Prototype or demo of game is mainly used as a negotiation tool if the track record of the developer does not grant the contract.

***Consumer*** is the player or other customer/client buying the game or game services. Customer is rarely in direct contact with the game developer, since most of the business is managed by the publisher.

Additional roles, however, can be defined depending on the level-of-detail and scale of the business model. Some of the business roles include:

*Platform Developer* can be a separate entity responsible of hardware and/or software where the game assets will be developed onto. This enables the game developer to concentrate on the contents of the game instead of heavy technological resource investment.

*Funding Body* represents the third party financial partner who invests in the development costs of the game. Venture capitals, non-profit organisations, and individual agencies may share the risks of development costs.

*Distributor or Operator* disseminates the game either to consumers or retailers. Distributor may provide part of the development funding, but generally distributor is contracted by a publisher.

*Retailer* manages the sales of the game to the consumers. Retailer generally purchases copies of the game from the distributor (or publisher) and then sells those with a profit margin.

## *Game Developer*

Game development is an area of work, which includes a high-level of motivation, enthusiasm, and visionary innovation. While the work processes themselves do not differ greatly from those of software industry, the nature of the game development seems to especially attract people who enjoy games. Many game developers genuinely love their work. Game production could be, and have been, referred to that of film industry in terms of creativity and artistic components.

The game development as a profession is a risky, hectic and competitive form of earning ones living. New developer (i.e., game studio) will be evaluated based on game concept, development team resources, production process competence, budget management capabilities and quality assurance skills. Therefore, the whole production machine needs to be high-quality. Most valuable and concrete asset for a game developer is a substantial and strong set of reference productions, i.e., the track record of previous game titles developed by the team.

Game development generally includes design, programming, creation of audio visual material and testing. Historically these tasks have been done by small development groups – even one man "teams". In the early days the artistic creativeness and authorship were strong parts of the game development. Nowadays the productions have increased in volume into a level where it is more suitable to talk about production factory.

Typically game studio designs and implements games based royalties or advances provided by the publisher. Studios work for the publisher because generally they do not have the

capital, distribution channels, marketing resources or will to publish their games. Therefore the studios are in the mercy of publishers – there is no stable income and if the production is shelved before release, the studio can end-up being out of assets. Intellectual property rights (IPR) go mostly to the publisher, however, growth of developer may be dependant on the possibility to hold IPRs.

Game studios, thus, rarely sell their products to end-users (consumers, customers, players). However, the studios usually involve consumers in terms of focus groups interviews and beta testers.

## *Publisher*

The role of the publisher is to fund the development of games in order to make profit by publishing these games to the market. Publishers generally take responsibility in manufacturing, marketing, PR, distribution, and customer support related to their games. Major part of publishing industry originally grew out of developers. Also, bigger publishers own, or they have internal, game studios.

Publisher acts as a source for total or partial funding in the form of development budget and/or royalty advances. With this type of funding model, the publishers assume most of the risk. However, they also tend to take most of the profits. The games business has ended-up in situation where consolidation of publishers acquiring individual studios produces massive global bodies that cover many areas of games industry (for example Electronic Arts).

The role of publisher is generally specified in the publishing contract made between the developer and the publisher. Prototype or demo of a game is mainly used as a negotiation tool if the track record of the developer does not grant the contract.

While most game studios – especially the small ones and star-ups - are at the mercy of the almighty Publisher, the so called “Star Developers” can often bully publishers, because publishers are desperate for content. This status. however, is not easy to acquire. Publishers generally consider game development as part of their profit-making. If one studio cannot deliver, they contract another.

## *Consumer*

The role of the consumer in games industry is not like that in overall software business. The players themselves are consuming games based on other than utilitarian reasons. Games business is, therefore, part of entertainment industry just like film and music industries.

The traditional models of supply and demand do not fit well into games industry. While there may be demand for a certain game, the consumers tend to buy what is on the market

based on their personal preferences. Marketing is a strong component from the consumers' point of view. Even high-quality and engaging games can go unnoticed if there is not enough marketing effort. Game previews, media publicity, hype and other forms of preliminary buzz are used to raise the attention of the consumers.

Individual game studio may provide some added-value (especially the so called respectable developers) for the consumer, however, it is usually the game (i.e., the product) itself that matters. Or, as the current trend seems to indicate, the previous versions of the game. Whereas publishers prefer low-risk and high-profit game titles, the consumers also seem to buy safely. Unique and off-mainstream titles have their audience, but it is still marginal.

The latest trends in games industry involve casual gamers (e.g., people playing occasionally just to pass their time) and serious games (i.e., games used for utilitarian purposes). The casual gamers offer different target group and business model for publishers. While small games with reduced production effort can attract consumers, it is difficult to make profit in this sector since casual gamers are not necessarily willing to invest in their gaming. Furthermore, with slightly different consumer models, the serious games sector seems to be closer to traditional software business. This trend indicates potential for aspiring game studios in terms of non-mainstream game development.

## *Business Models in Game Development*

Game business markets are experiencing a very strong period of centralisation. Role of editors and distributors are growing in significance. Game business is also a very capital-intensive business area and the market is highly competitive and global from its' very nature.

The current situation with global market segment of mainstream PC/console games software is highly fixated. Although many entrepreneurs have made attempts to create small-scale games companies or production, in current publisher-dominated global market only a very few emerging firms have attained the necessary linkages and scale to be able to be treated seriously.

There are a couple of basic business models followed by companies working within game development. Table 1 outlines the roles and main revenue models and substance of the companies.

**Table 1. Basic business models within games industry.**

| Role | Revenue models and substance |
|---|---|
| Software developers and publishers | Money from game sales<br>• Royalties, advances, sales revenue<br>Internet games<br>• Initial game sale<br>• Monthly fees |
| Console developers (i.e., hardware platform developers) | Proprietary media delivery<br>• Lose money on consoles (the faster they sell, the faster they go out of business)<br>Licences the platform to be used by studios/publishers<br>• Charge fee for each game sold |
| Tool developers (i.e., software platform developers) | Create "engines" and "middleware" and sell to game developers<br>• Licensing, consult services |
| Contract services | Motion capture, art, cut-scenes, sound effects, music, …<br>• Subsidiary, out-sourcing partner |

The biggest important notion about game business is the fact that regardless of personal interests, it is still all about business. There may be portions of art, culture, fun, personal fulfilment, and what not – but the core of the business remains. Money matters, profitable deals survive, and if the games cannot be sold, there will be no funding. Contemporary game business is a harsh world where naïve developers get stomped before they realise it. Idealistic goals are not bad – they just need to be profitable in one way or the other.

Big developers with a substantial and proven track record can have the luxury of dictating the direction of the product and production. Small studios and start-ups are generally under the mercy of the publisher. While this is by no means morally right situation, there are few exceptions. Game developers should, therefore, be as professional as possible in their planning, negotiations and agreements. Legal and business expertise are strong assets in any game studio – whether internal or via partnerships.

## *Game Development as Global Industry*

Games are universal phenomenon. Contemporary computer and video games are played across the globe. Cultural differences cause variations in game types, trends, and consumer behaviour, but games are games everywhere. Therefore, games industry is a heavily global

enterprise. The markets reach outside national and continental borders. Games are sold – and need to be sold if the success is to be achieved – within global markets.

Actually the word "global" has two meanings here: (1) global markets, gamers and games; (2) global development via mergers, partnerships, and out-sourcing. In the largest-scale scenario, a multinational and global game company is making and selling games for the global audiences. In a slightly smaller scene, there are numerous examples of game companies reaching global markets by local development. In those cases, however, the effective marketing and sales have previously required global partnerships. This may well change with the increase of Internet-based game stores selling downloadable content (e.g., Valve's Steam)

One common feature among the game development companies is the relatively strict and focused work allocation of the development personnel. Each member of the game production team has a specific role and responsibility area. The production process can be seen as an assembly line, where each step of the production is conducted by an individual.

Furthermore, the trend, especially in Russia, is to out-source the parts of production, which are more art and design oriented. The programming is done in-house, but graphics, 2D/3D design and animation, and audio is bought from outside. Only the biggest game studios have the possibility to keep all the required expertise in-house.

The following sections outline the game production process and the corresponding expertise areas in order to offer an insight into the world of game development.

# Game Production Process in Brief

Currently, there is no single game production process model, which could act as a standard for the industry. Each game developer follows its own adapted and modified process. While average budgets of game development lie somewhere around 2 to 10 million euros with 30-60 persons in development for up to 24 months, there are huge variations in the figures. For example, the development costs of a contemporary mobile game are much lower than those of a game console.

The characteristics of a game development have a lot in common with software development and film industry. The logically beneficial trend to shift game development more towards the software industry, however can cause some problems because software production processes tend to be

- goal-oriented,
- cost effective,
- efficient in time management,
- heavily using production breakdown in to reusable components.

This conflict between the creative work of game development and goal-oriented effective process can still provide creative innovations, but generally suffers from unstable production management and consists of high-risk investment from the publishers' point of view.

In order to solidify the creative process of game development, the following description of game production process starts from the generic top-level process model. This Chapter provides a quick and brief look into the game production phases. After the overall structure is established, the details of individual production phases will follow in Chapter "Game Production Process in Detail".

## *Generic Game Production Process Model*

The core aspect of the game production process is that a project is developed and approved in stages. Each stage is defined by milestones. Contracts between publishers and developers are typically based on these milestones, and the developer is paid a predetermined amount for each milestone it reaches. (Fullerton et al. 2004, p. 347)

The creation of a computer game is carried out in six major stages:

(1) **Concept Phase - Specification and planning** – including the main features of the game (concept document; target audience, platforms, genre, references, and first draft of the planning). The concept design, from idea conception to pitching, is the most creative part of the game production process.

(2) **Pre-production Phase** – including the game and level design issues such as the context of the game, global scenario, principles of gameplay, look and

feel, human factors principles, object-oriented specifications, and the geometry of the space (a given maze or a race circuit). Both game and level design are made to build a flexible prototype, allowing for a more refined cost and planning evaluation. The basic idea of preproduction, therefore, is to plan, test and evaluate everything possible before the final production starts.

(3) **Production Phase / Development** – including the creation of all the different elements of the game (graphics, sounds, full-motion-videos, programs), and their integration within the whole system.

(4) **Quality Assurance Phase / Validation and testing** – including the validation of a functional testing (Alpha phase), which consists of a QA (quality assurance) test that focuses on principles, gameplay, user interface, aesthetic qualities, and ability to respond to the market needs. Only after are we allowed to get to the debugging process (Beta phase).

(5) **Release & Launch Phase / RTM phase** – including the release to manufacturer (master disc).

(6) **Post-release / Maintenance** – including patches and upgrades development.

In addition to the development stages model, the production process generally involves heavily iterative tasks that may cause the boundaries of each stage to shift.

## *Game Production Process Outline*

The overall game production process is relatively large and contains numerous small tasks and project phases. The following process breakdown illustrates the phases starting from the initial game concept design and going all the way to the post-release activities. The point-of-view, in this case, is that of the game developer.

**1. Game Concept**
- 1.1. Idea Treatment
  - 1.1.1. Inspiration
  - 1.1.2. Synthesis
  - 1.1.3. Resonance
  - 1.1.4. Convergence
- 1.2. Synopsis (Pitching)
  - 1.2.1. Overall direction and description phase
  - 1.2.2. Rationale & Preliminary market analysis
  - 1.2.3. Target audience decisions
  - 1.2.4. Treatment Plans
  - 1.2.5. Key Idea Focusing
- 1.3. Initial Game Design
  - 1.3.1. Synopsis refinement and expansion
  - 1.3.2. Brief Design Document
- 1.4. Initial Concept Art Design
  - 1.4.1. Concept Art Catalogue
  - 1.4.2. Early Sketching

1.5. Initial Prototyping or Concept Demo Construction
1.5.1. Low-tech Prototyping
1.5.2. Concept Demo

**2. Pre-production**

2.1. Premise Refinement
2.1.1. Theme Research
2.1.2. Thematic Adaptation
2.2. Market & Competition Analysis
2.2.1. Requirements Gathering
2.2.2. Comparative Analysis of Competing Products
2.3. Revised Production Plan Documents
2.3.1. Vision Document (Concept Document)
2.3.2. Game Design Document
2.3.3. Art Design Document
2.3.4. Technical Design Document
2.3.5. Project Schedule
2.3.6. Risk Mitigation Plan
2.3.7. Software Testing Plan / Quality Assurance
2.4. Pre-visualisation
2.5. Demo/prototype development
2.6. Tool development
2.7. Manuscript Authoring

**3. Production**

3.1. Concept Art
3.1.1. Final Concept Art
3.1.2. Clay Models
3.1.3. Blueprints
3.2. Level Design
3.2.1. Level Geometry
3.2.2. Object and Actor Placement
3.2.3. Lighting
3.3. 3D Design
3.3.1. Static Objects
3.3.2. Dynamic Objects
3.3.3. Effects
3.4. Animation
3.4.1. In-Game Animations
3.4.2. Cut-Scenes and Full-Motion Video
3.4.3. Effects
3.5. Textures & Graphics
3.5.1. Menu & HUD Graphics
3.5.2. Skins – Textures for 3D Models
3.5.3. Textures for Game Levels
3.6. Audio Design
3.6.1. Menu and Background Sounds & Music

## *Iterative Cycles Define the Production Phases*

The aforementioned production phases and individual tasks do not follow a strict waterfall model where the production phases are sequential and follow each other in particular order. Most of the game design and development is iterative in nature. The main iterative cycles of game production are formed by design-implementation-testing –phases, which can occur in any point of the production. Figure 3 illustrates an example of cyclic iteration, which can overlap several parts of the production process phases. This iteration should always be taken into an account when planning and scheduling the game production.

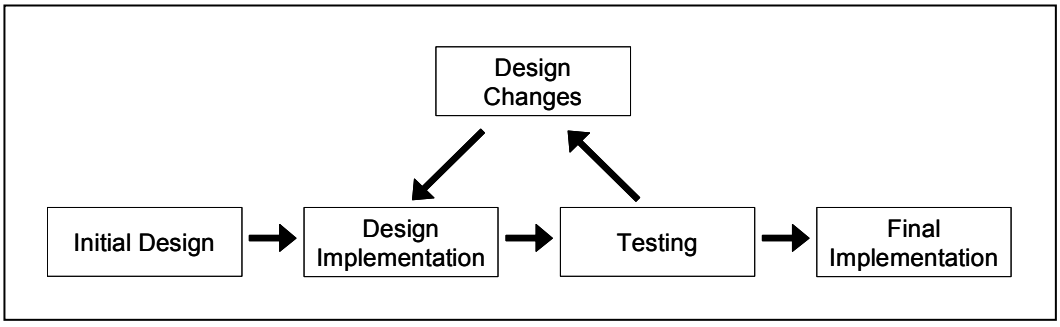


Figure 3. Iterative game design cycle overlaps the implementation phase.

In terms of out-sourcing and production distribution, the iterative cycles are risky to cut in pieces. If the production is distributed in a way, that parts if such cycle will be done in company A and parts in company B, the effect flow of production requires heavy inter-company communication and smooth material sharing channels.

## Iterative Development

The iterative design-implementation-testing loop is basically used either in order to reduce the risks or to gain incremental knowledge about the success of the plan (i.e., game design). The ambiguous player feelings of engagement, experience and fun are difficult, if not impossible, to design in advance. Thorough planning and strong experience increases the potential for success, but still most of the game developers are not willing to take the risk.

By dividing the design task into small and incremental phases, the development team will be able to get concrete feedback regarding the current state of the implementation. A common heuristic among games industry is that there should be playable version of the game when 20% of the project duration has passed. And that is only as a ultimate end-marker – usually the teams aim at producing something playable already earlier than that.

The numerous possible iterations during several production phases make it difficult to illustrate a single and coherent production model. However, the model described in this report forms the basic building blocks – or stepping stones - of the production. These steps can be taken in strict sequential order. Or, as is usually the case, some of them can be completed several times – each cycle ending closer towards the final stage.

For example, the iterative cycle illustrated in Figure x can be mapped on top of the early phases of game design. The implementation and testing, in that case, could comprise of, for example, initial prototyping and evaluation of the game mechanics. The final implementation of this example could then be the design document for game mechanics to be implemented during the next phases of the development.

## Testing as a Continuous Process

Another clear example forcing flexibility in the production model is the testing. Although Quality Assurance tasks and testing are defined as specific components of the production process, they are usually integrated into all production phases. The game developer should, therefore, consider issues related to testing during every step of the production – from initial design to release and post-release.

While the production model includes testing as one of the production phases, the task can be considered as a iterative sequence of critical procedures to be applied into each of the

underlying production area. The main tasks of continuous testing process include the following:

1. Seeing if things work
    - The content and art of the game is constantly being tested and scrutinised before it is integrated and approved. For example, Art Directors and programmers work together to make sure artwork will be compatible with the overall game.
2. Fixing problems
    - Artwork is fixed or changed by the artist and then tested in the game once again
    - Program modules are fixed by the programmers and tested again
3. Scrapping ideas that cause problems
    - Many times due to artistic, time-related, or technical reasons, artwork and other assets have to be shelved.
4. Planning for the unexpected
    - Planning time for the unknown (risk analysis, risk mitigation)
    - Extra time is allotted for any unexpected mistakes or upgrades made in the game. Time varies depending on the scope of the project.
5. Decision making
    - Preparation for making difficult decisions in the midst of a game project i.e.: reassigning an artist to another part of the level or firing staff. Also the game's size and scope may be reduced to fit certain needs.

## Review Processes

In addition to continuous testing process, the produced assets need to undergo either formal or informal review processes. These are used as controls and measures to minimise wasted effort and to weed out low-quality work before it pollutes the project. The procedures to review each asset created help in maintaining accurate project information and control during the lifetime of the game production. The reviews include, for example, design, document, technical and art asset reviews.

It should be clearly specified who will conduct the reviews, what are the procedures and how the changes will be implemented. These decisions are part of Quality Assurance and Risk Management tasks of the project and should, therefore, be finalised before entering the production phase.

The format of a review is roughly the same regardless of the material being reviewed. The two classes of review are formal and informal. A formal review is conducted by selected members of the team. The corresponding material will be distributed before the meeting, so the actual purpose of the meeting is to discuss the reviews that the reviewers performed before the meeting. The procedure involves the use of review forms and other structural

documentation. An informal review, on the other hand, involves gathering a couple of team members around a work station and talking them through the work, giving demonstrations when applicable. Despite of the informality, the review sessions should always produce clear and measurable outcomes, such as, design change-lists, suggestions, and prioritised refinement requests.

In projects without review procedures, sub-standard work can go undetected for a long time. This kind of problem shows up much later, usually when someone else has to maintain or update the affected area of production. Reviews allow the sharing and dissemination of information among the production team. Because the work is being discussed and reviewed in a group, the information is naturally transferred among the team, which greatly reduces risk.

### Planning for Changes

No matter how thorough the production plan and game design documents are, there are bound to be changes induced by various internal and external factors. Unexpected things can happen, and the more complex of a production the higher is the risk for something to go wrong.

While the design phase of the game production can never be emphasised enough, the design itself should not be above the practicalities of life. Flexibility and contingency plans are ways to introduce risk mitigation already during the planning and design phase of the project. If the developers do not know where to go, then they can at least have a couple of possible directions to be further evaluated.

Many of the game productions could increase in quality if there was enough resources and commitment for the design and pre-production phases of the game development. The more the team learns about the task at hand, the better they are prepared for the unknown as well. And when the unknown occurs, the production process model should adapt to the occasion accordingly.

## *Game Production Scheduling*

A production process becomes a project only when it is tied to a schedule and resources. In order to help bridging the gap between a continuous process and bound project, this chapter offers a generic overview of the most common tasks and milestones to be scheduled when planning the production. The milestones are tightly coupled with the production stages, and, therefore, may vary in scheduling and sequence.

Most publishers have a so called “Green-light Process”, which is used to determine the projects that can go forward. This process consists of a set of milestones, or stages, each of which focusing on a different version of the game. One common model for these

milestones is based on different steps in game design and development. Developers generally submit to evaluation committee at five, independent stages (Table 2).

**Table 2. Milestones based on different steps in game design and development.**

1. **Concept**
   - Initial game concept with corresponding plans and design documents
2. **Assessment**
   - Validation of game concept (market analysis, competition, etc.)
   - Validation of resources (development team, management, plans, etc.)
3. **Prototype**
   - Illustrative and communicative mock-up of the game
   - Visual and functional aspects
4. **First Playable**
   - Demo version of the game
   - Initial look-and-feel of a limited part of the game
5. **Alpha**
   - Feature complete version of the game
   - All the planned and designed features in place and integrated to work as a whole

These milestones are practically a series of deadlines, which are established in order to make the game production workload more organised and attainable. At each stage, the committee reviews the results and decides whether or not to continue funding. These Milestone meetings and walk-throughs are held to determine how well the game production is progressing. For a game developer the key is to show the progress – there needs to be increase in content, quality and level of functionality, as well as, clear indication of progress towards the right direction.

The analysis usually emphasises the evaluation of market potential and results in possible adjustments of unit forecasts accordingly. If the project results do not meet the expectations, or the objectives set by the publisher, the game production is not continued. Generally, this means that the publisher is not funding the development any further, thus, making the game studio to either find a new publisher or shelf the production.

# Game Production Process in Detail

This chapter illustrates the main production phases in more detail by providing descriptions of each of the production phases and tasks. This illustration lays ground for the forthcoming game production breakdown suggestions. While all of the production tasks are not necessarily required in every game production, the material presented here tries to be as exhaustive as possible.

## *1. Game Concept*

The Game Concept phase involves the initial steps in creating the idea, or high concept, of the game. The earliest stage of game design includes the tasks of getting and refining an idea for a game. The main goal, or result, of this phase is the writing of a synopsis (or high-concept document). Not all details are needed at this point, but the team needs to have a clear understanding of what the game is about.

### 1.1. Idea Treatment (or Initial Concept Treatment)

The idea treatment phase starts from getting the initial idea for the game. The idea is a single most persistent entity in the game development cycle. It can evolve and develop as the game progresses, but it is there from the start. The core idea can come from outside (i.e., from publisher, funding body, client, sub-contractor, etc.), or, in some cases, the idea is created in-house. Whatever the original source of the idea, the main tests for its rigour are the treatment process and the ultimate writing it down on paper when it is mature enough.

#### 1.1.1. Inspiration

Inspiration is all about where to get ideas. The industry is currently following safe sources for ideas. Game sequels, movie licences and other sure hits tend to form a majority in terms of idea pool. The idea creation process can be anything from a brainstorm to museum visit and from literature session to web search. Originality of ideas can present itself in many aspects of a game: the gameplay, the story, the setting, the characters, the interface, or in technology. Freshness and unique combinations is usually the key in terms of inspiration.

#### 1.1.2. Synthesis

Synthesis involves the combining of the ideas. While inspiration phase aimed at generating original and interesting ideas in abundance, the synthesis requires that all components match with each other. The separate ideas, or pieces of an idea, need to work together. The main question during this phase is about what each idea can contribute to the others so that the emergent game makes full use of them all. The synthesis helps making the game concept coherent, believable and thematically sound.

### 1.1.3. Resonance

During resonance phase the aim is to create synergy from different ideas. Resonance is the way of making the whole greater than the sum of the parts. It is an effective means of making the story and the subject matter significant to the game players. One starting point for resonance is the setting of the theme. This, then, guides all the decisions, refinements, adaptations and other creations towards a common goal.

### 1.1.4. Convergence

The last step brings everything back on the track by finishing the concept. This phase is the first point where critical and analytical thinking will be utilised in order to craft the idea into a game design. The designers need to judge whether their ideas are going to work together to make a good game. Usually, similar games, or games with similar components, are studied in order to help the decision making. The main question here is about why do those games work or why do they not work?

## 1.2. Synopsis (high-concept, pitching)

Writing the synopsis (or high-concept document) is the first step after scribbling down the initial idea. The aim is to express the fundamental spirit of the game. The purpose of the document is to get green-light from the stakeholders (e.g., publisher, funding body, client, etc.). The document itself is concise with having only the key elements of the game concept. The length should be restricted to 2-4 pages. An example synopsis template is provided as an Appendix 1.

### 1.2.1. Overall direction and description phase

The key part of the synopsis is the high-concept, which should be a single, punchy sentence that describes the game in a nutshell. The high-concept acts as the backbone, guiding light and design framework for the whole game development. The high-concept is further elaborated by outlining the premise of the game. This should set the overall direction of the concept and provide a concrete look-and-feel of the game to be developed.

### 1.2.2. Rationale & Preliminary market analysis

The synopsis should also cover the rationale for the game. This may include a brief analysis of the existing markets and a possible summary of the closest competition. Questions to be answered include: Why the game is worthwhile to develop? Why the game should be created by this team? What is the motivation, background and vision?

### 1.2.3. Target audience decisions

The game designers need to know their target audience. Who are the players and why will they play this game should be analysed before the design is worked on in detail. Furthermore, the stakeholders generally want to know why this target group is valid from their point-of-view (e.g., financial, educational, marketing, or image rationale).

### 1.2.4. Treatment Plans

The synopsis should briefly cover the issues of game development and production in order to help the stakeholders to get a realistic estimate of the resources required. If the synopsis is to be used as a pitching document to get funding, the treatment should also cover the competence and experience of the development team. The main questions to be answered include: How the game is going to be built? What are the main methods, tools and processes? What is the basic technological platform?

### 1.2.5. Key Idea Focusing

The brief summary of the game concept should include a list of unique selling points. The so called killer ideas or most important features of the game help the readers to position the game on existing markets. Basically, it may be relevant to know what is different in this game when compared to the other games in market. If there is no obvious competition, the key ideas nevertheless support the high-concept by providing a set of concrete features to be considered.

## 1.3. Initial Game Design

The purpose of the initial game design (or game treatment) is to present the game in broad outline to someone who is already interested in it and wants to hear more about it. The treatment is designed both to satisfy initial curiosity and to stimulate real enthusiasm for the game. At this point it is not yet necessary to cover all aspects of the game in rigorous detail. This phase will not produce a full game design document. The brief game design document (or the game treatment) is used as an illustrative and easy-to-access document for presenting the game to the stakeholders.

### 1.3.1. Synopsis refinement and expansion

During this phase the gaps and open questions in synopsis are filled in and answered. The text is enhanced by using screen mock-ups and other figures, which make it easier to grasp the essence of the game. Basically, everything that is crucial to understanding what the game will look and feel like to play will be added at this point. The concept is refined in detail, the competition analysis is expanded and target audience is studied in more detail.

### 1.3.2. Brief Design Document (treatment document)

The initial treatment is still a simple document. It is more like a brochure that sums up the basic ideas in the game. In addition to the topics covered in synopsis, the brief design document should include descriptions about game world and level design, look and feel, background story, game objectives, rules, gameplay, user interface, entities (players and computer), start up, feature list, marketing, technical requirements and planned resource allocation. An example brief design document template is provided as an appendix 2.

## 1.4. Initial Concept Art Design

The visual appearance and the aesthetics of the game are important in terms of setting the theme, conveying the designed message and pleasing the eye. Furthermore, the visual aspects of game design support the functional approach by providing another dimension for both the designer and his or her written documentation.

### 1.4.1. Early Sketching

The early concept art needs to be developed in a sketchy half-formed way that fosters creativity. The ideal concept art, like the game design itself, at this phase is more concerned with broad strokes and capturing a general style than in finishing the fine details. The early sketches – and working with a concept artist – help the game designer to balance the look and feel of the game. At best this phase contributes heavily to the whole design process.

### 1.4.2. Concept Art Catalogue

The concept sketches produced this early in development help to convey a shared vision to the whole team. They spawn additional ideas, they set the overall atmosphere of the game and they help stakeholders to visualise what the game might look like when it is finished. The concept art catalogue is a collection of sketches usually created by the lead artist or art director of the development team. This material will form the basis for "Art Bible" during the pre-production phase.

## 1.5. Initial Prototyping or Concept Demo Construction

In addition to the design documents and sketches about the game, it is necessary to have a more concrete proof of concept. Game is all about an interactive experience and that is generally difficult to portray by using only static documents. People need to get their hands-on experience of playing the game before they can state their final judgements. That is when the physical prototypes, game element mock-ups and a playable demo have their crucial role.

### 1.5.1. Low-tech Prototyping

Digital games are ultimately about playing in similar way than any form of game, no matter how advanced the technology used will be. A best and cheapest way to get more proof about the gameplay, game mechanics, game rules, and most importantly, about the fun and challenge factor of the game, is to build prototypes that let people play the game. The prototype may be a selection of Post-It notes, board game, enactment session with props, or quick-and-dirty implementation using the tools from another game. Whatever the tools used, the goal is to be able to figure out whether the game is worth building or not.

### 1.5.2. Concept Demo

Interactive demonstration, or demo, is a partly implemented game, which is used to showcase the progress of development and the maturity of ideas. This is generally done by creating a small portion of the game in a way that is both aesthetically and functionally pleasing. Concept demos are the main proof-of-concept when the developer tries to get a

contract from the publisher. The initial concept demo may still be highly limited in terms of content and assets, but it should be able to convey at least one key idea of the game.

# *2. Pre-production*

Due to the complexity of multidisciplinary work, high costs involved and the potential human error, game production requires careful planning. This is executed in the form of pre-production phase (much like in film industry) and the ultimate form of well-conducted pre-production is a playable demo or prototype of the game. True pre-production includes the distillation of all the game's requirements, an analysis stage to determine the implications of these requirements, a culling state to meet the business parameters, and a detailed game, art, audio, and technical design to detail the requirements. Pre-production is the most important stage of the project and in ideal situation it should cover up to 25-40 percent of the pre-release time.

## 2.1. Premise Refinement

One of the ways games create engagement is with an overarching premise, which gives context to the formal elements of the game. The base-level effect of the premise is to make it easier for players to contextualise their choices, but it is also a powerful tool for involving players emotionally in the interaction by establishing the action of the game within a setting or metaphor. The overall theme of the game provides both an extensive set of design material and an area to be thoroughly studied in order to make the best out of it.

### 2.1.1. Theme Research

After the premise has been defined in more detail, the refinement task can be used to gather more information, data and additional material about the set theme. The aim is to collect as much material as reasonably possible within the budget. This may include historical books, films, music, encyclopaedias, field trips to locations and many other forms of research. The more material there is to begin with, the better is the chance of the material to be useful during the following phases of production.

### 2.1.2. Thematic Adaptation

Creating a premise that unifies the formal and dramatic elements is an opportunity for the game designer to heighten the experience of players. Contemporary digital games include elaborate premises in their design. These have evolved to the point where they can be considered fully realised stories. The adaptation phase brings in all the material gathered during the previous phase and applies that on top of the games formal system. Rules, entities and interaction in the game are explained using the theme. Moreover, the imposed theme sparks potential implementation plans for the games features.

## 2.2. Market & Competition Analysis

Games do not exist in vacuum – they are intended to be played, and possibly bought, by people. Whatever the context of the production, there is always a market to be thoroughly studied and analysed. Even if the game is a purely fun and freely distributed product, it is beneficial to know the target audience. Design decisions, resource allocation and other aspects of the production should be guided by the data and analyses acquired during this phase.

### 2.2.1. Requirements Capture

The purpose of a development process is to take the player's requirements and transform them into a functional game system. Basically it is about taking the vision of the gameplay – how the game should play – and turning that into a finished game. The requirements capture includes identifying all of the core activities the player can perform in the game. Furthermore, the requirements should be analysed in terms of target audience. Focus groups, target group analyses and other forms of requirements gathering methods may be used to understand the prospective players – and their aspirations – better.

### 2.2.2. Comparative Analysis of Competing Products

Every game has something in common with other games, no matter how unique and original the concept. If not done before, this is the time to go out and play all the possible games that may have some common aspects with the one to be developed. A comparative analysis should be performed to get a holistic picture of the current competition. While it is not rational to include every potential feature into the design, the analysis of existing games is a best way to understand what works and what not.

## 2.3. Revised Production Plan Documents

The purpose of creating a set of production plan documents is to know and understand what will be done during the project. Writing down what have been thought about in the form of diagrams and notes forces the team to drive the quality and quantity of thinking to the required level for making a production plan. Game design is only one part of the plan. Other areas include project planning and management issues, technological specifications and quality assurance plans. This set of documents is disseminated to stakeholders (e.g., technology group, marketing people, designers, artists, executives, etc.)

### 2.3.1. Vision Document (or Concept Document)

The Vision Document is an executive summary of the game design document that touches upon all the key features of the game in such a manner that the readers are able to grasp it in a way that they will request to see the demo and move forward with a deal. The main purpose is to keep the game's focus clear to all stakeholders – including the development team. The Vision Document should be liberally illustrated with images – both concept art and screenshots – to accurately convey the gameplay as well as give the game life in the hands of the reader. The average length of vision documents is somewhere around 5 to 20 pages and they are usually highly imaginatively put together.

### 2.3.2 Game Design Document

The game design document is the soul of the game. It describes to all the team members the functional requirements of the features that will form the game. The ideal game design document is complete and has seen revisions to fix gameplay and add clarity. The game design document defines the game (theme, look and feel, genre, etc.), describes the core gameplay, illustrates the features and functions of the game (levels, menus, dialogue, etc.), outlines the story and lists all the required assets. Basically, everything that is going to be developed should be outlined in the design document. The art, technical and resource requirements will be generated based on the design document. The better and more detailed the document is, the more efficient is the production phase.

### 2.3.3. Art Design Document (Art Bible)

The art bible is often composed primarily of concept sketches and other resources that artists can refer to as they are working on creating various visual assets for the game. Some times text accompanies these images, whether in the form of hand written notes on concept sketches or text descriptions describing the parameters artists should follow when coming up with new elements for the game. The art bible is usually not compiled or written by the designer, but instead by the lead artist working with his team. The art bible is the place where the look and feel of the game is comprehensively established and illustrated in detail.

### 2.3.4. Technical Design Document

The technical design document is the blueprint for the software engineers to use in the creation of the game. Ideally, it should specify what needs to be created and how it should be implemented. The technical design document must synthesise the requirements of the game, develop a software design, serve as a testing plan, and also supply the project manager with critical information such as the required developer roles, dependencies between tasks and developers, and an estimate of how long it will take to perform each of the tasks assigned to developers. In brief, the technical design document consists of both requirements gathering and analysis part and software architecture section.

### 2.3.5. Project Schedule

The heart of the project plan is a schedule that describes what will be accomplished, how long the tasks will take, and who will perform these tasks. The schedule is supported by other information such as milestone dates, task dependencies, and risk management plan. The game production schedule is not much different from any software development project. However, the schedule should account for the iteration cycles in design and production.

### 2.3.6. Risk Mitigation Plan

The risk mitigation plan (or risk analysis document) is used to prepare the team for any possible risks that might occur during the game production. Any risky areas of the project need to be explicitly called out, and alternative plans need to be formulated to get around these risks. One possible format for this is the top ten risks document, which enumerates the ten most significant risks to the project and their possible effects. With each of the risk

items there should be list of the actions that need to be taken in order to contain or address the risk. This list should be maintained, updated and delivered with each milestone of the project.

#### 2.3.7. Software Testing Plan / Quality Assurance

The testing plan should cover both how the game is going to be tested and what tools will be needed to commence the testing. Furthermore, the quality assurance task should also take into account the assessment methods, criteria and procedures for art assets. The software testing can be conducted as unit testing, white box testing, black box testing, and/or beta testing. These follow the standard software development project testing procedures. The art asset evaluation and validation plan varies according to the project and personnel, but the general rule of thumb is to delegate the planning for corresponding art directors.

## 2.4. Pre-visualisation

The game industry has adopted the pre-visualisation method from the film industry. The aim of the possible pre-visualisation task is to create rudimentary, simple and draft-like visual scenes portraying parts of the game. These can be sketches, still images, animations, physical mock-ups or even interactive walkthroughs of the game. The purpose of pre-visualisation is to enable stakeholders and team members to get a concrete idea of the games look and feel even before the game is fully functional. Many development teams, however, utilise game demos for this purpose and conduct pre-visualisation during the production phase while creating, evaluating and validating art assets.

## 2.5. Demo/prototype development

The game demo, or prototype, is partially complete preliminary version of the game, which is used to showcase some of the core features and assets in the game. Game demos are currently more or less de facto standard for getting initial funding in the game industry. Even the most brilliant game idea is difficult to convey in paper format. The game demo allows people to perceive the game in action – and even to get hands-on experience in trying out the game. The basic rule of thumb is that there should be playable version of the game latest when 20% of the overall production is completed. This pre-production phase game development should not be confused as the marketing game demos used to promote the forthcoming title to the customers.

## 2.6. Tool development

Before the production is started by full force, there is a need to prepare all the possible tools for the team. The better the tools to be used are, the more efficient is the production phase. The possible tools to be developed include audiovisual asset creation tools, level editing tools and quality assurance and testing tools. These tools are created by the lead

programmer, or by a specific tool development team in bigger projects. Even if the project utilises third-party tools and engines during the production, there is a need for file converters, pipeline procedures and other support tools that make the mundane tasks easier.

### 2.7. Manuscript Authoring

The manuscript authoring phase can be seen as a task where all the narrative and verbal assets of the game are created. Furthermore, it may act as a task in which a narrative analysis and design is conducted in order to define the structures and components of the games narrative. From the asset point of view, the manuscript authoring includes writing down the dialogue, menu texts, thematic references and other textual material which is to be used during the production. The premise for the narrative has been set in the earlier design documents and generally the authoring task continues well within the production phase.

## *3. Production*

The production phase of the game development is the heaviest in terms of personnel and time resources. This is the time when the game factory works to its fullest effort. The main tasks during production are the various asset creation tasks, but there is a significant portion of play testing and general project management procedures involved from the beginning to the end.

### 3.1 Concept Art

The preliminary versions of concept art have already been done during the pre-production. The concept art created during the production phase aims at final, authorised and frozen versions of the concepts. These are then used as both reference and official working material for the 2D and 3D design and modelling tasks.

#### 3.1.1 Final Concept Art

Overall visual look and feel designed through sketching in finalised in the concept drawings. Final concept drawings are produced for different areas of the game such as levels, characters, different items and buildings. Final concept art is given for 3D artists and animators, as well as, for texture artists to be used as reference material. The concept drawings should be made visible for the whole team, since they are one of the easiest ways to convey the visual appearance of individual game assets.

#### 3.1.2 Clay Models

Sometimes more complex and organic objects, such as characters, can be created from clay. This process helps the artist to understand the geometry and form of 3D object better than 2D concept drawings. Clay model can be also scanned straight to 3D modeller

software for further processing. Clay models can, therefore, be seen as a special form of final concept art.

### 3.1.3 Blueprints

3D artists like to work in various ways. Some prefer to work with general concept art and some with detailed concept art. Blueprints are a form detailed concept art. They describe in detail the measures of the designed object. Blueprints are usually created from different perspectives such as side view, front view and behind view. They can be imported directly into the 3D modelling software to support the creation of the digital models.

## 3.2 Level Design

Level design includes the planning and integration of various 3D objects that form the game level. Structures, terrain, trees and other elements that are needed to create the game world are placed into the corresponding locations. Furthermore, the level designer arranges the lighting and other characteristics of the game level. The work of a 3D artist and level designer may include similar tasks. The level designer, however, generally works within the scope of a game level and receives material from the 3D artists. Furthermore, it is generally the responsibility of the level designer to create a required gameplay support in the form of the game level.

### 3.2.1. Level Geometry

Game levels need structural formations. These may be terrain, landscapes, platforms, race circuits or other bits and pieces, which are organised in a way that creates the required gameplay. Many game engines include tools for level editing. 3D level terrains can also be created with specific 3D modelling software.

### 3.2.2. Object and Actor Placement

Placing different types of objects (such as buildings, structures, vegetation, items) is one major task in level design. Actual 3D objects are set on their places into a level. Also there may be additional action related helper objects (triggers) that needs to be placed within the level. These objects tell, for example, where and when different visual effects, action sequences, sounds, weapons, etc. appear in the level.

### 3.2.3. Lighting

Lighting is an important part of the level design. Various moods and feelings are mediated through lighting. Different types of light sources are set to the level and their attributes tweaked to achieve the desired result. Some lights are static and their effect may be calculated in advance for example with level editor. Some lights are dynamic and their effect is seen only in the actual game itself.

## 3.3. 3D Design

After final concepts and blueprints have been finished, the 3D artists can start their work. They create the required 3D models, which form the objects seen in the game worlds. These objects can be roughly categorised into static objects, dynamic objects and effects.

### 3.3.1. Static Objects

Buildings, plants, decorative items, elements of terrain, etc., can be considered as static objects. In other words, static objects are models that have no animation and/or have only decorative value. The player only sees these objects in the game but cannot interact with them (other than collisions). The production of these objects includes geometry modelling and texturing (skinning). Modelling involves the creation of a three dimensional representation (mesh) of the designed object. Texturing entails the painting, or colouring, of the object with a required texture map (or skin).

### 3.3.2. Dynamic Objects

Characters, vehicles, different types of items used by characters, etc., may require the status of dynamic objects. If a model is going to be in use rather than just being a static prop it can be considered to be dynamic. In other words, the player can interact with these objects. These objects also include animation sequences. The creation process of dynamic objects is therefore a bit more complex compared to static objects. Dynamic objects are constructed using modelling, texturing and rigging. Modelling and texturing are similar than in the creation of static objects. Rigging means the creation of control mechanism (such as bone structure) inside the 3D model.

### 3.3.3. Effects

Some of the effects seen in games, such as, smoke or fire, can be created with 3D modelling tools. The effects, however, are often created as part of the level design and may be tightly combined with programming and/or texturing. Depending on the type of the effect it might be considered to be either dynamic or static.

## 3.4. Animation

All movements in a game world are presentations of some kind of animation. The animator makes the world look alive. The static world of level geometry, the objects (static and dynamic) in the game world and player-controlled movement are all brought to life using the best suitable form and type of animation. Animations can be effects, combinations from animation library (animation sequences) or programmed actions (e.g., inverse kinematics and graphics programming).

### 3.4.1. In-Game Animations

Game world may include pre-designed and real-time animations. Animations connected to dynamic objects serve a more functional purpose. They show what a character is doing, how vehicles work, etc. These animation sequences are usually designed specially for each dynamic object and its usage. In addition to pre-designed animation libraries, animation

can be created with code. This is generally seen as the player's real-time action. For example, when the game character runs on a street, in addition to the feet and body movements (from character's animation sequence library), the location of the character changes according to where the player moves the character.

### 3.4.2. Cut-Scenes and Full-Motion Video

A part of a story of a game can be revealed to the player by means of Cut-Scene animations. These animations represent a form of short animated movies in games. These sequences are shown in-between playing, and the player can not interact with them in any other way than skipping them. Full-Motion animations use high resolution models and rendering just like animated movies, whereas cut-scenes can be represented using the normal game engine features.

### 3.4.3. Effects

Game levels can contain general effects like fire or smoke that only serve as decorative or generally informative purpose. These effects may be running as animated sequences of 3D objects. However, the effects are usually created with particle effect systems that are automatically repeated throughout the whole game.

## 3.5. Textures and Graphics

Texture and graphics designers are usually 2D artists who paint the world according to the guidelines set by the concept art design and art bible. The types of 2D graphics used in games can be divided into groups that serve different purposes of functional and decorative nature.

### 3.5.1. Menu & HUD Graphics

User interface graphics portrayed in menus and HUD are the types of graphics that have both informative and functional role. These graphics define the general look of the game. Menu graphics give the player the basic tools for adjusting the options of the game to fit his or her computers resources, starting or saving the game, etc. The HUD textures have a very important role in giving vital game-related information for the player. Because the functional nature of these graphics, they need to be carefully planned to give the information in as easy to understand and access way as possible. At the same time both menu and HUD graphics have to follow the general visual code of the game.

### 3.5.2. Skins – Textures for 3D Models

All the dynamic and static objects in the game are also painted using textures to fit within the visual theme of the game. For example, material qualities, surface qualities and colours are given to objects with textures. Dynamic objects require special attention in texturing because of their functional role. For example, the character animations change the geometry and the textures have to be flexible enough to cover all the possible changing forms. Generally, the textures are also used to supplement the geometry by imposing a visual detail onto an object without actually creating the complex forms.

### 3.5.3. Textures for Game Levels

Terrain geometry and static models that are used in the levels have a fundamental role in creating the atmosphere of the game world. The final look of the level is usually a combination of many different textures used in carefully selected places on terrain, trees, buildings etc. All the textures follow the visual line designed for the game. The nature of these textures is mainly decorative.

## 3.6. Audio Design

The aim of the audio design task is to produce all audio assets the game needs. These include the music, menu sounds and the in-game sound effects with the possible voice-over dialogue.

### 3.6.1. Menu and Background Sounds & Music

Game worlds usually contain background music that is used during the actual game or in the menus. Music can be used to create strong atmospheric effects that can be separate from the story. Game music tends to follow the traditions of film industry, although the interactivity generally opens up new possibilities and challenges in terms of adaptive soundtracks.

### 3.6.2. Level, Character, Vehicle and Item Sound Effects

The sounds of the game world play a big part in creating the required atmosphere. Sounds of levels will usually be diegetic in nature. They complement the game's visual design. For example, a forest can have sounds of wind, birds, rain, etc. The vehicles have appropriate sounds that are, in addition to generating an illusion of living world, used as informative elements during the gameplay (e.g., speed, proximity, power, etc.)

### 3.6.3. Speech Effects (Voice-over)

Game characters can have communicative elements through speech. Voice-over actors are used to say specified words and lines that will be used in the game. These audio tracks are then integrated to the game as other audio components. Some multiplayer games utilise a specific phrase library which the player can use to replay in-game utterances. Voice acting is also used in cut-scene animations when the story is told by means of cinematography.

## 3.7. Program Code

Programming is one of the most integral parts of the game development. Since the program code binds the whole game together, the task of programming continues from the beginning of the project to the end. The programming can be conducted in various levels of detail – source code or script – but no matter what the approach; the aim of the software is to create the structure and dynamics of the game.

### 3.7.1. Source code

Source code programmers are responsible for writing code for the engine, its tools or extending the features of an existing engine. This code is usually C++, C# or some similar programming language. It depends on the engine how much of the game code is done with source code and how much with scripting.

### 3.7.2. Scripts

Script programmers write script code usually associated with game logic like AI, inventory system, weapon system, scoring system, menus, etc. Engines support scripting on different levels and for example in Unreal engine scripting is also used to program most of the network functionality for game objects.

### 3.7.3. Integration

Integration means importing, linking and adding the art assets into the source or script code. Usually this is done with the help of place holders, so that the coder handling the integration adds some mock-up texture or sound with the right name to the place where the final asset will be. Then the artist can modify the asset and test it in game without any need to contact the coder.

## 3.8. Play Testing

The play testing is a specific area of testing that should be heavily integral to the whole production process. It is a continuous form of assessment evaluating, analysing and validating the playability aspects of the game. Play testing is different from traditional software testing, since it generally targets elements that are difficult to describe in terms if measurable outcomes. It is more like user experience, user satisfaction and usability evaluation. Play testing should be carefully planned, evaluated and reported throughout the project.

### 3.8.1. Bug-hunting and priorisation

In terms of play testing, the bugs that need to be identified are factors, features or characteristics of game that break the playability. Fun, challenge, learning curve, replayability and the core gameplay should be tested and the problems identified. The break points should be listed and corresponding action points created in order to design for change. Some game studios utilise the beta testing as formal method for play testing. It should be noted, though, that testing needs to be started much earlier than when reaching the beta.

### 3.8.2. Reporting

Play testing is useless if the results and design changes are not reported. The design and development iterations need to be followed by carefully tracked testing procedures with the possible cross checks between various versions and different play testers.

# 4. Quality Assurance

Quality assurance overlaps the whole production phase. It focuses on the quality of the product and aims at assuring the best possible quality with the available resources. While play testing is also part of quality assurance work, there are other more measurable criteria to be validated during this task. The most important method for successful quality assurance is a formal review system that covers all aspects of design and production.

## 4.1. Final testing / System testing

The software testing of the game production follows the traditions of software engineering. Testing in this context refers to code and can be conducted as informal tests or formal tests. The testing tasks can be further divided into unit testing, integration testing, system testing, configuration testing, and regression testing. Whatever the subdivision, the testing should be well planned, procedural and efficient enough for the scale of the project.

## 4.2. Alpha and Beta Testing

The Alpha testing starts when the game is said to be feature complete. That means all the planned and designed features are in place and integrated to work as a whole. Some studios execute Alpha testing as a project aiming at feature completion status.

Beta testing can be conducted when the game contain no known bugs. The testing can be organised as open or closed beta testing - with full or partial game being evaluated. It should be noted, though, that Beta testing is not a design phase! Beta testing means that the game is feature-complete, all the known bugs have been corrected, there is no system or feature in the game or technical design document left to add, and no more original design work will be done on this game until after launch, period.

## 4.3. Bug fixes

The outcome of testing is a list of errors or problems (bugs) in the game. It is important to prepare for the fixing procedure by identifying what needs to be fixed, how it is going to be fixed and when it is to be fixed. Generally, the bugs are prioritised based on their impact and importance. This helps to allocate the resources more effectively by targeting the repairs onto the most significant errors first. Furthermore, the bug fixing task should be executed carefully in order to prevent new bugs from arising due to the changes made into the program code.

# 5. Release

The release phase is the climax of the game production – at least from the game developer's point of view. The aim is to get the game finalised and delivered to the publisher for printing and distribution. However, the last phases of production include additional tasks that may require substantial resources.

## 5.1. Master Gold

The aim of working towards Master Gold is the have the final version of the game ready for printing and distribution. This phase generally includes final candidate cycle (or release candidate cycle), where the incremental portions of the game are frozen. At this phase, even the bug fixes are considered risky, since one fixed bug can disturb the balance somewhere else, thus, causing additional bugs.

## 5.2. User documents (readmes, manuals, help-files)

The master version of the game should include all the possible documentation that needs to be shipped with the game. While some of these may be provided by the publisher, there are several that require the attention and knowledge of the development team. Installation guidelines, critical readme files, game manuals and integrated help system should be created during the production by the personnel who is responsible for the particular areas in question. Ideally, the game design document and other project documents should provide the basis for the required material.

## 5.3. Distribution

From the game developer point of view, the distribution relates to marketing material and especially to various game demos. While publisher handles most of the marketing, the developer is generally requested to prepare playable demos, show-reels and other technological demos to support the marketing effort. This work, generally, starts well before the game is finished and can continue well beyond the shipping of the game.

## 5.4. Support services

The support services that require the game developers' know-how are generally related to various technical support procedures. However, some developers participate in other forms of help desk services, especially in the context of on-line games. The support may be implemented in the form of web site, telephone hotlines, or in-game assistance.

### 5.5. Localisation, porting

The localisation and porting task involves changing the game content to suit different market areas and different technological platforms. Text, speech and even user interface symbols may require changes in order to be accessible for a different language group. If the changes are big enough, there is a need to change the underlying architecture as well.

While porting and language localisation may be conducted during the late phases of the project, some of the localisation issues should be considered as part of the initial game design. For example, some countries have cultural differences that mean a game has to be substantially modified in order to be acceptable for release. Furthermore, there may be several important marketing issues that can reflect onto the features, settings, theme, etc. of the game. These cannot be changed overnight, so the initial planning is key here as well.

## *6. Post-Release*

When the game has been released, most of the developers need their time to recover from the hectic crunch time. However, there is one final phase that needs to be tackled before the developer is completely over the project. Post-release phase, again, is an area where other stakeholders generally take responsibility although there are tasks that require the participation of the game developers.

### 6.1. Patches

Patches are separately released correction packages that generally fix the bugs that were not found during the final testing – or were not fixed at the time. Usually the aim is to solve critical problems in game, but sometimes patches are used to balance the gameplay, introduce new content, or to prevent some new-found cheating mechanisms. Patches are currently used in the world of PC gaming since contemporary consoles do not tend to support software installation, and thus, patching is impossible. It could be noted, that good quality game should not need patches in the first place.

### 6.2. Upgrades and add-ons

Sometimes a developers is asked to produce new content for the game, either in the form of upgrades (in the style of patches), or as a specific add-ons (distributed either freely or sold separately). These increase the life-span of a game but are generally less resource hungry than the production of a totally new game.

### 6.3. PR

Public relations and marketing is an important part of the games industry. While publisher generally takes the main responsibility in PR, the developer may be asked to participate in events, such as, game exhibitions, conventions, promotional tours, and other high-profile occasions. Well established developers with substantial track record are seen as valuable marketing and even branding asset.

### 6.4. Community

The game community, in this context, can be seen as two separate areas. The main community for a game consists of the players – customers, end users, fans. The biggest area for game community tasks is the genre of massively multiplayer online games that can be seen more like services than stand-alone products. The players need constant attention and nurturing, but the developer can also learn some valuable information while engaged closely with its target audience.

Furthermore, there is the professional community of games industry people who are keen on sharing their experiences and learning each others trade secrests. Post mortem analyses, game developers conferences, and other forms of knowledge sharing (and learning) are generally used by many game developers.

# Game Production Expertise Areas

Game production requires expertise from several drastically different competence areas. Although the traditionally biggest area, programming, is today a proportionally smaller part of productions than it used to be, the rough division between programmers and artists is about 50-50. The rest of the production team consists of management personnel and designers.

The increased capacity of PCs and game consoles has enlarged the proportion of artistic content creation. This phenomenon becomes even stronger with the introduction of 3rd generation game consoles.

The following figure 4 illustrates the numerous expertise areas involved in game production process. Although this report will not go into details of each these areas, the overview helps to understand the multidisciplinary and multitalented nature of game production.

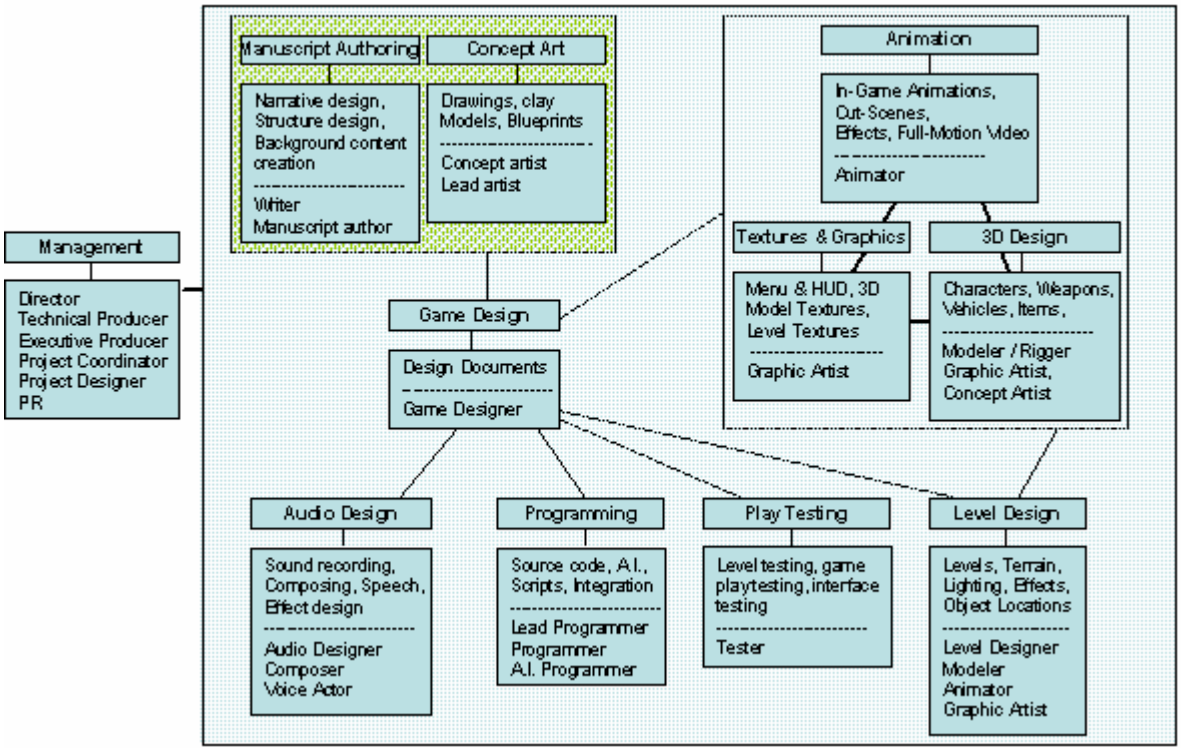

Figure 4. Example of expertise areas required in typical game production.

# Outsourcing and Production Distribution

Outsourcing involves a part of the production to be handed out for third party vendors who will execute the work under the direction of the game studio. This means that parts of the production phases, or tasks, will be conducted by people outside the core development team. Naturally, this distribution of work brings about several new challenges. The quality of work, smooth management procedures and knowledge sharing may be jeopardised if the outsourcing is not planned and executed carefully.

Almost all organisations outsource to some degree. Most publishers outsource game development to game studios. Even those who produce internally their own titles outsource a multitude of tasks, such as, disk manufacturing, payroll management and marketing. Outsourcing makes sense if the game studio does not have the required expertise or personnel to accomplish all the planned tasks.

## *When to outsource*

This report has outlined the numerous expertise areas required by game production. Even the smaller games tend to involve artists, programmers, writers, actors, motion capture experts, voice-over directors, quality assurance engineers and numerous other skilled individuals. It is impossible for a small game studio to employ all this talent full-time and year-around. Therefore the need for outsourcing parts of the development becomes a viable business solution.

The two basic principles related to game production outsourcing concern the questions of when and what to outsource:

1. When to outsource?
   - The outsourcing plan should be prepared in the earliest parts of pre-production phase.
   - Describe what work will be outsourced, by what contractor, by what date, and for how much money.
2. What to outsource?
   - The tasks that are not part of the company's core competence
   - The expertise in production, which is needed only for a short period of time

The biggest exception in the principles of outsourcing is programming. Game studio should never outsource the core programming tasks of game production. A game is a software and if the company does not have the expertise to create the software, it should hire programmers for in-house production. If there are no programmers in the company, the company should not make games – it should concentrate on other competence areas.

## *Check-list for Outsourcing*

The following check-list is intended for project managers and project leads as a set of basic issues to be considered when planning outsourcing or other form of production distribution. Naturally, the much of the same list applies for any game production, whether in-house or out-sourced. The criticality of these issues, however, will increase when the work is allocated outside the core team.

**Check-list for production distribution and outsourcing.**

**Cultural differences**

1. Language and communication
2. Terminology
3. Standard operating procedures (SOPs)

**Contracting**

4. Business considerations
5. Budgets
6. Money allocation, salaries, value-chain
7. Equipment
8. Responsibilities
9. Agreements

**Managing resources**

10. Naming conventions
    - Categories
    - Using numeration
    - Alphabetization
11. Storage conventions
    - File formats
    - File sizes
    - Locating
12. Sharing content
    - Checking material out
    - Building up ideas
    - Getting ideas rejected

# Game Production Outsourcing Case

An example outsourcing case, with the key document and asset excerpts, provides a concrete point-of-view on a simple outsourcing procedure. While the following case cannot possibly illustrate all issues related to outsourcing in general, it offers some insight and practical material to back up the previous sections of this report.

The outsourcing case in question involves a production of 3D game scene with corresponding textures. Furthermore, the project includes one game character without animation sequences. If compared to small-scale PC 3D-game productions, the portion to be outsourced covers only about 1 to 5 % of the complete assets to be developed. This estimate, however, is well below that if considering triple-A productions of contemporary games.

## *Outsourcing Procedure*

The outsourcing procedure involves 1) planning (what to outsource), 2) call for tenders (bidding process), 3) evaluation of tenders (who will offer best suitable results with competitive pricing), 4) refinement for tender (further details and adjustments), 5) contract signing (official outsourcing contract), 6) asset review iterations (quality assurance), and 7) preparing for integration.

### 1. Planning and Specifications

Before a game developer can execute the outsourcing it needs to decide and define what will be outsourced, how the outsourcing work is directed, what are required from the assets and what are the major milestones for the task. The outsourcing plan should be created as early as possible in order to keep the project management clear.

In the outsourcing case, it was decided to start with a preliminary outsourcing round in order to evaluate the possible vendors before initiating the full scale project. This reduces the financial and quality risks, while still providing concrete assets for the final product. Therefore, two separate art assets were defined as main components for outsourcing. These were a game scene comprising of a Kalevala style smithy and one game character without animations.

> "A 3d character model based on a painting by Tamara Jufa (used with permission), and a 3d model of a smithy, based on a small excerpt of Kalevala."

The future third party vendor was decided to have a relatively large degree of freedom in terms of art design, so the outsourcing brief was supplemented with additional material for

references. It should be noted, that this decision makes the corresponding outsourcing case slightly different from the cases where detailed plans, specifications and possible designs are provided for the third party vendor. Outsourcing the design distributes the responsibility and increases the costs. Furthermore, without clear specifications, the outcome of outsourcing may prove to be risky.

## 2. Call for Tenders

After the initial planning, the outsourcing process went on to formulating the call for tender to be sent for the potential third party vendors. The amount of work to be invested in preparing the call can be relatively significant. Especially if the call is to be distributed widely, the more detailed the request, the easier it is to select the most promising partner. Well prepared call for tender, therefore, reduces the risk for inappropriate proposals and makes the next phases much easier to conduct. This call for tender acts as the initial assignment and also fulfils the legal requirements for public bidding procedures.

In the outsourcing case the main request for contributions consisted of 1) sketches of the smithy and 2) a short description of how the vendor would implement the two tasks (i.e., the smithy model and the character model. It was also requested to include any possible 3) limitations of the schedule. Furthermore, each prospective vendor was asked to send a 4) detailed estimation of costs for both tasks.

The final call for tenders is relatively large document that covers the required details of the outsourcing project. See Appendix 3 for the outsourcing call for tenders, which provides an abundant amount of details concerning the outsourcing plan. The original document has been altered only by removing the identification information. All the relevant information is stated as is.

The call for tenders was distributed among a pre-selected group of Russian game studios. Most of these studios had been contacted before, so there was an amount of existing connections to ease up the business procedures. Still, the response for the call was relatively low. Most probably the small size of the project did not raise enough interest among the biggest studios – although it was stated in the call that the first phase is a pilot stage with subsequent continuation projects.

## 3. Evaluating and Selecting the Winning Tender

Based on material provided by the prospective vendors (i.e., the sketches, descriptions, estimation of costs, the track record and the relevant expertise of the vendor), one applicant was selected for the production of the task. The selected vendor was allowed to include 15 hours of sketching work to the financing offer in order to compensate the initial design effort. This method of requiring the vendors to contribute pre-required assets can greatly reduce the risks of outsourcing since the serious parties prove themselves with this initial

investment. However, if the requirement for risky initial effort is too high, there is a possibility of loosing otherwise promising future partners.

Appendix 4 contains the tender from the vendor, which, in this case, is the winner of the bidding process. The tender contains all the original information, except the identity of the vendor. The information is represented as is – even the cost estimates are valid and portray an example of the price-range of Russian game developers.

In addition to the detailed work estimate and descriptions, the selected game studio responded to the call by providing a set of preliminary concept sketches and renderings of the proposed objects and the smithy environment. Figure 5 depicts a set of objects to be modelled as part of the environment, while Figure 6 illustrates the main scene of the smithy. These concrete designs portray a huge amount of information, both in terms of the production and also in terms of the design capability of the vendor.

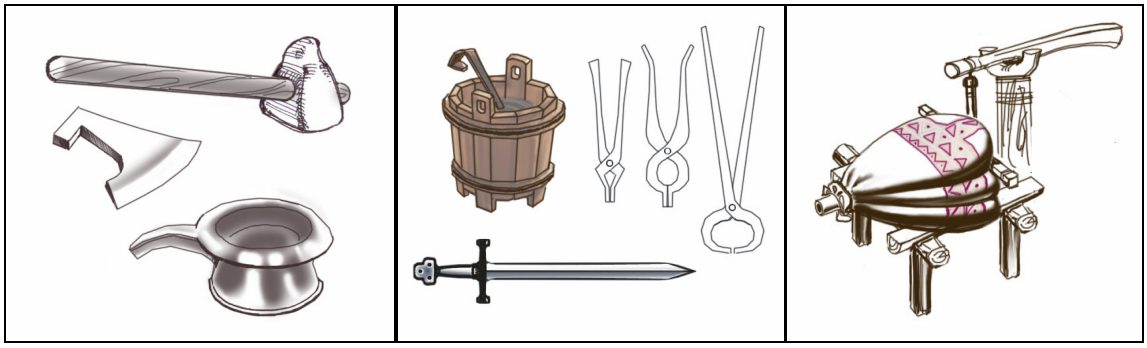

**Figure 5. Object concepts from the game studio**

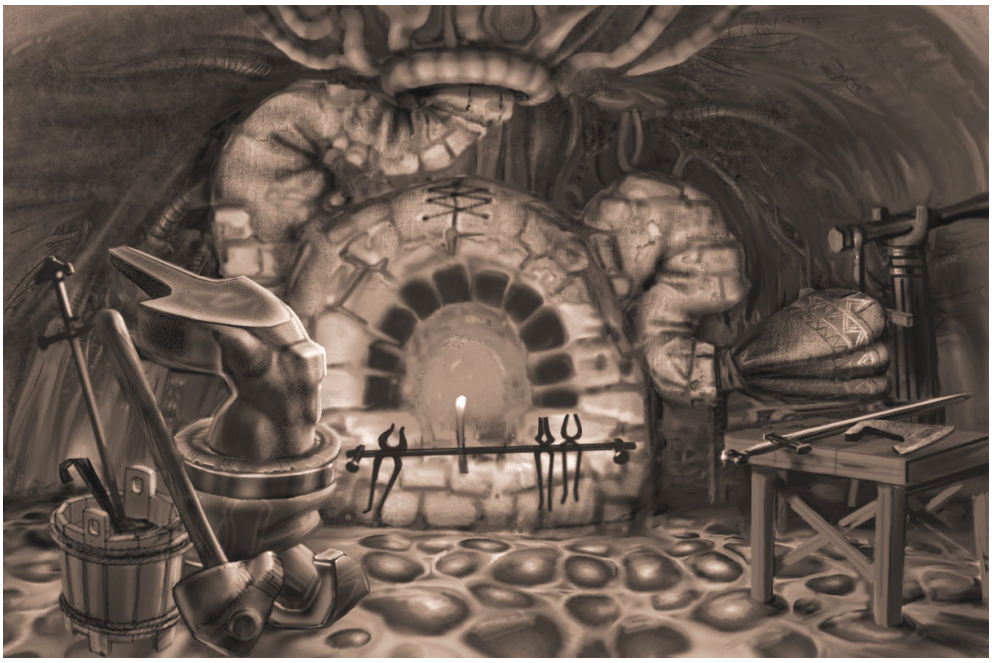

**Figure 6. Layout and look-and-feel of the smith's workshop (smithy).**

In terms of detail, the overall smithy design was relatively well within the scope of the design brief. However, most of the individual objects did not follow the premise provided by the Kalevala excerpt (for example, the hammer does not indicate the use of Väinämöinens elbow as source material). While these are minor details, they should be accounted for in the earliest stage possible. If they go unnoticed through the contracting phase, there is hardly anything the core developer can do – apart from contributing more money.

## 4. Refinements based on the tender

After the selection, agreements and working plans were negotiated with the selected game studio. The tender included some significant suggestions, provided by the game studio, for reducing the costs of production. These enabled the core development team to revise the initial plans accordingly.

The initial material and plans provided by the vendor went through meticulous review procedures among the corresponding members of the core development team. All the further requirements were listed and thoroughly communicated with the vendor. In addition to these, one face-to-face meeting was organised with the vendor. This involved walking through the tender and the corresponding material. Furthermore, the concept designer of the core development team provided the vendor with guidelines and suggestions how to best proceed with the task.

The refinement procedure included further information requests on behalf of the art director working for the vendor. While some of the details may be complicated and slow to tackle because of the distance, they are not impossible if carefully following the procedure. Written questions and answers (e.g. email messages) are excellent, in addition to communication, in documenting and clarifying the details and the process.

An example of further guidelines requested by the vendor (see Figure 6 for reference):

> 1. Smithy model. If you saw the sketch we rendered you may noticed that a lot of curved surfaces are supposed to be created in the smithy: sleeves of the fur coat, socks, the walls (inner surface of Antero Vipunen's stomach), etc. Just to be safe I attached the sketch I am talking about. I suspect that 10.000 triangles won't be enough to create decent looking geometry for such objects. Can we increase polygon budget? If yes, what is the upper limit?

Another example concerning the game character points out the different points-of-view of the vendor and the core development team:

> 2 Questions on the maiden model: 1) It is said that in order to make face animation (2 expressions) we should use at least 30 bones. Why can't we use

> less bones? 2) Should be there any other bones except for facial ones in the model we are to create? I mean should we create a skeleton for the main body and skin the maiden to it?

The aforementioned examples clearly illustrate the need for careful planning and adequate communication of the production. If the vendor misinterprets, or does not holistically understand, the overall design, there is a great possibility of non-usable results. In many cases, it may be beneficial to provide more complete design documentation for the vendor, so that they can see the big picture even if they only focus on small portion of the product.

## 5. Contracting

The contract is a legal document binding the outsourcing parties together. The contract follows standard software industry protocols by stating the scope of the work, deadlines, budget and IPR issues. Appendix 5 provides the complete outsourcing contract with all the details intact. The identities are concealed for the purposes of confidentiality.

## 6. Asset Review Iterations

When the full scale outsourcing production starts, there is a constant need for communication and reviewing the material and designs provided by the vendor. As pointed out by the vendor's representative, the early feedback is the key for successful quality control:

> I am sending the maiden model and the textures we made. Please, look at them and provide us with any comments you have. In order not to do double-work it is important for us to get your comments and fix them before we start to link the model to the bones.
>
> I am sending the smithy model with the textures - the smithy has about 10.000 polys, so we stay within allocated poly budget. I'd like you to look at the smithy and provide us with a feedback: what you like/don't like in the scene, what you want us to change/remove/add - both in terms of geometry, textures and lighting. Any comments are highly appreciated.

The aforementioned requests were accompanied by a set of 3D models. This material formed a first round of asset creation and, therefore, the vendor was making sure they are good to go to the second iteration phase. This may not be the standard operating procedure of all the game studios, so it should be defined in the call for tenders and the contract. Same rules for iterative work comply in outsourcing – and some may be even more critical (e.g., review procedures, change plans, milestones, etc.)

A detailed review analysis and feedback report, based on the two previous requests is attached as Appendix 6: Outsourcing CASE: Review Comments.

The asset review procedure should include full-scale testing and evaluation with the core game engine. While the technical specifications are generally easier to convey and to conform, it is best to conduct thorough tests before signing off the outsourcing project. If the vendor cannot produce iterative and integration-ready versions of assets, the review procedure should include the integration testing as final step.

### 7. Integration Issues

When the outsourcing project is completed, it is time to integrate all the produced assets into the core product. If everything was planned well, clearly specified and carefully reviewed, the material should correspond to the standards and guidelines set by the core development team.

However, the project planning should account for certain amount of overhead in terms of personnel resources. The integration process may reveal some hidden surprises that require further work. In any case, the outsourced material needs to be signed-off by the contractor. The sign-off review should be specified in the contract, or in the attached documents, in order to prevent final misunderstandings of whether the work is complete or not.

## *Lessons learned*

The outsourcing case provided a set of production planning and design implications to be considered for future projects. While the material analysed was limited in scope and volume, the indications can be applied into larger productions.

First of all, dealing with professional vendors, there is a great possibility that they actually help you to refine your designs and production plans. Although the professionals may pardon your mistakes, it is always risky if you do not know exactly what do you want. The communication of plans and requirements should be extensive enough to allow shared understanding of the outsourcing project and the required outcomes.

You should always clarify the milestones, iterations, and sign-off criteria for the material or components that are outsourced. Make sure your vendor agrees with them before proceeding to the productions. Even with the most careful and well-planned initial correspondence, you should be prepared to communicate in regular basis. If you have a chance to visit the studio, then do so and do it often.

Whatever the case, never assume anything before seeing the final results. There may be some unanticipated last-minute changes or misunderstandings that can lead to a major disaster if the products are signed off before careful review and validation. Therefore, all the material should be evaluated and the results should be thoroughly validated before signing off.

Outsourcing is a feasible way to increase your company's production volume and competence. While there is always an element of risk involved in the outsourcing, careful planning and thorough communication can take you a long way. And, with established partnerships, the future outsourcing projects tend to improve since both parties learn something about each others working methods and culture.

The case illustrated in this report is relatively limited, so there is a need to conduct further studies both in terms of volume and in terms of outsourcing areas. The aforementioned results, however, provide some insight into the world of game production outsourcing. There is large number of competent game studios in Russia alone, so the potential for fruitful partnerships should be tapped. By joining the forces, it is possible to focus more tightly on core competence and, thus, increase the overall productivity of the company.

# Conclusions

This preliminary study, although limited in volume and scope, illustrates a general game production process model that can be used as a guideline for various purposes. The industry seems to lack a standard production process model, since most of the game developers follow their own tried and true methods. There is, however, a need for clear and adaptable game production process model. Start up companies, students, aspiring game professionals and many small and medium enterprises can use the model illustrated in this report as a starting point in developing their own production pipelines.

Naturally, the generality of the model reduces the level of detail. All the different phases of the production process could be described and analysed in more detail. However, more evidence is still needed for the validity and exhaustiveness of the model before it is viable to go deeper into the details. Further refinements are necessary in order to cover the heterogeneous industry processes.

The game production process model offers a high level structure that can be easily modified. The brief descriptions of each production task are indented to provide enough information to satisfy the basic needs for knowledge. The future work includes adding more details into the descriptions. Furthermore, concrete illustrations will be provided in terms of empirical anecdotes and journal entries combined with images depicting corresponding material.

The next version of this report will be published by *LudoCraft* after we have conducted a round of further research. Feel free to contact the main author for more details and discussion about the background and future plans of this study.

# References

Bethke, Erik (2003) Game Development and Production. Wordware Publishing, Inc., ISBN: 1556229518, p. 412

Crawford, Chris (2002) Chris Crawford on Game Design. Prentice Hall, ISBN: 0131460994, p. 300

Fullerton T, Swain C and Hoffman S (2004) Game Design Workshop: Designing, Prototyping and Playtesting Games. CMP Books, San Francisco, USA.

Grassioulet Y (2003) A Cognitive Ergonomics Approach to the Process of Game Design and Development. In Proceedings of DiGRA2003 Conference - Level Up.

Grossman, Austin (2003) Postmortems from "Game Developer". R & D, ISBN: 1578202140, p. 448

Mulligan, Jessica & Patrovsky, Bridgette (2003) Developing Online Games: Insiders Guide. Prentice Hall PTR, ISBN: 1592730000, p. 495

Rollings, Andrew & Adams, Ernest (2003) Andrew Rollings and Ernest Adams on Game Design (NRG S.). New Riders, ISBN: 1592730019, p. 500

Rollings, Andrew & Morris, Dave (1999) Game Architecture and Design. Paraglyph Pr, ISBN: 193211131X, p. 742

Rouse, Richard (2001) Computer Game Design: Theory and Practice. Wordware Publishing, Inc., ISBN: 1556227353, p. 500

# APPENDIX 1: Synopsis Template

## Synopsis [Template]

Tony Manninen
v.2 - 14.12.2005

The synopsis should be no more than 1-2 pages long. Try to capture the essential aspects of the concept in a way that makes it easy for the reader to understand your brilliant idea. Remember to be illustrative and try to show your enthusiasm. The headers can be omitted if wanted, but try to keep the structure clear in order to make it easy to access the relevant bits.

**Description**
- High-concept (a punchy sentence, slogan or set of keywords that best capture the concept of the game)
- Brief description of the game - maximum ½ page overview

Answer questions like: What is the game? What are you working on? What are the genre and the main theme of the game?

**Philosophy**
- Main philosophy behind the game design
- Is it a concept, gameplay, content, theme, user interface, technology, etc., or bits and pieces of them all?

**Why create this game?**
- Why are you creating this game?
- Motivation, background, vision?
- Why is this game worth the effort?

**Target Audience**
- Who are the players?
- What makes this group important?

**Treatment**
- How the game is going to be built?
- What are the main methods, tools and processes?
- What is the basic technological platform?

**Key Ideas**
- Unique selling points (USPs) and killer ideas
- Most important features
- What's different when compared to the games in market?

# APPENDIX 2: Brief Game Design Document Template

## Brief Game Design Document [Template]

Tony Manninen
v.2 - 14.12.2005

This template provides an example for the structure of game idea document. The overall number of individual chapters makes the document rather long. Therefore, the designer should pick and choose the applicable sections for his or her initial idea description (pitcher, treatment, synopsis). The order of sections is not necessarily from the most abstract to the most detailed one, although this has been the main philosophy in arranging the list. However, the designer may want to choose slightly different set of sections, or describe them in different order, depending on the situation. The brief game design document should not be longer than 10-20 pages. Illustrations and figures should be used to ease the access.

**Description**
- Brief description of the game - maximum ½ page overview
- Will form the basis of the documentation later
- What is the game? What are you working on?
- What is the high concept of the game - the basic concept?
- Genre, theme

**Philosophy**
- Main philosophy behind the game design
- Is it concept, gameplay, content, technology, or bits and pieces of them all?

**Why create this game?**
- Why are you creating this game?
- Motivation, background, vision
- Why is this game worth the effort?

**Target Audience**
- Who are the players?
- What makes this group important?

**Treatment**
- How the game is going to be built?
- What are the main methods, tools and processes?
- What is the basic technological platform?

**Key Ideas**
- Killer ideas
- Most important features
- What's different when compared to the games in market?

**Gameworld and Level Design**
- Where does the game take place?
- Describe the world that your game takes place in.
- A description of the various levels in the game (?)
- The style of the levels, their main characteristics, number of levels, …
- How does gameworld/set of levels affect the game and gameplay?

**Look and Feel**
- Style of the game (artistic point of view)
- What does the game look like?
- Main artistic sources (books, movies, music, etc.)
- What games are used as main sources for ideas?

**Background Story**
- If applicable, tell the story of the game that leads into the beginning of the game, and tell the story that unfolds during gameplay, if any.
- What is the tone?
- What is the basic narrative?
- What is the "heart" of the story? Is it a linear story?

**Objectives**
- Describe the objective of the game.
- What is the main focus?
- What is the player supposed to achieve in this world?
- What is the player's goal and why would he or she want to accomplish it?

**Rules**
- Basic rules and rule sets (which are known at this point)
- What are the core rules that make the game as it is?
- Evaluation of other rules (their estimated functionality, possible problems, etc.)
- What are the supporting and/or possible additional rules that enhance the game?

**Gameplay**
- Describe the way the game works, from beginning to end.
- How do the rules affect the features?
- Describe the basic logic on general level
- Describe the A.I. of the computerised opponents/characters/objects, if any.
- It is sometimes helpful to write a "walkthrough" of the game to further enhance the reader's ability to visualise the game.
- What is the basic interactive structure? (e.g. Chapters vs. Great Middle Section, Levels, etc.).
- What is the "heart" of the gameplay? (e.g. speed, actions, style, continuous, turn-based, etc.?
- How does multi-player work?
- How difficult is the game?
- How long will it take the average player to complete?

**User Interface**
- A description of the user interface
- What is the planned interface?
- What is the planned perspective (1st person vs. 3rd person)?
- What does the player control?
- How many characters does the player control?
- Interface mechanisms and techniques
- Session control (saving, loading, instructions, etc…)
- Some examples

**Entities (players and computer)**
- Units, objects, characters, resources, …
- Number of entities

**Start up**
- Beginning of the game/levels (resources, locations, situation, etc.)
- Possible choices

**Feature List**
- A list of all the features (enemies, power-ups, special stones, resources, etc.), with information on what their function is and where and how they appear in the game

**Marketing**
- USPs (Unique Selling Points)
- Market analysis, sale estimations, …

**Technical requirements**
- Various versions
- Platform(s)
- Requirements (graphics, audio, processor, …)

**Planned resource allocation**
- What skills are needed to develop the game?
- How many man months are estimated for production?
- Estimated duration of the production?

# APPENDIX 3: Outsourcing CASE: Call for tenders

*ELIAS -project* *CALL FOR TENDER*

Dep. of Information Processing Science
P.O. Box 3000
90014 UNIVERSITY OF OULU
FINLAND

03.11.2005

**SMALL SCALE 3D PRODUCTION OF KALEVALA, FINNISH NATIONAL EPIC – TESTING THE POSSIBILITIES OF OUTSOURCING**

Issued November 3th 2005, valid until November 18th 2005

**BACKGROUND FOR ELIAS PROJECT**

ELIAS- Game the Border Region is a preliminary study of competence, knowledge-base and business potential of computer game research and production in the Finland-Russia border areas. The project aims to establish a network of computer game design expertise, emphasizing co-operation between Finland and Russia.

This network will be build upon a solid research background and a strong knowledge of business practices. The expertise of this network would serve the needs of education, cultural institutes, travel industry and mainstream game business, among others. The expected result of this project is an expanded view of the applications of game development and marketing: Games are not only a tool of entertainment, but can cover a much wider field of applications, such as training, cultural experiences, historical simulations etc.

http://www.ludocraft.oulu.fi/elias

**BACKGROUND OF THIS CALL**

ELIAS will serve as a basis for game-oriented digital media projects, commencing from the summer 2006 onwards. This call will be a small-scale test of outsourcing, with the aim of finding the right partner(s) from the field of Russian game development. The material produced through this call will serve the interests of later projects and productions, and serve as a rudimentary test of co-operation procedures.

**ABOUT THE CALL**

Attached to this call are detailed specifications to two tasks: A 3d character model based on a painting by Tamara Jufa (used with permission), and a 3d model of a smithy, based on a small excerpt of Kalevala.

By November 18th, each applicant should send us a sketch, or sketches (at least one required) of the smithy and a short description of how they will implement these two tasks - this should include any possible limitations of schedule. Each company should also send a detailed estimation of costs for both tasks (see specifications for more details).

Based on these sketches, descriptions, estimation of costs, the track record and the

relevant expertise of the company, one applicant will be selected for the production of the task. The selected company can include 15 hours of sketching work to the financing offer. After the selection, agreements and working plans will be negotiated with the selected company. Note: the production of these tasks must be accomplished by December 31st (if this is not possible, mark the earlier possible time you could finish the project in your proposal). If the work is satisfying for both partners a co-operation for near-future projects will be considered.

**NOTES**
By November 18th 2005, send the application to:
[redacted]
The subject line should read "Tender – confidential"

The proposal should be written in English and the estimated prices should be given in euros. Include a preferred e-mail contact address in the proposal.

We recommend you send a confirmation e-mail to: [redacted] that your proposal has been sent. We also suggest you mark your sent proposals to be confirmed upon reading. The ELIAS-project accepts no responsibility of applications that are lost either in the mail or due to a technical problem.

The winning proposal will be selected by November 25th 2005; all applicants will be notified of the selection during the selection day (by e-mail). Thank you for your interest.

## Attachment 1: Specifications for the character model

**DESCRIPTION**
The source material for the character model is the painting *Aino and the Maidens of the Lake*, by Tamara Jufa (used with permission). The painting can be accessed here: [redacted] (includes a high resolution image of the painting in TIF-format). Note: do not distribute this link publicly, as the permission to use these paintings is to this project only.

The character model should be based on the foremost and most visible maiden figure in the painting (the three characters on the left), adhering to the general outlook, and keeping the details as closely as possible. Invisible areas should be produced in the spirit of the painting.

**GENERAL GEOMETRY**
· Polygon count: around 10 000 (triangles)
· File format: Either Maya or 3ds Max native
· Clothes separate from the body (as a separate, rigged mesh)
· Compatible with actorX (http://udn.epicgames.com/Two/ActorX)
· Compatible with Unrealengine 2.X
(http://udn.epicgames.com/Two/SkeletalAnimation for details)

**CHARACTER RIG**
· A fully functional rig included (naming hierarchy will be provided when contracted)
· Facial structure for animation made with bones (minimum of 30), not with Morph Targets (include two expressions: surprise and frown, and a mechanism to control them)

· All control scripts included (Either Maya or 3ds Max based)
· Compatible with actorX (http://udn.epicgames.com/Two/ActorX)
· Compatible with Unrealengine 2.X
(http://udn.epicgames.com/Two/SkeletalAnimation for details)

**CHARACTER TEXTURES**
· 3 texture maps: head, body, clothes
· Texture base maps (unwraps) included
· Texture resolution: 2048x2048
· Texture file format: psd (preferred) or tga

**NOTES**
Give a detailed cost estimation for the work – you can also suggest options not included in the specifications. Also include a cost estimation for the concept sketches and blueprints used in modelling the character.

## Attachment 2: Specifications for the building model

**DESCRIPTION**
The model should be based on the excerpt from Kalevala (the Finnish national epic), included below. The excerpt describes a smithy, magically build by Väinämöinen when trapped in the stomach of Antero Vipunen, using his clothes as building material.

When designing the model, take the following requirements into consideration:

· The smithy should be a composite model, formed from 4-5 separate parts (separate meshes and texture maps for each part). Positioned together these separate parts form the smithy as a whole. Also, produce 2-3 small, minor objects (separate meshes and texture maps for each part) as items made in the smithy. Such items are not detailed in the text – try to design them in the spirit of the smithy as a whole.

· The smithy should have an interior space and an entrance (room for at least one character).

· Väinämöinen builds the smithy with magic. Therefore, the clothes from which he makes the smithy can and should change in shape and form when transformed into building parts. The smithy should not be realistic – try to maintain the surrealistic elements of the text.

· There is no "right" way of envisioning this smithy. As a general guideline, the setting of Kalevala could be described as primitive iron age with strong presence of natural elements (forests, lakes, animals) – the atmosphere being more mythical than realistic. Also typical for Kalevala are transformations and interminglings of different elements (humans to animals or plants, dead to living, small to big etc.). Try to avoid clichéd fantasy style, typical of many fantasy games.

· The last line in the excerpt concerns Väinämöinen using his elbow and knee as a hammer and an anvil. This line can be used as an inspiration for designing the interior, but the character Väinämöinen need not be modeled.

The excerpt in Russian (Mishin-Kiuru translation; A longer version, giving more context, can be found from here: **http://kalevala.gov.karelia.ru/songs/song17.shtml**):

Вот тогда-то Вяйнямейнен
кузнецом оборотился,
стал кователем железа.
Сделал кузницей рубаху,
рукава - мехами горна,
поддувалом - полушубок,
сделал трубами штанины,
раструбом чулок устроил,
наковальнею - колено,
молотом - свой крепкий
локоть.

English translation for comparison (by Eino Friberg):

*Whereupon old Väinämöinen*
*Turned himself into a blacksmith*
*And became an iron worker;*
*Turned his shirt into a smithy:*
*From the sleeves he made the bellows*
*With his fur coat as the blowers;*
*From the pant leg made the air pipe,*
*From his sock the mouthpiece for it;*
*Set his knee up for the anvil,*
*Used his elbow for a hammer.*

**GENERAL GEOMETRY**
· Total polygon count (sum of all parts): around 10 000 (triangles)
· File format: Either Maya or 3ds Max native
· Separate meshes for each part

**TEXTURES**
· Separate texture maps for each part, base maps (unwraps) included
· Texture resolution: 2048x2048, or larger if necessary
· Texture file format: psd (preferred) or tga
· Avoid texture clipping or clearly visible seam lines where parts of the model are joined.

**NOTES**
Give a detailed cost estimation for the work – you can also suggest options not included in the specifications. Also include a cost estimation for the inclusion of the blueprints used for the model of the smithy.

# APPENDIX 4: Outsourcing CASE: Tender from the vendor

**ELIAS-project**
**Tender November 18th, 2005**

**TASKS DESCRIPTIONS AND ESTIMATIONS**
We used specifications for two tasks in the Call for Tender of 3.11.2005, as well as our experience in game development.

## 1. The character model estimate

A model creation involves:

1. A concept sketch rendering. In our case it is not needed since we are already provided with it. If the maiden is not supposed to get undressed during the course of the game we can create pretty generic body and its texture concentrating mostly on the head and the robe.

2. Blueprints (orthographic views). Since it is supposed that the body is concealed by the robe and is not visible we need blueprints for the head only. It will take **0.5** day to render it.

3. Geometry creation and applying UV coordinates.
3.1 Body model – about 500-1000 triangles. That will take about **0.5** day because we will just adjust one of already existing models.
3.2 Robe model – about 4.000 triangles – about **0.5** day
3.3 Head model – about 5000 triangles - about **3** days

4. Textures rendering
4.1 Body
Taking into account that nobody will see the maiden's body it does not make sense to render such a huge texture as it was specified (2048x2048) for it. The only parts we will see are hands and toes and it will be enough to have two 256x256 textures for them.
Time estimate: **1** day
4.2 Head
It takes at least **7-8** days to render detailed texture of 2048x2048
4.3 Rig
It takes at least **7-8** days to render detailed texture of 2048x2048
5. Skeleton creation and rigging the model – **1** day.
6. Facial bones set up and creation for two expressions - *surprise* and *frown* - **1.5** days.

Total: **22 - 24** man-days.

**Note:** Are you sure we should render textures of such a big size? The most of RPGs do with 1024x1024 textures – they have enough details to ensure very good visual quality even for short distance views. In this case it will take about **4-5** days instead of 7-8. Reducing the texture size would expedite the process a lot! If this is acceptable, the task would take **16-18** man-days.

One artist's man-day in ***<Game Developer>*** costs on average 100 euros. (This is overall cost including salary, taxes, equipment, internet, overheads, etc.)

Total cost for the character model:

- In case of huge (2048x2048) textures - **23** man-days, i.e. 2,300 euros
- In case of normal (1024x1024) textures - **17** man-days, i.e. 1,700 euros

## 2. The building model estimate

A building model creation involves:

1. Concept sketches:
1.1 Collecting reference material. In our case it is reference on the look and design of blacksmith tools, reading Kalevala, etc. It took **2** days.
1.2 Rendering concept sketch itself. It took 1 day to render black-and-white sketches and 2 days to make it colour – **3** days in total.

In our case this work is already done (see attached sketches of the smithy).

2. Geometry and mapping.
It will take about **5** days to model the geometry and UV-map it.

3. Textures rendering
It is supposed in the specification to create 4-5 pieces of geometry and render 2048x2048 texture for each piece.
It takes at least 7-8 days to render one detailed 2048x2048 texture. So, rendering all textures for this single room will take about **40** man-days.
Again, in our opinion, it is unnecessary to render such a big textures. It will be enough to render 3 textures of 1024x1024 (for big unique objects as the ceiling or furnace) and 4-5 textures of 512x512 for smaller objects or for the cases when the textures can be tiled. In addition, if we are going to use Unreal engine it will allow us to use “detail” textures – small tiling greyscale ones which show “texture” of the material when the player approaches close to the object. In combination with regular textures they are able to ensure photo realistic quality.
If this is acceptable, the task will take about **20** days instead of 40.

Anticipated question: you might be wondering why rendering 2048x2048 texture takes 7-8 days whereas doing 1024x1024 one takes 4-5 days? – 1024x1024 texture is as much as 4 times smaller and arithmetically it should take 2 days, not 4…
Answer: There is no linear dependency between the texture size and its men-hours because in any case some time is needed for the artist to design the texture.

4. Applying lighting – **1** day.

Total cost for the building model:

- In case of huge (2048x2048) textures - **46** man-days, i.e. 4,600 euros
- In case of normal textures - **26** man-days, i.e. 2,600 euros.

# APPENDIX 5: Outsourcing CASE: Contract

**SUBCONTRACT**

**1 Contracting Parties**

The University of Oulu, Department of Information Processing Sciences (hereinafter ”the Client”)

[redacted]. (hereinafter ”the Supplier”)

**2 The Purpose and Scope of Application**

2.1 These Contract Terms shall be used in the contractual relationship between the Client and the Supplier concerning the subcontracting of Project “ELIAS”.

2.2 The Client is committed to the implementation of the project ”ELIAS" Contracting parties of the project have agreed upon that the Client may use subcontractors for a part performance of the project.

2.3 Subcontract or responsibilities therein cannot be transferred to a third party without written consent from the contracting parties.

**3 Definitions**

3.1 *Ownership* refers to patents, financial copyright and other intellectual property rights/incorporeal rights in addition to the right in rem.

3.2 *Pre-existing know-how* refers to any technical, commercial, economic, or financial information in the actual or relevant field of the project that is owned by the contracting party and developed outside the project.

3.3 *Knowledge* refers to all the information and the results that the contracting party, or a third party working on behalf of the contracting party, has achieved in the project, and patents, inventions and computer software (including source codes) and other intellectual property rights.

3.4 *Tender* refers to a called proposal for and accepted by the project “ELIAS”.

**4 Schedule and Performance of the Subcontract Work**

4.1 Subcontract work includes tasks prescribed in tender (Appendix 1).

4.2 The Supplier shall perform the tasks prescribed in the tender by 15.2.2006.

4.3 The Client transfers the material needed for the completion of the task to the Supplier.

4.4 The Supplier makes certain that it has the staff, premises, equipment and material needed for the completion of the tasks defined in 4.1.

4.5 The performance of the Supplier is supervised and guided by a contact person appointed by the project manager of the project ” ELIAS” or the Client.

4.6 The Supplier must immediately notify the Client about any circumstances that may endanger the carrying out of the performance, delay it or cause unforeseeable costs.

4.7 The performance is regarded as completed when the Supplier has met the quota of time as defined in the tender, or the tender expires. The Client may ask for extending the quota if so defined in the tender.

**5 Invoicing and payment schedule**

The research is invoiced at a combined contractual price of 4300 EUR, consisting of two tasks, 1700 and 2600 EUR respectively, as detailed in the Appendix 1. Lump sum of 4300 EUR will be paid on 15.2.2006 upon the completion of the performance.

**6 Ownership and Utilisation of the Knowledge**

6.1 Knowledge, including inventions, patents, copyright (with forwarding and modifying rights) and other intellectual property right such as reports, accounts, parts of software (with source codes) and documentation, are the Client’s property.

6.2 Such pre-existing know-how (computer software, equipment, documents, reports, patents, inventions) that the Supplier provides for the use of the project, is not regarded as knowledge from the project herein.

**7 Confidentiality**

7.1 The Supplier shall not give any confidential information or documents received from the Client during the research to a third party. The Supplier shall not use any confidential information to any other purposes than to the carrying out of the tasks prescribed in the contract herein. Especially all financial or technical information in oral, written or electronic form, including knowledge and research plan, which is transferred to the Supplier by the Client, is regarded as confidential information.

**8 Force Majeure**

8.1 An incident that prevents or makes it unreasonably difficult to carry out the Project within set deadline is regarded as force majeure. Such incidents include war, mutiny, natural disaster, general break in energy supply, fire, significant limitation to the operations of the university caused by the state budget or set by the Finnish government, strike, boycott or any other significant or unusual cause that is beyond the control of the contracting parties.

8.2 Neither contracting party is responsible to the other party for any damages or delays caused by force majeure.

**9 Termination of the Contract**

9.1 If the contracting party commits an essential violation of the Contract Terms, the other

contracting party has the right to terminate the Contract. A notice or termination shall be submitted to the violating contracting party in writing without delay.

9.2 Both contracting parties shall have the right to terminate the Contract, if, owing to force majeure, the completion of the Contract is impossible, or the implementation of the contract is delayed significantly or more than six (6) months.

9.3 The Client shall have the right to terminate the Contract, if the Supplier is put into liquidation, financial or operational restructuring, or declared bankrupt.

9.4 In case the Project Contract is terminated, the Client shall pay for the part performance conducted until the date of termination as per the payment schedule of the Contract, or if the contracting parties agree that work will continue after the date of termination, until the date of termination. The knowledge achieved by the date of termination is property of the Client.

**10 Responsibilities of the Supplier**

10.1 The Supplier is responsible for implementing the research assignment as per the Contract and the appendices thereto. The assignment shall be implemented with due care, using professionally skilled staff and suitable methods.

10.2 The Supplier is responsible for any direct damages caused to the Client by breach of this Agreement.

10.3 The Supplier guarantees that to the best of its knowledge the work the Supplier performs, defined in section 4.2, its results or the use thereof, or transferring them forward, does not infringe any patents, copyright, registered models or other intellectual property rights belonging to a third party.

**11 Governing Law and Disputes**

11.1 In case of dispute or difference of opinion between the parties arising out of or in connection with this Contract, the parties shall first endeavour to settle it amicably. All disputes arising in connection with this Contract shall be submitted to the Oulu District Court (Oulun käräjäoikeus). Law of Finland (excluding its principles on conflict of laws) shall be applied to this Contract.

**12 Appendices**

12.1 In case there are any discrepancies between this Contract and Appendices thereto, the applying order is as follows: 1) this Contract, 2) Appendix 1, 3)Appendix 2.

This Contract has been drawn up in two identical copies, one for each contracting party.

**Signatures**

University of Oulu, Finland _____/_____ 2005

_____

[redacted] Department of Information Processing Science

_____

[redacted] Department of Information Processing Science

[redacted] _____ / _____ 2005

_____

[redacted] (The supplier's representative)

Appendix 1: [redacted] Tender
Appendix 2: ELIAS-project, Call for Tender

# APPENDIX 6: Outsourcing CASE: Review Comments

**The Maiden-model:**

Generally, we think you have captured the colour-scheme quite well, with a few exceptions. The textures are generally good, although the shoes are a bit problematic. Their texture is too generic and bland, and stands too much in contrast with the rest of the model: You might try to texture them similarly with the clothing.

Also, the texture area for the shoes is too small: This leads to one pixel being stretched over the whole top of the shoe. The tip of the shoe – with its sharp, polygonish shape - looks likewise problematic, especially given the high-polygon roundness of rest of the model.

The face is a bit angular and the chin and the cheekbones tend to protrude in a bit elongated way (fairly noticeable when rotated in the perspective view). Also, the colour-scheme of the face is too natural, when the faces in the painting are bluish and pale. The eyes are too big and blue, too prominent - both their colour and style clashes with the painting's unnatural, timid faces somewhat.

Also, the eyeballs are too deep in their sockets, one can see through to the inside of the skull, while looking from the side.

Of the sleeves: I know that the sleeves in the painting are quite muted in the colours - but you could try to give the patterns on the sleeves more prominence and contrast, since in the 3d-model, the sleeves behave differently, and hence look too uniform in their prominent presence.

The veil very well done - it realises the back of the model very much in the spirit of the painting; commendations. The head-dress is likewise successful - in both the veil and the head-dress we especially liked the "floating" border-ornaments - they have a similar, ghostly ambience as the painting.

Lastly, one major technical point:

The character is currently missing a body mesh; we were in understanding that separate meshes for the body and clothing would be created, in regards to our previous discussions, relevant parts quoted below:

On Tue, 6 Dec 2005, [redacted] wrote:
> OK, we will create different sets of bones for the body and clothing, say - biped (character studio for 3ds max) for the body and separate sets of bones for the clothing and the face linked to the biped. Is it OK for you?

Overall, these comments may seem more critical than they should, as they concentrate mostly on the issues we would like to give attention to. Overall, we think that the model is proceeding in a good direction, especially given the constraints of the task.

**The Smithy:**

First of all, I should stress how generally pleased with the smithy we all are. We think you have really succeeded in realising the vague text through interesting interpretations and solutions, yet keeping the general feel of the smithy coherent, so that both the general impression and the longer inspection are interesting in themselves; commendations are certainly in order.

Couple of small notices, though: Most of the small pots and pans are one-sided. The water texture looks quite artificial and its brilliant blue clashes with the colours of the dark and earth-coloured smithy.

The fireplace texture is somewhat grainy, and the colours are a bit artificial - perhaps you could add some coals in there. Also the edges of the fireplace are a bit sharp and polygonish. Lastly, the tongs in front of the fireplace are understandably low-polygon; however, their prominent place (centre, in front of a strong light) makes their simplicity too noticeable - you might think of adding a bit more polygons to them, so they won't stand out.

But there's much more to like than not to like: We liked how you solved the knee/anvil - it was a tough one to solve without a character model, but your solution has visual strength, and making the other shoe a cooling-bucket gives the set-up a sort of logic of its own. Also, you managed to use the shirt in good visual correspondence with the stomach-environment (the sleeves remind the viewer of intestine-like shapes) - with this, both the surrounding environment and the contraption support each other well.