

Riikka Kilpeläinen

# Game Graphics: Achieving the Goals of Pre-production

Bachelor of  
Business Administration  
Technology  
Autumn 2020



KAMK • University  
of Applied Sciences

## **Tiivistelmä**

**Tekijä:** Kilpeläinen Riikka

**Työn nimi:** Peligrafiikka: Esituotannon tavoitteiden saavuttaminen

**Tutkintonimike:** Tradenomi (AMK), tietojenkäsittely

**Avainsanat:** pelit, esituotanto, pelituottaminen, grafiikka, peligrafiikka

Opinnäytetyön tarkoituksena oli tutkia työkaluja ja -tapoja, jotka ovat yleisesti käytössä pelituotannon esituotannossa grafiikkatiimin toimesta. Opinnäytetyön lopussa opittu teoria laitettiin käytäntöön ja suunniteltiin grafiikkatiimin tehtävät esituotannossa ensimmäisen persoonan kauhupelille.

Ensimmäinen osio toimii johdantona pelien tuotantocykliin kokonaisuutena. Se antaa kontekstin esituotannolle ja avaa sen tarkoitusta pelituotannossa. Opinnäytetyön keskittyessä esituotantoon, se käytiin myös syvällisemmin läpi myöhemmässä osiossa. Osiossa käytiin läpi kysymykset, joihin esituotannon on tarkoitus vastata ennen kuin kehitystiimi siirtyy tuotantoon.

Erilaisia dokumentteja ja niiden sisältöä ja tarkoitusta tutkittiin. Eroavaisuudet usein käytettyjen dokumenttityyppien välillä käytiin läpi ja ne demonstroitiin esimerkkien avulla. Opinnäytetyö paneutui tämän jälkeen syvemmin grafiikkatiimin esituotannon toteuttamisen yksityiskohtiin. Tutkittiin versionhallintaa ja sen käyttökohteita sekä eri tekniikoita, joilla saadaan tuotettua grafiikkaa tehokkaasti.

Lopuksi opittu teoria laitettiin käytäntöön ja suunniteltiin grafiikan esituotanto meneillään olevaan kauhupeliprojektiin. Suunnitelmassa otettiin huomioon kehitystiimin koko, tiimin aikaisemmin tekemät suunnitelmat ja muut tiimin asettamat rajoitukset. Tehty suunnitelma auttaa kehitystiimiä saavuttamaan tavoitteensa pelin graafisessa tyylissä sekä nostaa esille mahdollisia ongelmia ja haasteita joihin tiimi saattaa törmätä. Opinnäytetyön teoriaosuus toimii oppaana kelle tahansa graafikolle, tiiminjohtajalle tai tuottajalle, joka haluaa suunnitella ja toteuttaa esituotannon huolellisesti.

## **Abstract**

**Author:** Kilpeläinen Riikka

**Title of the Publication:** Game Graphics: Achieving the Goals of Pre-production

**Degree Title:** Bachelor of Business Administration, Business Information Technology

**Keywords:** games, pre-production, production, graphics, game graphics

The purpose of this thesis was to study the tools and workflows used in pre-production in video game development from the perspective of the art team. In the final part, this theory is put into use as the plans for executing the pre-production of a first-person horror game are built.

The first section functions as an introduction to the game production cycle. It gives context to pre-production and what informs the reader about the other stages of game development. As the thesis focuses solely on pre-production, it is covered in depth in a latter section. In this section, the purpose of pre-production is explained. What questions should be answered in pre-production and what limits the development team may have when entering production.

Then different types of documents are covered, as their function and contents are studied. The differences between the documents often used are covered and examples of documents are shown and briefly examined. Then the thesis delves deeper into finer and finer details of carrying out the pre-production. Version control and its use cases and importance are also explored as well as different techniques used in creating digital graphics effectively.

Finally, the learned theory is put into use as the plans are made for an ongoing horror game project. The plans consider the team size and other hard-set limitations set by the development team. The previous plans and decisions made by the team are incorporated into the planning. The plans will be utilized as the projects' graphics team enters pre-production. The plans made help the development team achieve the graphical style they are aiming for and shines a light into possible pitfalls and problems the project may face. The theory functions as a guide to any graphic artist, team lead or producer who wishes to thoroughly plan and execute pre-production.

## Forward

I'm grateful for all the people that have been on my side during my time at KUAS. Thanks for the good times we had, and for the good times we'll have in the future.

A handwritten signature in black ink, appearing to read 'Riikka Kilpeläinen', with a stylized, flowing script.

Riikka Kilpeläinen

## Table of Contents

1	Introduction.....	1
2	Introduction to game production practices.....	2
2.1	Stages of production.....	2
2.1.1	Pre-production.....	3
2.1.2	Production.....	3
2.1.3	Post-production.....	5
3	The aim of pre-production.....	6
3.1	Limitations.....	7
3.1.1	Hardware.....	7
3.1.2	Team composition.....	7
4	Approaching pre-production.....	9
4.1	Documentation.....	9
4.2	Tools and techniques.....	12
4.2.1	Moodboards.....	12
4.2.2	Thumbnails.....	13
4.2.3	Colour palettes.....	15
4.2.4	Concept art.....	16
4.2.5	Mock-ups.....	17
4.3	Version Control.....	17
4.4	Techniques.....	18
5	Planning the pre-production of a horror game.....	20
5.1	Background information.....	21
5.2	Goals.....	22
5.3	Planning the pre-production.....	24
5.3.1	The overall mood.....	24
5.3.2	Environment assets.....	24
5.3.3	User interface.....	26
5.3.4	The monster.....	28

6	Conclusion.....	30
	List of references.....	31

## List of Symbols

AAA, triple-A	(Game) Games developed by large development studios, usually large in scale. (Company) Large game development companies, 50-100 and more employees.
Agile	Agile software development. A collection of methods that enable iterative and adaptive development. [1]
Asset	Pieces used in building a finished game. 3D models, scripts, textures, audio files are assets.
DLC	Downloadable content
HUD	Heads-up display
Low-poly	A mesh with low polygon count
PS1	The PlayStation
Mod, modder	A modification made to a video game by a fan. A person who makes mods.
PBR	Physically based rendering.
Shader	A piece of program that draws graphics on the computer screen on the graphics processing unit.
UI	User interface
VFX	Visual effects
VR	Virtual Reality. In games, this usually references the headset and controllers that are used to track player's movements and render the game world, or the VR-games themselves. [2]
Waterfall	A project management methodology. Also known as linear-sequential life cycle model. [3]
Whiteboxing	The act of designing a level using primitive shapes (white boxes) in the place of finished level graphics.

## 1 Introduction

Developing a commercial game is a lengthy process with many steps and risks that should be taken into consideration. Proper planning helps to avoid the common pitfalls that many projects face and to provide a peace of mind for the development team. This thesis focuses on one of those steps, the pre-production, from the point of a part of the team, the graphics department.

Each discipline in game development has their methods, common workflows, and tools that they use to achieve their goal. This thesis will study the different ways the graphics department has as their aid to achieve the goals set for the pre-production. How to set those goals and what they may contain as well as the requirements and challenges the team must keep in mind will also be explored.



## 2 Introduction to game production practices

This section is an introduction to game production, to better understand how the graphics team's work differs from the rest of the development team.

Video game development borrows a lot from traditional software development, but merely adopting software development practices does not benefit the team or the product. As the expertise of game development team is usually broader than the expertise of a software development team, different strategies are to be used when managing the project and the people. On top of more traditional developer roles, game development teams often include musicians, writers, and consults on topics that the game may address, like health experts with exergaming titles. [4]

Another industry the game development often borrows from is the film industry. Titles like Director, Art Direction, Stunts, and even Costume Design are nowadays seen in the credits of AAA game titles [5][6]. This means that game development teams may have to adopt some tools and methods used in cultural management.

The combination of practices from software development and the highly creative film industry means that game developers tackle with unique problems. The need to re-shoot scenes with voice actors or to iterate on major gameplay elements can get in the way of more traditional development models, like the waterfall methodology. Therefore, game studios often utilize more flexible (i.e. agile methods) project management tools, like Scrum or Kanban. The mixture of agile methods and waterfall allow the needed structure for larger studios, while the more reactive mixture of agile methods allow smaller studios to take advantage of their smaller size and more flat leadership structure.

### 2.1 Stages of production

Video game production consists of defined phases, which vary between teams, companies, and projects. The order of these phases stays the same, but the definitions of the phases vary and

the lines between the different phases may blur. The phases can be divided into three categories, based on where they appear chronologically in the production timeline. [7]

- Pre-production
- Production
- Post-production

#### 2.1.1 Pre-production

Pre-production is the phase that comes first in the production timeline. The development team may enter pre-production at different times since management is likely working on the pre-production of the next project while the previous project is in live or approaching live. Before the production can start, some plans must be made. What needs to be done, when, by who, and how. These questions should be answered in pre-production, so the team has a clear plan for how the project should be approached. Pre-production is covered in-depth in a later paragraph. [7]

#### 2.1.2 Production

Production is the phase, where actual development happens. Phases often included in production are development (sometimes just called production), beta, and alpha. Like mentioned before, these vary between companies and development teams. Some include beta and alpha as parts of production, some visualize them as separate phases that come after production and before post-production. [8]

In development, the features for the product are made. Depending on the project management style of the project, these features may go through smaller sprint cycles. In figure 1, a traditional scrum sprint cycle is shown.

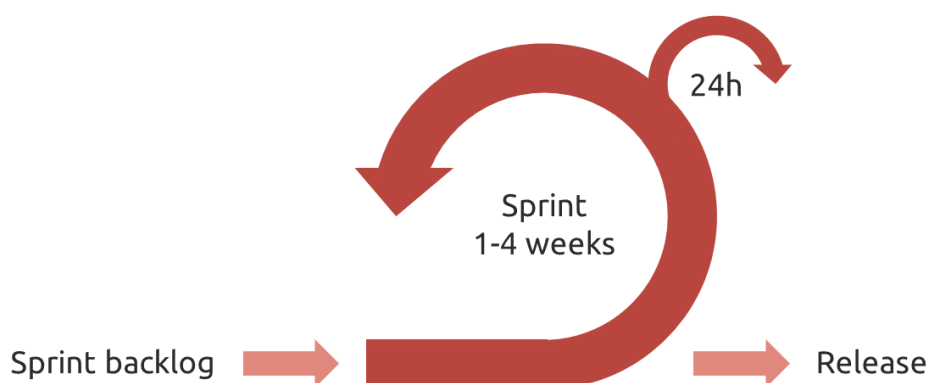


Figure 1. Simplified scrum sprint cycle

During this cycle, which is called a sprint, the feature goes through development (so a gameplay mechanic is programmed, or an art asset is produced) and enters testing or review, where peers give feedback and evaluate the created asset. If the asset is of needed quality, it is marked as done and the creation of a new asset can begin. If the assets need to be iterated on, it goes back to production. Sprints are planned before they start and the features and tasks that should be done during that sprint. This meeting that happens before the development team enters sprint is called sprint planning, or sprint meeting. This structure helps the management follow if features are delivered in time, and how much work is done in a certain timeframe. Not all development teams use sprints, but since they make following the progress of tasks easier, they are often utilized. When a sprint ends, the development team reviews the work done during that sprint and plans the next one. [9]

Beta and alpha usually refer to the late stages of production when testing is concluded. Near or during alpha, the project enters feature lock (also called content lock or “freeze” in software development). The lock means that no more features are to be added to the project and puts an end to the iterative process of development so the product may proceed to launch. In alpha, internal testing of the product is concluded. When the results of the internal testing are sufficient, the beta stage starts. In beta, the product is sent to an external tester, which tests the different aspects of the product. When consumers talk about a video game being in beta, they

mean that the game is not yet finished, but is available to everyone through open beta, or to selected individuals/regions through closed beta. [8]

### 2.1.3 Post-production

Post-production means the phase that comes after production. The style of post-production varies heavily based on the type of game, since many smaller titles may have a very minimal post-production phase. Before the product is ready to be published globally (also called going live, or launching), it can go through soft launch. In a soft launch, the game is published on a certain region, or for certain demographics before going global. This is usually exclusive to mobile games. Using the term “going live” may be confusing, since it is often associated with live games. Live game means a game, that has ongoing support and new content is added regularly. This is not the case with many smaller PC and console titles, that are launched and have no ongoing content creation aside from patches and occasional extra DLC’s. For this reason, the term launch is used in this paper and we will be focusing on games that are not live. [4]

After launch, there are many tasks not directly related to development. Customer support, managing different stores, bundles and sales are some of those tasks. The development team may be involved when new marketing graphics or patches need to be made. Patches are additions to the game, that often include bug fixes and small adjustments based on player feedback (direct, or feedback gotten through in-game analytics). The patch may also include some new content.

DLC refers to extra content made to the game (that is digitally available). DLC’s often include new maps, weapons, or cosmetics. When a new DLC is worked on, it goes through its own development cycle with the phases mentioned before. Unlike patches. DLC are optional content while patches are direct additions to the original game that may have to be downloaded for the game to function as intended. [7]

### 3 The aim of pre-production

The purpose of pre-production is to prepare for production. Figuring out the graphical style, which tools should be used, and defining pipelines are some of the important tasks. Assembling the team and defining clear roles for each person should also be done as early as possible to avoid confusion within the team. If the team needs to learn new skills, be it new workflows or tools, it should be clear before entering production.

The workload often differs vastly between different disciplines in the team as the project progresses. The art team might be ahead of the programmers, which can lead to problems if they are too dependent on the programmers, or the designer. When figuring out the pipelines, this should be taken into consideration. How can everyone work as independently as possible, without being dependent on other people for their tasks to progress? Having the artists comfortable with a broad array of tools helps to distribute the workload. For example, having the artists assemble the assets in-engine themselves instead of the art lead, programmers or the technical artist helps with distributing the workload. The vision for each individual asset stays consistent, with clear, purposeful documentation and peer-reviewing (or having the lead check each asset before they go forward to the rest of the team).

At the very start of pre-production, goals should be set. What things need to be done to proceed to production? These goals vary each project and may change during pre-production. The questions pre-production should answer are “What? How?, When?, Who?, and Why?”. [10]

“What” should be done means figuring out the style of the game, how many assets should be made and what type of assets they are. The Art Bible, moodboards and thumbnails help answer this question. “How” they should be done means tools, and creation and design pipelines. Art Design Document and the Technical Design Document should answer these questions. Scheduling the work will also be done during pre-production, to an extent. Figuring out what is needed for the minimal viable product, and are there assets that other disciplines might need right away? The technical artist or gameplay programmer might need the player character as soon as possible, so it should be prioritized over minor environmental assets. Defining clear roles for everyone answers the “who” question. It helps to avoid confusion and prevents tasks being slipped in the cracks when no one is sure who should be working on it. The roles are not

set in stone and may change later in production as the project progresses, but these changes should be discussed so everyone is on the same page. “Why” should be asked when defining any goal. Why is something done this way? Is it because it has always been done like that and is so, are there new, better ways to achieve that goal? This also goes for asset creation. Why a certain asset should be created, what does it add to the experience? During pre-production, the art team should be closely working with the designers, so that the experience is coherent and that the art amplifies the effect the designers are going for.

### 3.1 Limitations

The art team can rarely create whatever they want with no limitations. The target audience and hardware, the team size and composition limit what can and should be done. Some of these requirements are sometimes decided before the production starts, for example when working on a DLC or a sequel to an existing title. Usually, this is not the case, so all these aspects are decided as the pre-production progresses.

#### 3.1.1 Hardware

Hardware limitations need to be kept in mind from the very start. Some styles are not easily achieved for certain hardware and some are almost impossible to recreate without heavy modifications to the standard workflow for that style. For example, creating HDPR materials for a mobile VR game will not be feasible [11]. The target hardware often tells something about the target audience, as well; customers on PC likely expect different graphical style than customers on mobile devices.

#### 3.1.2 Team composition

The composition and size of the development team play a major part in deciding what kind of art should be made and how much of it can be done. If the graphics team is small, creating large

amounts of realistic characters and environment assets likely will not be possible. In that case, it should be evaluated what corners could be cut to achieve the goal. Buying some assets from third parties, using photo bashing for textures instead of creating everything from scratch and making the assets more versatile and repurposing them are all valid options. The size of the rest of the team should also be considered. If the graphical style calls for modular assets, someone should design those and the ways they interact with each other. Furthermore, if the programming team is small, it is not likely to have programmer-heavy art like intricately rigged, procedurally animated characters or procedurally destructive environment.

Custom tools might be something that profit the team, for example, being able to create colliders inside the 3D modelling software might benefit some teams greatly. These tools need to be developed or searched for and tested and the value they bring to the project should be evaluated. Like all tools and programs, they should only be used if they solve a problem the team is facing or if they make the existing workflow more efficient.

## 4 Approaching pre-production

When the team starts pre-production of a project, the tools at their disposal that help them achieve their goals should be known. As mentioned in the previous chapter, pre-production should have a set goal in mind. Selecting the tools and practices that help achieve those goals is also part of pre-production if there are no existing pipelines to build on.

### 4.1 Documentation

Documents are a tool that the team uses to communicate the vision of the game to each other. Usually, these documents are either created by the team lead or overseen by them so that the vision stays consistent. The format of the document varies between disciplines and the preferences between documents vary between individuals. The game itself also affects the contents of the document; a 2D mobile game might have extensive coverage for UI elements, while atmospheric 3D VR game might not. Some studios prefer wiki-style documentation, where everything is stored on a wiki, while a light art bible may be a PowerPoint presentation. These should be discussed with the people who will be using the document. Documents should be kept up to date if it is feasible, but obsolete documents that no-one is using can and should be abandoned. They should be created as they are needed, because if they are created too soon in advance, they are later no longer likely to represent the product the team is developing. [12]

Art Design Document or Art Bible (ADD and AB) are often the only documents created for and by the art department. When a document is more text-heavy and resembles a traditional document, it is often referred as the Art Design Document and when it is more visual, broad and high level, it's referred to as the Art Bible. AB often deals more with the visual style of the game, while ADD contains more technical information like the tools used, naming conventions, and UI flow charts. The Art Bible may also be referenced as the Style Guide. A great example of an AB, or a Visual Guide, is Dota 2s Character Art Guide created by Valve Corporation in figure 2.



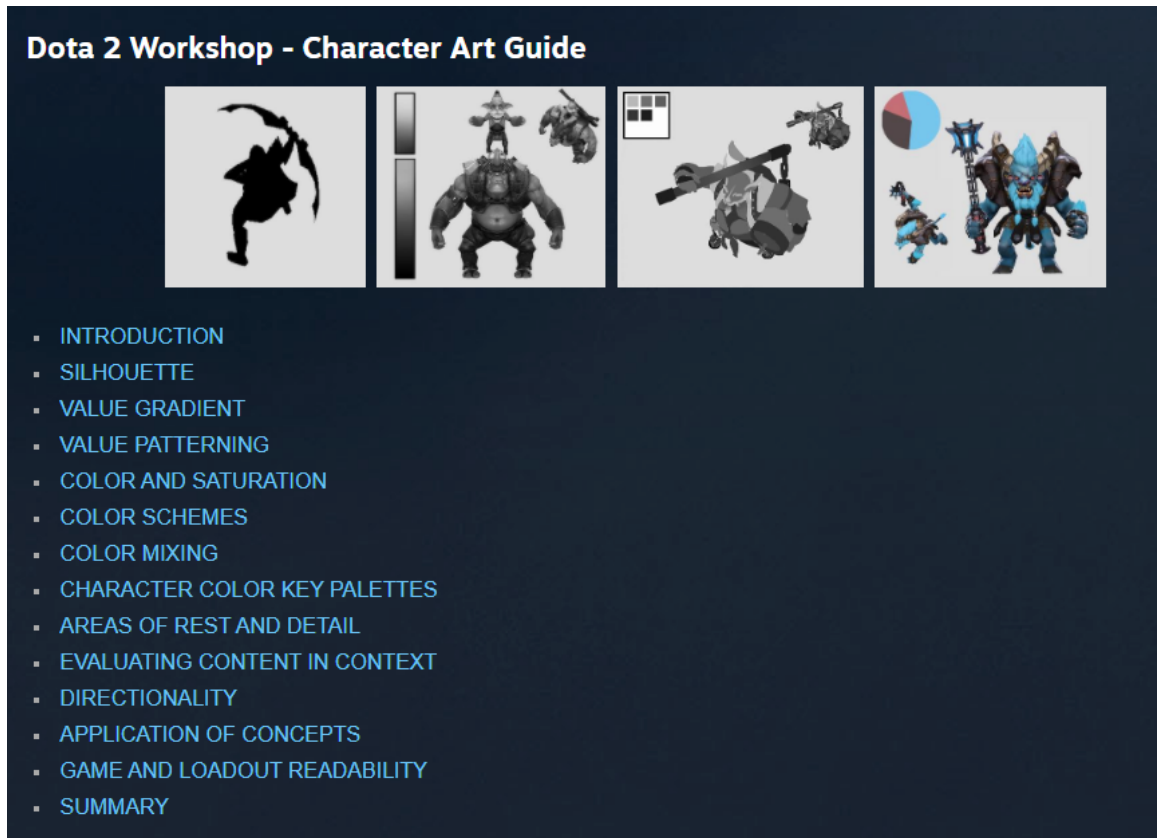


Figure 2. Contents of Dota 2's Character Art Guide

The Visual Guides for Dota 2 can be found on Steam in Dota 2's Workshop page. The site contains individual pages for character art, in-game items, chat emoticons, animations and more and it is kept up to date with the game. The site helps modders create content that is in-line with the game's visual style as Valve often involves the community in content creation. [13]

On top of guiding the visual style of the game, documentation can be used to communicate technical information. What software and add-ons should be used, how large the items should be in relation to each other, what resolution or polycount should be used, et cetera. One highly practical document (or part of a bigger document) is the asset list. The asset list contains all the assets that need to be made for the game. It should be decided if the asset list holds every asset and their variations, or just UI assets or important quest items that need to be made. If the list contains information about the assets' status (is it made, who is making it, when) the asset list can overlap with the development team's Kanban board if they have one. This may or may not be desired, so it should be taken into consideration.

Naming conventions are often overlooked but become important as soon as multiple people are working with one asset. Consistent naming conventions help differentiate between assets fast and easily and leave no room for guessing. Especially with PBR maps, it might be hard to differentiate with different grayscale maps if they are not correctly named. In this naming guide by Michael Allar for Unreal Engine 4 projects, assets are named using prefixes and suffixes.

Asset Type	Prefix	Suffix	Notes
Texture	T_		
Texture (Diffuse/Albedo/Base Color)	T_	_D	
Texture (Normal)	T_	_N	
Texture (Roughness)	T_	_R	
Texture (Alpha/Opacity)	T_	_A	
Texture (Ambient Occlusion)	T_	_O	
Texture (Bump)	T_	_B	
Texture (Emissive)	T_	_E	
Texture (Mask)	T_	_M	
Texture (Specular)	T_	_S	
Texture (Metallic)	T_	_M	
Texture (Packed)	T_	_*	See notes below about <a href="#">packing</a> .
Texture Cube	TC_		
Media Texture	MT_		
Render Target	RT_		
Cube Render Target	RTC_		
Texture Light Profile	TLP		

Figure 3. Naming conventions for textures inside Unreal Engine 4

In figure 3, the naming convention is documented by Michael Allar for textures inside Unreal Engine 4 (UE4) projects. While some of these are UE4 specific, this type of system can be

adopted for any engine or be made engine-agnostic. Like any document, naming conventions should be adapted for your game as a system like this is likely not useful for a 2D game.

Along with naming conventions, the scale of the assets should be decided and communicated somewhere. Scale, in this case, can mean the resolution of textures, polycount, or the “physical” scale of 3D assets. If the player needs to interact with the assets directly, this becomes increasingly important. There may be assets that the player must be able to jump on top of or duck under or behind.

## 4.2 Tools and techniques

This sub-chapter introduces some of the tools that are useful for defining the graphical style of the game. They are listed as separate, but they often bleed into each other in some way or other. For example, one might use existing thumbnails to help refine the colour palette.

### 4.2.1 Moodboards

Moodboards may sound self-explanatory. They are collections of images or other media, that help define the mood the game or a certain part of the game, is aiming for. In the beginning, there may be general moodboards for the whole game, and later there can be ones for different areas in the game, characters, or props.



Figure 4. A moodboard made for *Korpus: Buried Over the Black Soil*

It is considered good practice to include a wide variety of pictures in one's moodboards, not only concept art/screenshots from existing games. Having a wide array of references helps the team to innovate and not just mimic existing titles and ideas. In figure 4, the moodboard was made to capture the mood and overall aesthetics of an old run-down bathroom.

#### 4.2.2 Thumbnails

Thumbnails are small, usually grayscale images that contain very little detail. The purpose of thumbnails is to get the general shape language of the thumbnailed object/environment down, without having to care about smaller details or colour. The amount of values used is also often limited and sometimes a thumbnail is only a silhouette. Thumbnailing is often one of the first processes when designing something visual, and it is generally considered that quantity should be valued over quality. Every thumbnail should still have a purpose and they should explore an idea, but time should not be wasted on refining and polishing thumbnails. A large quantity helps

the team eliminate ideas that are considered worth pursuing while eliminating those that do not need further iterating on. [14]



Figure 5. A small selection of character thumbnails

Once there are enough thumbnails to compare them, they should be evaluated. It should be decided which shapes and values work, and which do not. Based on those, a new set of thumbnails can be done. The thumbnails in figure 5 have been drawn further along in the design process. The character's main traits have been somewhat decided, and these are the later iterations. This time the level of detail can be higher, and the amount of valued used can be increased. This process is repeated as many times as is necessary. Like characters, environments can also be thumbnailed.

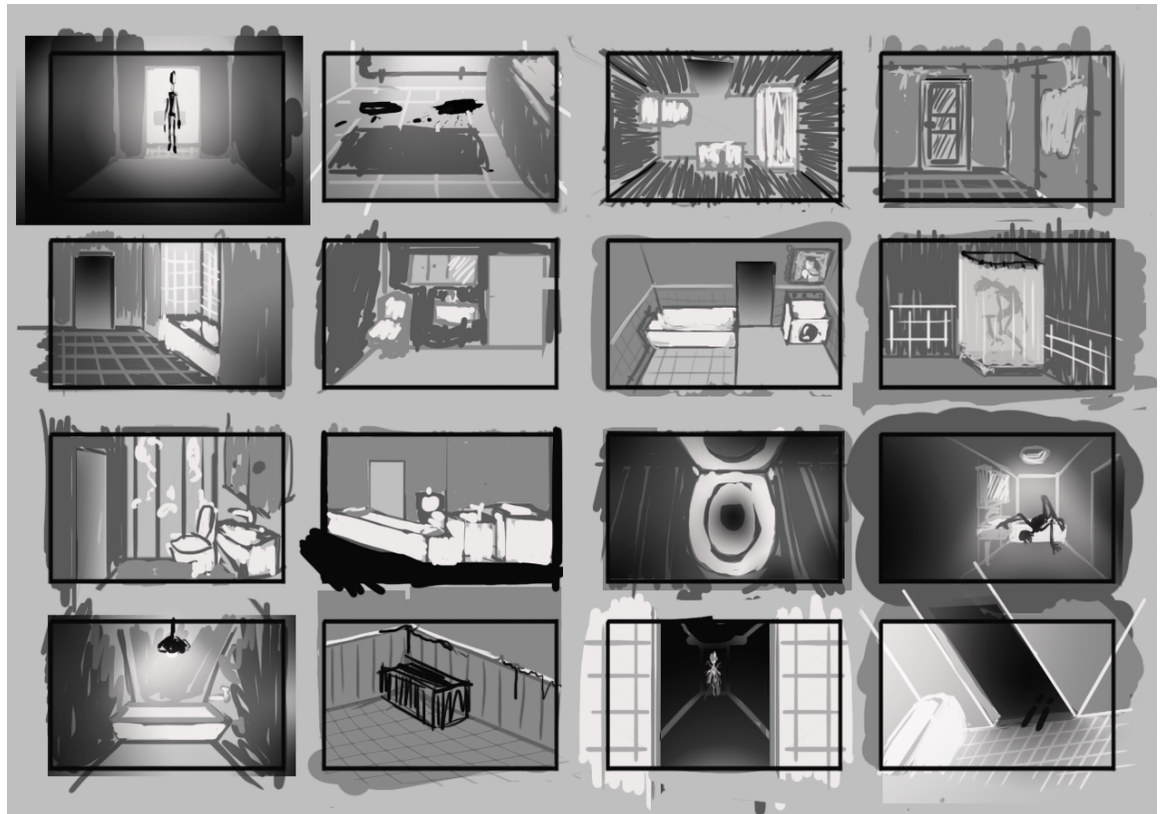


Figure 6. Thumbnails for the interior environment for *Korpus: Buried Over the Black Soil*

In figure 6, the thumbnails are a part of a larger set. The goal of these thumbnails was to explore different ideas for not only the layout and furniture but also the mood. These types of thumbnails may not be as useful in designing the precise layout of a level, but they offer a good starting point and help communicate different ideas to the other members of the team.

#### 4.2.3 Colour palettes

Colour is an important factor, whether the game is realistic or highly stylized. Even when working on a realistic game, the colours used should be considered. Items that are not of importance to the player should not draw the player's attention needlessly since it will cause confusion. If there are objects that the player needs to pay attention to, colour should be used as well as shape, movement, and lighting. Colour theory is a broad topic on its own and important in every kind of visual design, so designing a palette for your game will likely take more time than initially anticipated. [15].



Figure 7. The colour guide used for the environments in thatgamecompany's game *Journey*

The colour guide in figure 7 by Matt Nava shows how the palette of the environment changes as the game progresses. This creates dramatic shifts in mood and sets the player up for the upcoming challenge by the colours and values of the environment only. The cooler and darker environments are more hostile and possess more threats to the player than their lighter, warmer counterparts. [16]

#### 4.2.4 Concept art

Concept art, in this instance, refers to polished illustrations of the game. They are often used to convey a mood that the game aims for. Concept art is often perceived to be the most important piece of pre-production when it comes to art, which is usually not the case. Flashy pictures are great for selling the game and getting the team motivated, but they often do not benefit the art team that much. They are often not accurate enough to be used as a reference for 3D models and take up too much time to be churned out like thumbnails or other sketches.



This does not mean that concept art does not have a place in the pre-production; many studios outsource a few pieces of high-quality concept art right at the start of the production, to use in marketing (for pitching externally and in-house). [10]

#### 4.2.5 Mock-ups

Mock-ups can often be confused with concept art. Mock-ups are often visualizations of certain aspect in a game, like a certain level or the user interface, whereas concept art aims to convey a feeling.

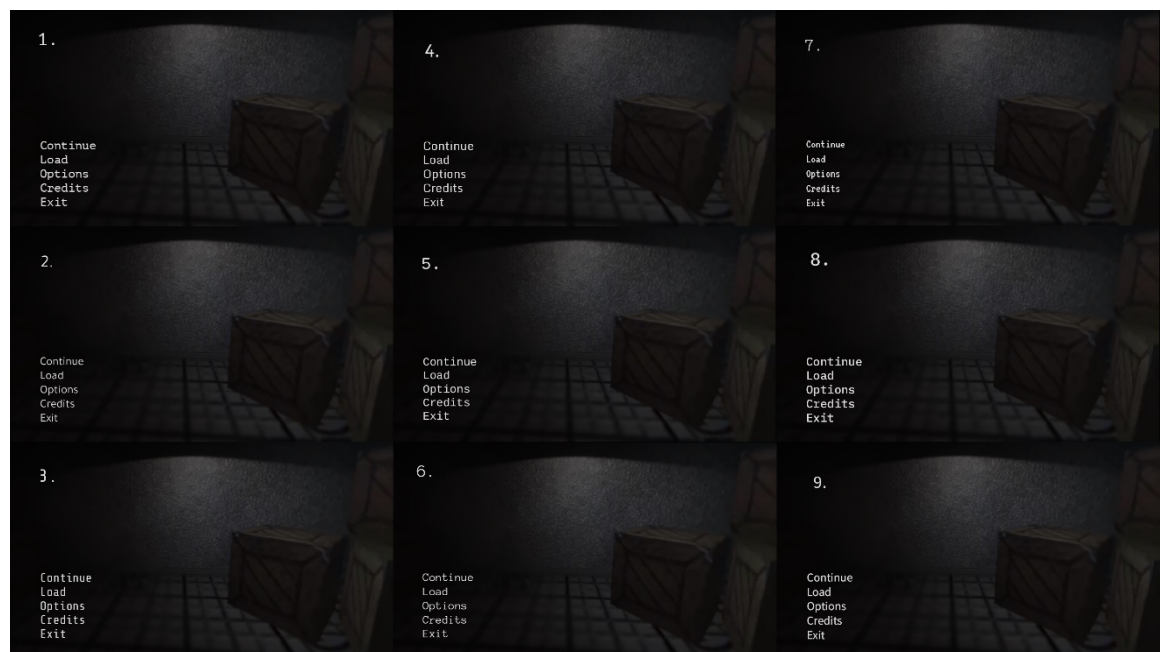


Figure 8. Mock-up of various font variations for the game *Korpus: Buried Over the Black Soil*

They can be used in various ways; to get a better feeling of a level layout or showcase how the game would flow. The mock-ups in figure 8 are done to find out the main font style to be used in the game's UI. While mock-ups usually are images, they can also be in video format. This way animations, the flow of the game and transitions between screens can also be experimented with. For example, a mock-up gameplay trailer could be done [10].



### 4.3 Version Control

Version control is vital with any commercial project, as it holds a record of changes made in files. This allows the users to collaborate on files together and revert to a previous version of the file, if necessary. Version control is a broad topic on its own, so it will be only covered briefly. It is important to select a version control system (VCS) that works for the team and thoroughly coach everyone how it should be used.

There might be a separate version control structure, a repository, for the game itself and the art files. The art repository holds all the working files; like .PSD and .BLEND files. This is so that every artist has access to everyone else's work. This is useful for checking the scale, vertex counts, colours and patterns used, and making sure that the assets being worked on look well together. Having all the files on a centralised system instead of individual computers/accounts is also part of risk management. If one hard drive corrupts, or a person leaves the team abruptly, all that person's work is not lost. The VCS of your choosing should be implemented as early as possible since it also needs testing and there is a learning curve for anyone who is not familiar with the VCS system used.

### 4.4 Techniques

Now that the different methods for figuring out the graphical style of the game have been discussed, some tools and workflows will be introduced. Individual programs will not be discussed as they are not relevant to the topics discussed. All the workflows and such latter mentioned are not program-specific. Efficiency is the key, when in any phase of game production. More so in pre-production, since the plans may change rapidly from what they once were. Creating graphics that serve their purpose at a rapid speed is an essential skill, and the graphics team should use every tool available to them.

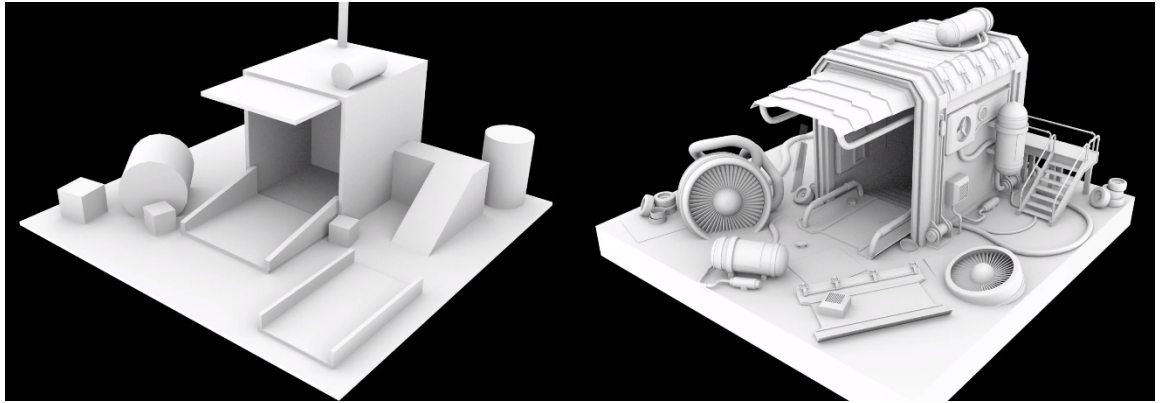


Figure 9. A 3D-scene created to be painted over for concept art

Painting over 3D-graphics (also called painting over) speeds up the process of creating a finished 2D-image. Be it concept art or environment mock-ups, painting over means that one does not have to figure out scale, lighting, or perspective only on a 2D surface. [17] In figure 9 by Sam Kemp, the same scene is seen in different stages of modelling. How far these 3D-models are taken depends on the artist, are they more confident in 3D or 2D? If 3D is their strength, they may build the scene further along and paint only the values, textures, and smaller detail on top of the 3D-models.

Whereas painting over is combining 3D models and 2D painting, photobashing means combining a collection of 2D images to create a new image. This technique, like painting over, greatly reduces the time spent on mock-ups and concept art, especially if one is aiming for a realistic style. In photobashing, multiple images are combined to create a new illustration. The photorealistic detail from the photos means that the artist does not have to spend time painting all the details in an image. [18]

## 5 Planning the pre-production of a horror game

The theory that has been concluded in previous sections can now be put into action with a case, by planning the pre-production of a video game. These specifications latter mentioned come from plans made within the development team and the design and production documents made by the team.

The game is a puzzle action horror game for PC, where the player finds themselves in an empty arcade hall. As the game progresses, the player finds a haunted arcade machine that they can use to manipulate the world that they are in. At the time of writing, the project has been in production for seven weeks, and the workflows have been established for designing, programming and audio. The game has PS1-esque graphics, with low poly 3D-assets with low-resolution textures and pixel graphics for the haunted arcade machine game. The project utilises heavy post-processing effects, most notable being pixelization effect that artificially lowers the resolution of the game.

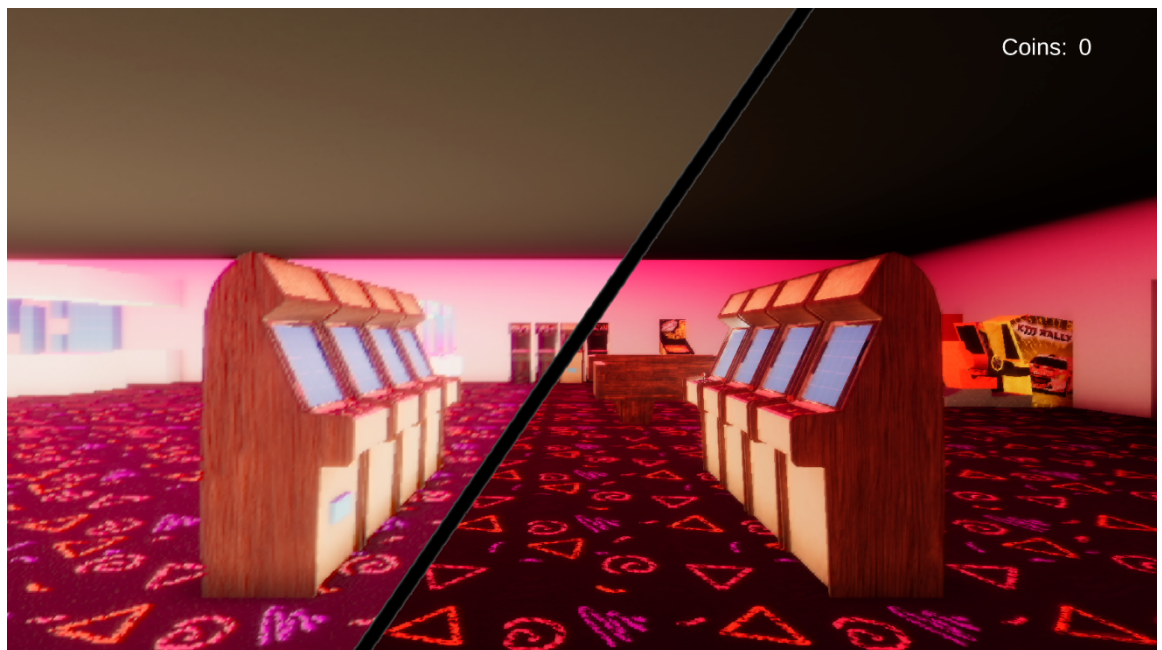


Figure 10. Comparing the look with and without the post-processing effects.

While the aim is to resemble old PS1 era games, the team will also utilize more modern techniques like complex lighting setups and PBR materials. On the left side of figure 10, a

relatively light pixelization effect can be seen in action (along with some other effects ). This way, the game looks polished and it is made clear to the players that the PS1-esque graphics are a deliberate, artistic choice and not just a technical limitation.

The game engine used is Unity 2019.4.9f1, which offers a lot of new post-processing and lighting tools with the high-definition render pipeline (HDRP). Some notable features include volumetric fog, built-in post-processing effect (PP-effects) and ability to create custom PP-effects, screen-space ambient occlusion and screen-space distortion [19]. On top of the built-in PP-effects, the team is using a third-party retro image effect that helps to achieve the PS1-esque look.

### 5.1 Background information

The development team consists of three people: one designer, one audio engineer/gameplay programmer and one graphic artist. The team has worked together before on a published larger-scale game with a similar art style and core gameplay. The previous title was published on Steam and Itch.io and has an active social media channel.

The overall schedule has been decided; the game should go from initial prototyping to content-freeze in 17 weeks (total development time around four and half months) and the core gameplay elements have been implemented as of the time of writing. Two people (the designer and the audio engineer) work on the project full-time as it is also their thesis project, while the artist works on the project as they are able, remotely from the rest of the team. The artist will handle asset implementation in the game engine and any graphics programming necessary.

Based on these requirements and limitations, a SWOT analysis (Strengths, Weaknesses, Opportunities, Threats) can be done.

- Strengths
  - Development team has worked together before in a similar project.
  - All team members are flexible, they are comfortable working outside of their area of expertise.

- Small team size enables faster reaction time in development
- Weaknesses
  - Uneven workload between team members as not all work on the project full-time.
  - Small team size is heavily dependent on everyone; if one were to become unable to work on the project, it would heavily impact development.
  - The artist works remotely from the rest of the team.
- Opportunities
  - Pre-existing published title means that the development team has at least some existing following on social media.
- Threats
  - As the development team is currently finishing up their studies, they may enter employment during the project.
  - The team must rely on some third-party assets due to the strict schedule, which may have inadequate customer support in case of technical problems.

Keeping these observations in mind, the next step is to define the goals for the project.

## 5.2 Goals

Before any plans can be made, it is important to set goals that the plans aim to achieve. With art, these goals can be broken down in multiple ways. In this case, the goals are set for each asset type (environment models, UI, VFX, et cetera).

The overall mood is important to nail down and each asset must work to achieve that mood. The game is set in an old, empty arcade so vibrant colours and flashy lights are used. The lights and post-processing effects are what help with the horror aspect.

As the game is a 3D game, most of the game assets are textured 3D-models. To achieve the modernised PS1-esque graphics, the models need to be low-poly and textures can be low-resolution. The pixelization effect used helps with the retro aesthetic, while also lowering the workload of the artist since the 3D-assets cannot be viewed in detail. The goal is to create as high volume of assets as possible with relatively low effort since the resources are limited. Assets should be made so that they are usable in many different scenarios.

The project being a horror game, the player needs to have something to be afraid of. In this case, the team has opted for a monster that chases the player around in the 3D and 2D worlds. The simple premise of the monster is that it is a glitch in the haunted arcade game's programming. The visuals need to convey this in both the 3D and 2D world in their own way. Likely heavy VFX and custom shaders need to be done for the monster to appear broken. The designer has wished for the monster to appear to distort reality somehow and the monster to be somewhat abstract.

The game also features 2D-gameplay segments, where the player plays on an old arcade machine. The graphics should be heavily simplified so that they look significantly older than any other game in the arcade. This means that the game will likely have a limited colour palette and small texture resolutions for the sprites.

The user interface for the game should also work with the retro aesthetic, so in-game menus should be thoroughly researched from games originally released for PS1. The game contains a heads-up display (HUD), a simple inventory system that is always visible on the screen. There will also be a main menu, options menu and pause menu. The HUD is only UI element that the player needs to interact regularly with during the game sessions, so it should have a higher priority than the other UI elements.

VFX help to achieve the retro look, while lighting brings a more modern twist to the graphics. Lighting setups consisting of multiple light sources and volumetric fog create quickly and effectively a range of moods, that the designer can utilize when trying to depict the atmosphere of a certain scene. The game will feature a mixture of real-time and baked lights to keep the game performant on multiple devices.

The only animations the game will have will be on the 2D world. The animations will be only a couple of frames to help empathize the retro look.

### 5.3 Planning the pre-production

Knowing the limitations and goals set for the project, planning for the pre-production can start. The biggest limitation is the amount of work a single artist can make. But having a single artist has also some unexpected advantages, as it almost eliminates one of the biggest challenges in any project: communication. Documentation for the graphics does not have to be as comprehensive as it would have to be in a bigger team. Since the artist works on the assets by themselves from start to finish, the workflows they utilize can change rapidly as they wish.

#### 5.3.1 The overall mood

Planning for the graphical style will involve the designer as kind of a creative director and they will be involved in the early planning stages heavily. The designer also may want to place some assets themselves in the levels. This means that the assets need to be clearly labelled in case he wishes to do so. As the designer is working on level design and whiteboxing, he likely knows what 3D assets must be made for each level. There should be a way to communicate this efficiently with minimal chance of losing information, so an asset list should be considered. As the graphical style is very similar to the team's previous title, most of the graphical design has to do with the overall mood rather than the visual style of individual assets. Moodboards should be used for this purpose. Depending on the designer's preference, one moodboard can be made for the whole game or multiple ones for the different areas or levels in the game. All the moodboards do not have to be created in pre-production if it is not certain that they will be used. In this case, it is creating one basic moodboard for the whole game and more if needed is likely the best approach to avoid doing unnecessary work.

#### 5.3.2 Environment assets

Creating individual environment assets is quite straightforward, as there is an existing game and workflows that can be referenced. Creating low-poly, low-resolution assets with photobashed textures is a fast process so even if an asset needs some rework, it is not as large of a setback as

it would be on a project with a more complicated art style. It might be worth taking the risk of not designing individual assets in depth before starting the modelling to save time. This is the case with smaller assets, that the player will not pay much attention to. But important assets like objects that need to draw the player's attention, or hero items such as the haunted arcade machine should go through a more refined design process. This process may contain thumbnailing, sketching or moodboards of their own. Such items would likely be the haunted arcade machine, the monster, and items that the player can collect like keys and coins. All the important items that should go through this treatment likely will not be known at the time the art is in pre-production. This design process should be somewhat thought through in case it needs to happen on a tight schedule latter in production.

The game also features some 2D gameplay. The game features a haunted arcade machine which should look and feel older than the rest of the arcade. This is also the game's main gimmick. The game's designer has already thought about the graphical style of the 2D game, and it is already further along in the design process than the 3D environments.





Figure 11. A mock-up made by the designer

The mock-up in figure 11 already answers a lot of questions along with the design documents made by the designer:

- All the elements have one or two colours (the main colour and the colour of the background).
- The individual sprites are low-resolution.
- Limited colour palette. Basic environment elements are in white, and the background is black. Important elements have dedicated colours.
- The style does not strictly follow any style of an existing retro system, so the sprite sizes, available colours and such can be decided by the team.

The design of the 2D world still needs some polishing. Due to the simplistic nature of the style mock-ups are quick to produce and all additional designing can be done with complete mock-ups. The only animations in-game will be the player character's animations in the 2D world and retro games should be researched to figure out what style of animation works best with this style.

### 5.3.3 User interface

Since the UI is the first thing the player encounters in the game (in the form of the main menu), it should set the mood and set the player's expectations for the rest of the game. As this research has been done with *Korpus: Buried Over the Black Soil* some of that research can be repurposed for this project.

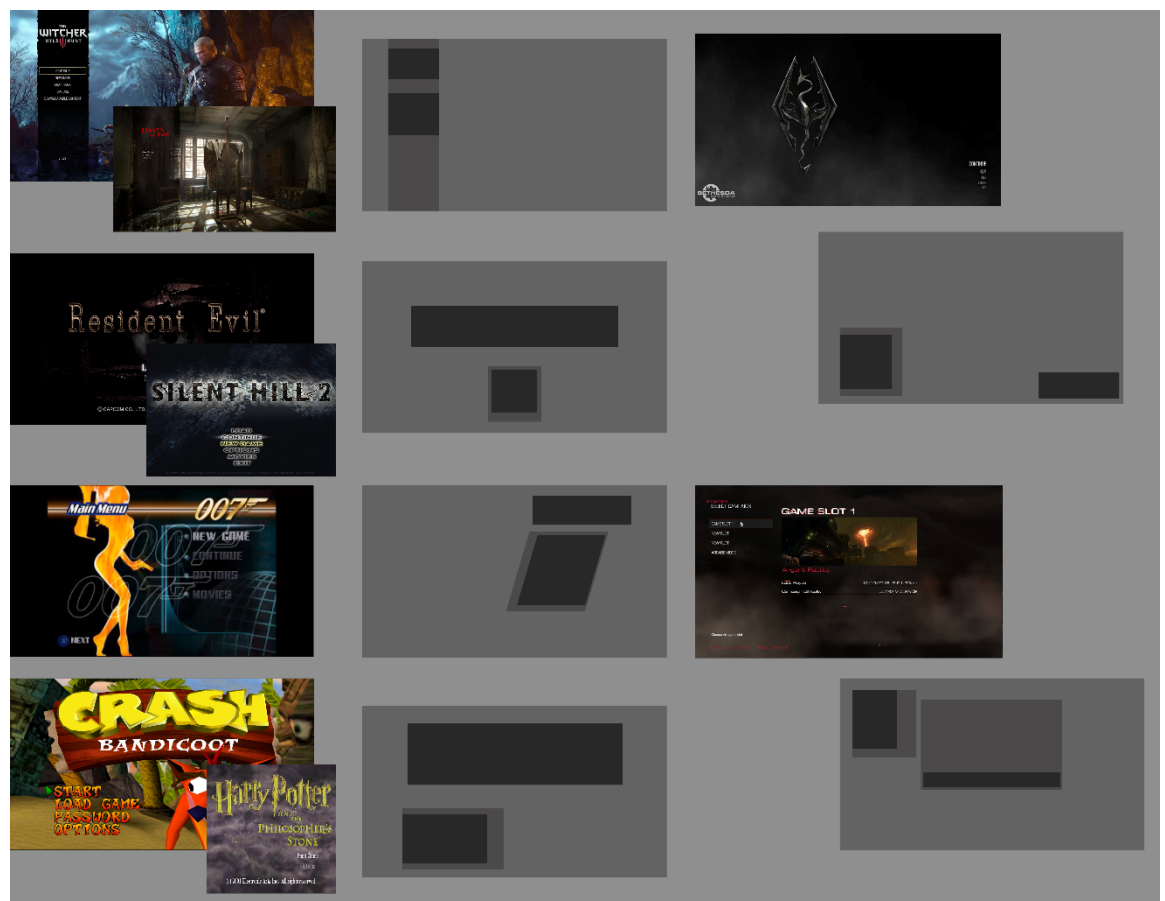


Figure 12. Studies of main menu layouts made for the *Korpus: Buried Over the Black Soil*.

In the studies in figure 12, the main menus from multiple games have been simplified to simple shapes based on the elements in the UI. These can be used as the base when discussing with the designer, what the basic layout for the menus should be. There are also some existing design and ideas for the UI; the menus would follow the style of the 2D game, and the main menu would be viewed through the haunted arcade machine.



Figure 13. In-engine mock-up of the main menu layout.

The mock-up in figure 13 shows the basic idea for the main menu. The menu is drawn on the haunted arcade machine that is seen in the 3D world. The 3D background can be the main game scene so no additional 3D assets would have to be made.

#### 5.3.4 The monster

As the monster is supposed to look like a glitch and distort the reality around itself, it presents one of the greatest challenges in the project. Designing such a creature is difficult with the traditional methods such as thumbnailing, as the monster's visuals depend on the context it is set in. In-engine mockups can and should be created as early as possible. Custom VFX in the form of shaders and particle systems need to be made to achieve the desired effect.



Figure 14. First in-engine test for the monster

The test in figure 14 shows a highly deformed mesh with a custom distortion shader that reflects the environment. The mesh is animated and snaps between different positions while the particles stay still as they re- and despawn. This works as a starting point as this effect is refined and more particles and distortion layers are put on top of the monster.

## 6 Conclusion

The different methods game studios use to approach pre-production were mostly studied by talks given by the art team leads or directors. These talks were not always strictly about the pre-production of art. Whenever the different pipelines and choices made were brought up, questions about the pre-production were also answered. What choices were made, why, and how well they achieved the goals they set were also discussed. On top of these talks and interviews, the author's university studies heavily influenced the way this thesis was approached. The author has studied both production and graphics courses and worked as the producer, art lead and a programmer in the projects she has been a part of. This has widened the perspective, which meant that parts of those three disciplines were covered, or mentioned, in this thesis. Hence the first paragraphs go over the basics of the game production cycle, briefly cover scrum cycles, and why there is a mention about the importance of version control.

Planning the pre-production for an on-going project provided an interesting challenge. Like any development team and game project, this case was unique with its own challenges and quirks. Tasks had to be heavily prioritized and relying on research done during the team's previous project was essential. As enjoyable as creating the plans was, a more conventional game project would have worked as a better example. Many questions were already answered either by the previous project or by plans made by the team's designer. Some aspects of the visual style were not covered in the Planning the pre-production -section, such as the VFX as it had been mentioned in the Background information and Goals -sections. The pre-production has begun, and the plans made during this thesis are considered as the production progresses.

Many things could be improved in this thesis; the scope could be narrowed down for a more focused result. The focus could be more on the topics covering graphics, different tools and methods and more practical information. More could have been said about whiteboxing, planning with the performance of the game in mind and about placeholder assets. How to set-up version control specifically for artists, or how to create useful and actionable documents, for example. Overall, I am happy with the result and I think this thesis functions as a good starting point for someone who is about to step into pre-production as an art lead for the first time.

## List of references

- 1 Beck, K. Beedle M, van Bennekum A, Cockburn A, Cunningham W, Fowler M, Grenning J, Nighsmith J, Hunt A, Jeffried R, Kern J, Marick B, C. Martin R, Mellor S, Schwaber K, Sutherland J, Thomas D. The Agile Manifesto. 13.02.2001. [Cited: 29.10.2020.] <https://agilemanifesto.org/>.
- 2 Bardi J. What is Virtual Reality? [Definition and Examples]. Marxent 3D Commerce. 21.09.2020. [Cited: 14.08.2020] Available at: <https://www.marxentlabs.com/what-is-virtual-reality/>
- 3 Lucidchart Content Team. What the Waterfall Project Management Methodology Can (and Can't) Do for You. 2017. [Cited: 20.09.2020] <https://www.lucidchart.com/blog/waterfall-project-management-methodology>
- 4 Pascarella L, Di Penta M, Palomba F, Bacchelli A. How Is Video Game Development Different from Software Development in Open Source? IEEE, Gothenburg 2018. [Cited 20.09.2020] Available at: <https://fpalomba.github.io/pdf/Conferencs/C28.pdf>
- 5 IMDb. The Last of Us II, Full Cast & Crew. 2020. [Cited: 14.08.2020] Available at: [https://www.imdb.com/title/tt6298000/fullcredits/?ref\\_=tt\\_ov\\_st\\_sm](https://www.imdb.com/title/tt6298000/fullcredits/?ref_=tt_ov_st_sm)
- 6 Toivola T. Artisan of the gaming world. Aalto University Magazine, issue 23. 10.2018. [Cited: 14.08.2020] Available at: <https://www.aalto.fi/en/news/artisan-of-the-gaming-world>
- 7 Stefyn N. How Video Games Are Made: The Game Development Process. CG Spectrum. 23.10.2019. [Cited: 29.10.2020] Available at: <https://www.cgspectrum.com/blog/game-development-process>
- 8 Widayani Y. Game development life cycle guidelines. Bandung Institute of Technology. 09.2018. DOI: 10.1109 [Cited: 03.09.2020] Available at: <https://jiharu.github.io/lowpoly/img/06761558GameDev.pdf>

- 9        Sergeev A. What is Scrum Lifecycle? Hygger. 10.06.2020. [Cited: 03.09.2020] Available at: <https://hygger.io/blog/what-is-scrum-lifecycle/>
  
- 10       Foertsch G. The Art of Pre-Production. GDC talks 2017. 23.08.2017. [Cited: 24.09.2020] Available at: [https://www.youtube.com/watch?v=s2u4jhpZkTQ&ab\\_channel=GDC](https://www.youtube.com/watch?v=s2u4jhpZkTQ&ab_channel=GDC)
  
- 11       Lang B, Rizotto L. Dev Shader Crash Course on VR Game Optimization for Oculus Quest & Mobile Hardware. Road to VR, Youtube. 11.04.2019. [Cited: 29.10.2020] Available at: <https://www.roadtovr.com/vr-game-optimization-for-oculus-quest-mobile-hardware/>
  
- 12       England L. Make Actionable Documentation. Rules of the Game: Five Techniques from Quite Inventive Designers. GDC talks 2016. 39:44-. 30.07.2017. [Cited: 29.09.2020] Available at: [https://www.youtube.com/watch?v=d8QAVGeEj-U&ab\\_channel=GDC](https://www.youtube.com/watch?v=d8QAVGeEj-U&ab_channel=GDC)
  
- 13       Valve Corporation. Dota 2 – Workshop. 2017. [Cited: 04.11.2020] Available at: [https://support.steampowered.com/kb\\_cat.php?s=fb70b4e46eb4fcbdac693a42fd0efc76&id=109](https://support.steampowered.com/kb_cat.php?s=fb70b4e46eb4fcbdac693a42fd0efc76&id=109)
  
- 14       Introduction to Thumbnailing And Quick Sketching. Concept Art Empire. [Cited: 06.10.2020] Available at: <https://conceptartempire.com/intro-to-thumbnailed-sketching/>
  
- 15       Tulleken H, Bailey J. Color in games: An in-depth look at one of game design's most useful tools. Gamasutra. 29.07.2015. [Cited: 06.10.2020] Available at: [https://www.gamasutra.com/blogs/HermanTulleken/20150729/249761/Color\\_in\\_games\\_An\\_indepth\\_look\\_at\\_one\\_of\\_game\\_designs\\_most\\_useful\\_tools.php](https://www.gamasutra.com/blogs/HermanTulleken/20150729/249761/Color_in_games_An_indepth_look_at_one_of_game_designs_most_useful_tools.php)
  
- 16       Nava M. The Art of Journey. GDC talks 2013. 18.10.2015. [Cited: 29.09.2020] Available at: <https://www.youtube.com/watch?v=RoHrwAacTwo>
  
- 17       Neve D, Tokarev K. Mixing 2D and 3D in Concept Art. 80lv. 17.06.2019. [Cited: 29.09.2020] Available at: <https://80.lv/articles/mixing-2d-and-3d-in-concept-art-007gai>
  
- 18       Jacobs TJ. What is photobashing? The art & the controversy. Improveyourdrawings.com, Toni Justamante Jacobs Art Blog. 10.2018. [Cited: 06.10.2020] Available at: <https://improveyourdrawings.com/2018/10/what-is-photobashing-the-art-the-controversy/>

- 19      Unity Technologies. High Definition Render Pipeline overview. 2020. [Cited: 26.10.2020]  
Available    at:    <https://docs.unity3d.com/Packages/com.unity.render-pipelines.high-definition@6.9/manual/HDRP-Features.html>



- Figure 1. Simplified scrum sprint cycle. Kilpeläinen R.
- Figure 2. Contents of Dota 2's Character Art Guide . Valve Corporation. Screenshot: Kilpeläinen R.
- Figure 3. Naming conventions for textures inside Unreal Engine 4. Allar M. Screenshot: Kilpeläinen R.
- Figure 4. A moodboard made for Korpus: Buried Over the Black Soil. Kilpeläinen R.
- Figure 5. A small selection of character thumbnails. Kilpeläinen R.
- Figure 6. Thumbnails for the interior environment for Korpus: Buried Over the Black Soil. Kilpeläinen R.
- Figure 7. The colour guide used for the environments in thatgamecompany's game Journey. Nava M. Screenshot: Kilpeläinen R. Permission from author.
- Figure 8. Mock-ups of various font variations for the game Korpus: Buried Over the Black Soil. Kilpeläinen R.
- Figure 9. A 3D-scene created to be painted over for concept art. Sam Kemp. Screenshot: Kilpeläinen R. Permission from author.
- Figure 10. Comparing the look with and without the post-processing effects. Kilpeläinen R.
- Figure 11. A mock-up made by the designer. Massa E. Permission from author.
- Figure 12. Studies of main menu layouts made for the Korpus: Buried Over the Black Soil. Kilpeläinen R.
- Figure 13. In-engine mock-up of the main menu layout. Kilpeläinen R.
- Figure 14. First in-engine tests for the monster. Kilpeläinen R.