Product Vision

Team Free Pizza

Name	*NetID*	*Studentnumber*
Tim Yue	tyue	4371925
Bart van Oort	bvanoort	4343255
Steven Meijer	stevenmeijer	4368061
Thomas Kolenbrander	tjkolenbrander	4353137
Etta Tabe Takang Kajikaw	eettatabe	4373758

TABLE OF CONTENTS

1. INTRODUCTION

All around the world, everyday in life, there are moments people may want to remember until the end of time. Whether it is something deeply personal, like a holiday with loved ones, or something that brought entertainment and happiness to many, people will want to remember it, ideally, they want a way to make that memory tangible. And people have found ways to do so, in the form of paintings, pictures and video recordings, the latter being the most capable of conveying the atmosphere present at the time of action. Yet, in the case of a large venue, where multiple different cameras have to be used, it gets very difficult for human operators to manually coordinate and control the cameras [1]. Directors may compose a script for the cameras to follow, but these still have to be operated by humans, meaning mistakes are likely to occur. This makes recording the venue in question fairly expensive.

Our product changes that. We aim to innovate the workflow of recording a concert. We want to give directors and cameramen the opportunity to focus on the artistic side of recording a concert, rather than just the technical side. Our product automates the preloading of presets specified in a digital script, allowing the cameramen to focus on creating beautiful, animated presets, capturing the most beautiful moments of the event and correctly timing transitions, instead of operating a switchboard to switch between static presets, in some cases leading to beautiful shots not being taken.

This document is divided into several sections. Section 2 discusses our target group and what customer needs the product will address. Section 3 contains a MoSCoW [2] model of the features we aim to implement.

2. THE TARGET

Our product is mainly aimed at companies who want to record concerts with one or multiple automated cameras. More specifically, our product is aimed at the cameramen and directors who work at these companies.

One problem they currently encounter is that they spend too much of their time on switching between different saved presets on their cameras, rather than actually focussing on artistically good looking camera shots. Another problem they encounter is that every panning or zooming operation has to be done by hand, which can cost the camera operator valuable time. While some panning or zooming operations require an advanced understanding of the scene and artistic insight, a lot of these operations are relatively simple. Therefore, we want to give the cameramen and directors a tool with which they can easily manage their camera presets.

This tool allows them to create a digital script for where different cameras have to be at different times. It does this by allowing the user to create a timeline of operation blocks per camera. An operation block is defined with an initial state (orientation, zoom, aperture etc.) and a time to be active. The user can add effects like ‘Zoom’ or ‘Pan’ to create automatically generated camera movements, with the ability to fine-tune the settings of these effects. The user can also use the timelines to control which camera is ‘actual’ (being broadcasted) at which moment and should have ability to adjust the timeline on the fly. During a

performance, the users can focus on choosing the right moment to switch to the next camera and fine-tuning upcoming presets.

But switching to a next camera can bring problems of its own. When the operator decides to make the next camera ‘actual’, then that camera should not be transitioning between presets. If it is, then the application should prevent the operator from making the next camera ‘actual’. The other way around, when a camera is ‘actual’ it should not be allowed to transition to another preset.

It is important to realise that the application is supposed to be an extension of the user, in the sense that in all cases the user controls the program, instead of vice-versa, so there should be an override feature: users should be able to manually take control over any camera and its respective settings. This should preferably integrate with the old joystick controls, but controlling a camera should ideally also be possible through the application. This override feature allows our product to reduce the workload on the camera team and bring in more versatility with regards to the controls of these different cameras, while maintaining manual control in case anything goes wrong.

As mentioned in the previous paragraph, the application is supposed to be an extension of the user. This also means that the user interface should be easy and intuitive to use, because new users cannot spend hours of time learning how to use a program. We know from the analysis of our target group, that they have experience with video editing software like Adobe After Effects and Premiere Pro, so we want to incorporate ideas of their interfaces into ours, to create an interface in which certain items are in the same general area as the users are used to, thus creating an intuitive user experience.

3. FEATURES IN MOSCOW

This section lists the features to be implemented, according to the MoSCoW method. Thus, this section is divided into four subsections, corresponding to the four categories of MoSCoW prioritisation.

3.1. Must-haves

The features in this list are required for correct functionality of the application.

- Create presets for different cameras.
- Create a digital script with these presets using a graphical interface.
- Saving and loading of digital scripts.
- It should be possible to go to the next shot with just one click of a button.

3.2. Should-haves

The features in this list are not required for correct functionality of the application, but are highly preferable.

- Automatic loading of presets for the cameras.
- It should be possible to make a preset dynamic, making it possible to program camera movements like zooms and pans.
- When the camera is actual it should not be allowed to switch presets.

- When a camera is switching to a preset it should not be allowed to become actual.
- It should be possible to configure the track of a camera beforehand.
- During live production it should be possible to override the script and control everything manually.

3.3. Could-haves

The features in this list will only be implemented when there is enough time to do so.

- Dynamic presets could be set to use not just a linear curve, but also a quadratic, logarithmic or custom curve for the movements.
- A director can view a preview of the whole concert when all cameras are configured.

3.4. Would-haves

The features in this list will most likely not be implemented in this timeframe, but are interesting to implement at a later point in time.

- Cameramen get live notifications about abrupt changes in the script during live production from both directors and other fellow cameramen.
- If an actual camera gets stuck or does not work for some reason, then a camera not in use can automatically take over its task.
- Communication between cameras, making the work of the controllers less tedious.

4. BIBLIOGRAPHY

[1]: Natarajan, P., Atrey, P., & Kankanhalli, M. (2015). Multi-Camera Coordination and Control in Surveillance Systems. *ACM Trans. Multimedia Comput. Commun. Appl.*, *11*(4), 1. http://dx.doi.org/10.1145/2710128

[2]: DSDM Consortium. (2015). *MoSCoW Prioritisation*. Retrieved 29 April, 2016, from https://www.dsdm.org/content/moscow-prioritisation