Cine-AI: Generating Video Game Cutscenes in the Style of Human Directors

INAN EVIN, Aalto University, Finland, Finland
PERTTU HÄMÄLÄINEN, Aalto University, Finland, Finland
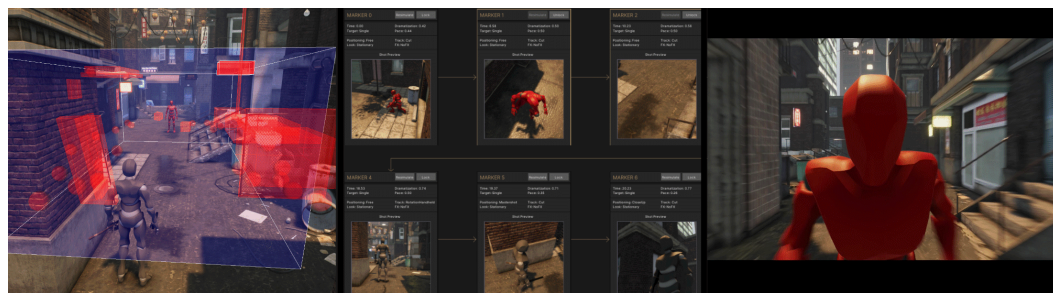CHRISTIAN GUCKELSBERGER, Aalto University, Finland, Finland

Fig. 1. Cine-AI allows users to procedurally generate camera behaviour for in-game cutscenes by mimicking a target director's style. Left to right: cutscene edited by the user at design-time in the Unity game engine; Cine-AI storyboard interface allowing the user to fine-tune the generated scene composition; a still from the runtime cutscene in the style of the director Guy Ritchie.

Cutscenes form an integral part of many video games, but their creation is costly, time-consuming, and requires skills that many game developers lack. While AI has been leveraged to semi-automate cutscene production, the results typically lack the internal consistency and uniformity in style that is characteristic of professional human directors. We overcome this shortcoming with Cine-AI, an open-source procedural cinematography toolset capable of generating in-game cutscenes in the style of eminent human directors. Implemented in the popular game engine Unity, Cine-AI features a novel timeline and storyboard interface for design-time manipulation, combined with runtime cinematography automation. Via two user studies, each employing quantitative and qualitative measures, we demonstrate that Cine-AI generates cutscenes that people correctly associate with a target director, while providing above-average usability. Our director imitation dataset is publicly available, and can be extended by users and film enthusiasts.

CCS Concepts: • **Human-centered computing** → **Human computer interaction (HCI)**; *Interactive systems and tools*; *User studies*; *Empirical studies in HCI*; *Visualization techniques*.

Additional Key Words and Phrases: procedural cinematography, video games, cutscene, imitation, storyboard

Authors' addresses: Inan Evin, Aalto University, Finland, P.O. Box 11000 (Otakaari 1B) FI-00076 AALTO, Espoo, Uusima, Finland, FI-00076, inanevin@gmail.com; Perttu Hämäläinen, Aalto University, Finland, P.O. Box 11000 (Otakaari 1B) FI-00076 AALTO, Espoo, Uusima, Finland, FI-00076, perttu.hamalainen@aalto.fi; Christian Guckelsberger, Aalto University, Finland, P.O. Box 11000 (Otakaari 1B) FI-00076 AALTO, Espoo, Uusima, Finland, FI-00076, christian.guckelsberger@aalto.fi.

1 INTRODUCTION

In-game cutscenes are non-interactive sequences in a video game that pause and break up gameplay. They are often used to progress the story, e.g. by showing critical events or conversations between the characters. Especially in high-quality productions, in-game cutscenes feature elaborate character animations, complex scene composition and extensive cinematography. Conveying a targeted experience requires deep engagement in digital cinematography and directing techniques [1] to arrange the placement of potentially several virtual cameras, along with their motion and behaviours. Game companies thus may need to hire dedicated directors, cinematographers, and entire movie productions teams.

Alternatively, it is possible to leverage AI in co-creative tools [23] to semi-automate the process of directing and camera management. This holds the promise of reducing the costs and repetitive labour associated with traditional cutscene production, and to empower game designers with the creative agency to realise their cutscene ideas themselves. Research has made considerable progress towards overcoming some of the major challenges involved, e.g. automated camera placement, subject visibility, shot continuity and scene composition [9, 10, 19, 21]. However, existing systems fail to produce compositions which feel internally consistent and uniform in style - features that can be particularly well observed in film cutscenes produced by eminent human directors.

To address this shortcoming, we have developed and evaluated Cine-AI, a semi-automated cinematography toolset capable of generating in-game cutscenes in the style of a chosen, eminent human director, and according to user specifications. For our proof-of-concept, we analysed 160 movie clips of two directors with respect to different cinematography techniques, resulting in a style description dataset to be leveraged in procedural cutscene generation. Cine-AI affords user control through design-time interaction with an animation timeline and a storyboard: Based on the user's selection of significant content on an animation timeline, the system visualises the resulting scene composition as an interactive storyboard, which the user can then tweak further. The camera placement, transitions and cinematography techniques in the generated result closely resemble a chosen director's style, providing uniformity and overcoming the stylistic shortcomings of previous work. We make the following contributions:

- A technique to co-create in-game cutscenes that mimic a target director's style in interaction with a user (Sec. 4).
- Cine-AI, a novel, open-source cinematography toolset which realises this technique. To maximise accessibility and support immediate production use, we implemented the system in the popular game engine Unity (Sec. 4).
- A publicly available, extendable style description dataset for use in Cine-AI or related projects. At present, it encodes the style of two eminent directors based on their usage of cinematography techniques (Sec. 3).
- A user study that supports the system's ability to generate cutscenes in a target director's style (Sec. 5).
- A usability study that suggests no major flaws (System Usability Scale grade B) and identifies topics for future improvements and research (Sec. 6).

Our implementation and dataset are available under the MIT license on GitHub (https://github.com/inanevin/Cine-AI), allowing for the straight-forward evaluation, extension and application by researchers and end users. Study materials and results are provided as Supplementary Material. A Supplementary Video presents an overview of the system, its design-time user interaction, and its director imitation functionality.

2 BACKGROUND AND RELATED WORK

In each of the following subsections, we introduce one challenge of automatic cutscene generation, how it has been addressed in previous work, and how it is overcome by Cine-AI. We moreover consider a small body of work on generating storyboards, a central means of interaction with Cine-AI. We only take into account existing publications on storyboards for design-time manipulation, but omit work dealing with e.g. the creation of storyboards for Web video in-sights [15] or the use of storyboards in live-action movie production [18], as it is out of scope. We also exclude work on physical cinematography (e.g. [5, 44]), since the proposed solution space for e.g. subject visibility, camera orientation and occlusion differs too much from Cine-AI's 3D game engine environment.

2.1 Camera Placement

Arijon [2] introduces idioms for appropriate camera positioning during dialogue sequences. He holds that, in a single character environment, the camera should be placed within the field-of-view cone of the subject. For multiple characters however, he suggests to create a line of action using the middle points of the characters' positions. He et al. [19] exemplify a set of rules of thumb and constraints implemented in their *Virtual Cinematographer* that tend to submit to broader approaches and idioms defined by filmmakers throughout the years.

Cine-AI adopts Arijon's [2] idioms and He et al.'s [19] rules of thumb for camera placement. It moreover uses common cinematography approaches to determine camera angles.

2.2 Subject Visibility

The challenge of camera placement is closely intertwined with that of subject visibility, as virtual objects, characters and particle effects in a cutscene can block a clear shot of the the target subject. Lino et al. [31] propose to identify subject visibility areas for camera placement through *visibility volumes*. This technique involves projecting the 3D environment onto a 2D image, which is then divided into smaller cells. By iterating over the pixel information on these cells, one can swiftly determine key areas of subject visibility. Meanwhile Oskam et al. [33] propose an algorithm for visibility-aware path-planning in a virtual environment. It uses visibility data for various parts of the scene along with pre-computed representations of collision-free paths to execute a camera transition with clear subject focus during runtime. Rucks and Katzakis' [36] CamerAI uses reinforcement learning to optimise subject visibility. More specifically, they use a continuous flow of runtime collision information along with a custom reward function to train a neural network camera control policy through Proximal Policy Optimization. The resulting model is capable of successfully tracking target objects in unseen 3D environments similar to those in their training maps. Burg et al. [7] introduce a fully dynamic occlusion avoidance system to maximise target visibility. Drawing on previous contributions [29, 30] on Toric coordinate systems, they project occluder information into Toric Space to create an occlusion map. This is then used to derive an anticipation map by looking up the velocity information of the occluder vertices. This anticipation map is fed into a system that calculates appropriate camera orientation in runtime.

Both Rucks and Katzakis' [36] as well as Burg et al.'s [7] approaches provide accurate subject visibility in runtime. Crucially though, Cine-AI is designed for use in game engines, allowing us to address subject visibility at design-time in order to free resources at game runtime. Consequently, Cine-AI adopts a similar approach to Lino et al. [31], who analyse the current scene from the camera's point of view and perform raycasting to calculate *visibility volumes*. We employ a similar analysis and raycasting within *scene proxies* – user-defined areas in the scene – to calculate potential collisions at design-time and ensure collision-free camera motion during runtime. These proxies are also inspired by Oskam et al. [33] and will be elaborated on in our system description (Sec. 4).

2.3 Shot Continuity

Another challenge in a virtual environment is to create meaningful transitions, i.e. camera cuts and jumps that do not confuse the viewer but support the story, and achieve shot continuity. Cine-AI adopts the method by Jhala and Young [21], who abstract the animation timeline into a network of states. Pre-conditions and goals facilitate transitions between these states, determining whether a state can be entered or left, respectively. This allows treating the scene as a *state machine*, generating shot sequences and motion plans to choose the best result according to the current state. They propose a number of parameters including shot significance to rank and select the best generated sequences. Here, *link planning systems* such as *Longbow*, introduced by Young and Moore [45], can be leveraged for sequence generation.

Similar to Jhala and Young's [21] use of user-parameters and shot ranking, Cine-AI provides various parameters that users can tweak to determine a shot's importance, pace and action value for ranking. In contrast to most other techniques though, the best possible shot is not auto-selected by the system at runtime, but it is offered to the user at design-time through storyboards. This realises complete scene design flexibility. In addition, Cine-AI ensures scene and shot connectivity by relying on the cinematography rules derived from best cinematography practices; to be considered eligible, each generated shot undergoes a series of checks against these rules and is compared with the previous shots.

2.4 Cinematography

An umbrella term, cinematography denotes the art and technology of motion-picture photography [42]. We aim to realise *procedural cinematography*, focusing on how camera placement, shot continuity and composition can be brought together by algorithmic means. Tackling part of this challenge, Christianson et al. [9] propose a *declarative camera control language* to formalise a selection of common cinematography idioms for automation. The formalisation of such idioms makes it easier to categorise rules of a scene along with the goals of particular shots. A similar representation is used by Jhala and Young [21] to declaratively represent storytelling plans. Along with selecting the best shot sequences, it becomes possible to create uniformity across transitions that meaningfully convey the narrative elements. Similarly, Karp and Feiner [24] as well as Drucker and Zeltzer [12] propose encoding idioms and rules in *film grammars* to generate a set of shot sequences via top-down analysis. Such sequences allow for the calculation of motion planning to achieve certain visual goals. Since this technique relies on animation timing information, it is suitable for realising static cutscenes, but does not allow for cutscenes to unfold dynamically based on the previous game state. Jiang et al. [22] propose an approach for translating cinematography information from video clips to runtime camera motion. It consists of extracting cinematography features from sample video clips, deriving camera behaviour from these features with a deep learning algorithm, and applying this behaviour to a 3D camera motion controller. While both projects are in the realm of automated cinematography, Cine-AI focuses on the complementary tasks of defining directorial styles and providing users with a rich interface to manipulate the generated cinematic shot composition.

Instead of solely focusing on film idioms, Cine-AI creates meaningful scene compositions based on real director data, derived from 160 different movie clips. We build on existing work by representing the extracted director style data as hierarchy of idioms, enabling our algorithm to determine the best cinematography technique based on the user's choice of director and scene parameters. Cine-AI allows for cutscenes to unfold dynamically based on the previous game state, rendering the production of different, static cutscenes for each possible gameplay outcome obsolete.

A remaining challenge consists in selecting the best possible cinematography techniques and shot sequences for a particular time during a specific in-game cutscene. Soares de Lima et al. [41] have addressed this with supervised machine learning (ML) methods. They use *support vector machines* (SVMs) to determine the best possible shot selection for a specific cinematography technique, based on the scene type, number of actors and their features. Such an approach can produce great results, but offers little control to the user.

Cine-AI only leverages machine learning for the initial definition of the style description dataset. At runtime, it relies on rules and parameters that warrant transparency, resource-efficient execution, and user control.

2.5 Storyboards

We are not aware of existing approaches leveraging storyboards in 3D cinematography. Consequently, we outline related work on alternative storyboard uses in videogames, and on similar uses in different creative domains.

In a different videogame application, Pizzi et al. [34] visualise simulated gameplay solutions to a level in the form of storyboards to retain designer control in interactive storytelling. Cine-AI similarly adopts storyboards as an established and familiar tool from game design. In contrast to Pizzi et al. [34] though, who visualise gameplay as a sequence of a character's actions, Cine-AI visualises the composition of a game cutscene as a sequence of cinematography techniques.

Baikadi et al. [3] bring storyboards to the domain of writing. They propose Narrative Theatre, a creativity support tool which leverages storyboards for narrative visualization. Users enter their narratives into the system, which are then fed through a natural language processor and a narrative reasoning module. Similar to Cine-AI, Narrative Theatre converts the resulting data into a storyboard, allowing quick iteration times and instant editing. Related, Skorupski and Mateas [40] introduce Wide Ruled, a story authoring tool harnessing the power of a plan-based story generation model. Wide Ruled uses a hierarchical structure of story actions and plot fragments to achieve a particular author-goal.

Only loosely related, Ronfard et al. [35] propose a storyboard language that describes each shot in terms of a sentence, which can be used to build software systems that convert formal shot descriptions into visual storyboard panels. This provides a way to automate the storyboarding process as well as virtual directing. Cine-AI in contrast uses storyboards as a visualisation and editing tool to present and tweak a produced scene. Acting as a summary of the user-customised scene, this interface allows to regenerate one or multiple shots, and to adjust the generation parameters. To the best of our knowledge, no such storyboard interface exists.

3 DIRECTOR STYLE DESCRIPTION DATASET

To imitate director styles in automatic cutscene generation, our toolset relies on data which encodes the style of a certain director (Supplementary Video, 00:42). We next describe the acquisition and analysis of a proof-of-concept dataset. To extend this dataset to other directors, one only needs to replicate the steps described in Sec. 3.2.

3.1 Selection of Directors

Our dataset captures the characteristic style of two eminent directors: Quentin Tarantino and Guy Ritchie. We chose these directors for three reasons:

- People commonly consider both directors to have well recognisable and unique shooting styles, making them good candidates for evaluating our system's ability to capture and emulate human-perceived style differences.

- Many cinematography techniques are dependent on post processing, audio and visual effects, i.e. technical domains that we chose not to consider at this stage of our project. The selected directors' styles differ particularly in terms of camera management – the present focus of our modelling work.
- Both directors can be assumed to be well known beyond an expert audience due to the wide popularity of their films, allowing for our results to be evaluated by a general readership.

3.2 Data Annotation & Aggregate Statistics

As basis for our dataset, we extracted 80 one-minute clips from the most highly rated movies of each director on IMDB[1]. For added representativity, we chose half of the clips to be action-heavy, and the other half to be strong on dialogue.

Technique	Description
God's Eye Shot	The camera is placed right above the subject, capturing an overhead angle.
Close-up Shot	A shot taken in close range, usually to show detail of the subject's face.
Master Shot	A distant shot of the entire scene and all characters, e.g. for dramatisation.
Pan Shot	The camera is placed directly to either side of the subject while moving only horizontally.
Medium Shot	The camera is placed so that the subject is visible waist-up (waist-shot).
Long Shot	The camera is placed far away from the subjects such that they appear as indistinct shapes.
Free Shot	Camera placement at any distance between close-up and long shot.
Close-up Zoom	Close-up shot combined with slow lens zoom towards the subject's face.
Quick Zoom	The camera quickly zooms towards the subject, usually for dramatisation of a reaction.
Dolly Zoom	The camera zooms towards the subject whilst moving further away from it, keeping the subject the same size in the frame, thus undermining normal background perception.
Stationary Tracking	The camera tracks the subject without changing location, only by rotation.
Handheld Tracking	The camera tracks the subject, but shakes as if moved by hand.
Steadycam Tracking	The camera tracks the subject with a gimbal lock, i.e. without rotational or locational noise.
Slow-Motion	The in-game time processing is slowed down, impacting every animation and effect.
Cut	Simplest transition, in that the camera only switches views.

Table 1. Annotated cinematography techniques, some descriptions drawing on Arijon's "Grammar of the Film Language" [2]. A video with examples of each technique can be found at https://youtu.be/egHGcp3zqks.

For each clip, we counted how often a specific cinematography technique was used. We selected 15 techniques in advance. based on their potential to be implemented via camera manipulation

---

[1]Internet Movie Database, http://www.imdb.com/

(Table 1). Each clip was also assessed in terms of its dramatisation level and the scene's pace, encoded as scalars between 0 and 1. We understand the dramatisation of a scene as the emotional intensity of character reactions within. We noted a high pace if the scene unfolds quickly. In contrast, we assigned low values to scenes with a calm character dialogue. For each of the 160 clips, we thus obtained a 17 element vector with 15 integers encoding how often a certain technique has been observed, and two real numbers representing dramatisation and scene pace.

To capture a director's style, we calculated aggregate statistics on the per-clip data. We determined how often a director used a specific technique, counting over all clips in the sample. Moreover, we calculated the means of the dramatisation and pace values over all techniques used by a specific director. These values allow us to capture the relationship between a director's use of a cinematography technique and the type of the scene.

3.3 Evaluation of Cinematography Techniques' Discrimination Potential

For Cine-AI to generate recognisable output for a given director, we need to ascertain that the selected cinematography techniques are distinguishing features of the directors. We trained a logistic regression [4, p. 205 ff.] model to predict the director of each annotated video clip, using the annotated cinematography technique frequencies as the regression features. Logistic regression models the probability of a director as proportional to $\sigma(w^T x + b)$, where σ is the logistic sigmoid function, x is a vector of features, w is a (transposed) vector of feature weights, and b is an optional scalar parameter. The absolute value of a feature weight can be interpreted as the feature's estimated importance in correctly predicting the director. If a feature is not at all predictive of a director, logistic regression will assign it a zero weight.

Technique	Weight	Technique	Weight	Technique	Weight
God's Eye Shot	0.84	Dolly Zoom	0.41	Free Shot	0.15
Steadycam Tracking	0.61	Close-up Zoom	0.39	Pan Shot	0.13
Handheld Tracking	0.59	Slow-Motion	0.34	Master Shot	0.07
Close-up Shot	0.53	Medium Shot	0.31	Stationary Shot	0.05
Quick Zoom	0.48	Long Shot	0.38	Cut	0.02

Table 2. Logistic regression weights for each feature (cinematography technique) in ascending order, representing their contribution to distinguishing our directors, Quentin Tarantino and Guy Ritchie.

The logistic regression model is able to predict the correct director with an accuracy of 83.75%, demonstrating that the selected features, although focusing on camera management only, can characterise exemplary director's styles. The feature weights are shown in Table 2 in descending order of their discrimination potential. The identified values confirm our observations from the initial movie clip annotation. A god's eye view for instance is more commonly used by Quentin Tarantino (33.7% of our clips) than by Guy Ritchie (8.75% of our clips). Ritchie also uses steadycam tracking more often (24.6%) than Tarantino (12.4%). Despite some regression weights being closer to zero, we did not discard any techniques in implementing Cine-AI, as they might later prove useful in discriminating between other directors.

We argue that selecting the sample clips differently would not yield strong variations in the inferred styles, as there exists strong style uniformity amongst different scenes shot by the same director – one of our primary motivations to mimicking the style of eminent human directors.

Order	Category	Techniques	Default
1	Positioning	Close-up Shot, God's eye Shot, Master Shot, Medium Shot, Long Shot, Pan Shot, Free Shot	Free Shot
2	Look	Quick Zoom, Dolly Zoom, Close-up Zoom, Stationary Shot	Stationary Shot
3	Tracking	Steadycam Tracking, Handheld Tracking, No Tracking	No Tracking
4	FX	Slow-Motion, No FX	No FX

Table 3. Cinematography techniques, including a default choice, mapped to camera manipulation categories. The categories enable a sequential camera manipulation workflow, in which a technique from each category is chosen and executed in the given order.

3.4 Categorisation of Techniques

The techniques used in the statistical analysis are realised through different means of camera manipulation. For instance, a close-up shot requires re-positioning the camera near the subject's face [32], and the quick zoom technique requires re-adjusting the camera lens' field of view towards the subject [43]. To guide Cine-AI's implementation, we assigned each of our technique to four categories of camera manipulation, ordered based on their sequential dependencies. Each of these categories in Table 3 contains a default technique to fall back to if Cine-AI can not determine a specific technique to use later on in the simulation.

The technique chosen from the first category, *Positioning*, defines where the system initially places the camera within the 3D scene geometry. Once placed, the camera always orients towards the user-defined target subject. Whether the camera lens is manipulated after this orientation step, e.g. in form of a zoom, depends on the corresponding technique selected from the *Look* category. In the next step, positioning and lens manipulation can be augmented by techniques in the *Track* category, defining potential means of moving the camera between two short markers. The final *FX* category can comprise any form of camera manipulation that does not fit into the earlier categories. Since our study does not focus on effect post-production, we only allow for a slow-motion effect, or no effects at all.

We note that dolly zoom could also be considered in the Positioning category, in that the camera is supposed to be moving on a line parallel towards the target. The Positioning category here however is concerned with static placements, not changes in position. We consider dolly zoom under the Look category, as it hosts techniques which require dynamic changes such as zoom ins/outs, field of view changes and similar.

4 CINE-AI SYSTEM AND USER INTERFACE

We next describe the processes and user interfaces through which Cine-AI accomplishes the procedural generation of in-game cutscenes in a target director's style, with an emphasis on the design-time user interaction. The following subsections provide detail on one or multiple design-time (sub-)processes, as illustrated and referenced in Figure 2. In the last two subsections, we elaborate on the system's runtime support for dynamic in-game cutscenes, and summarise the user's required and optional interaction with the system.

In order to support Cine-AI's straight-forward application in real-life game production, we have implemented the system in the popular Unity game engine. As an additional advantage, the Unity Editor provides an extensive set of tools for cutscene production, such as sequence and timeline editors, as well as immediate GUI libraries.
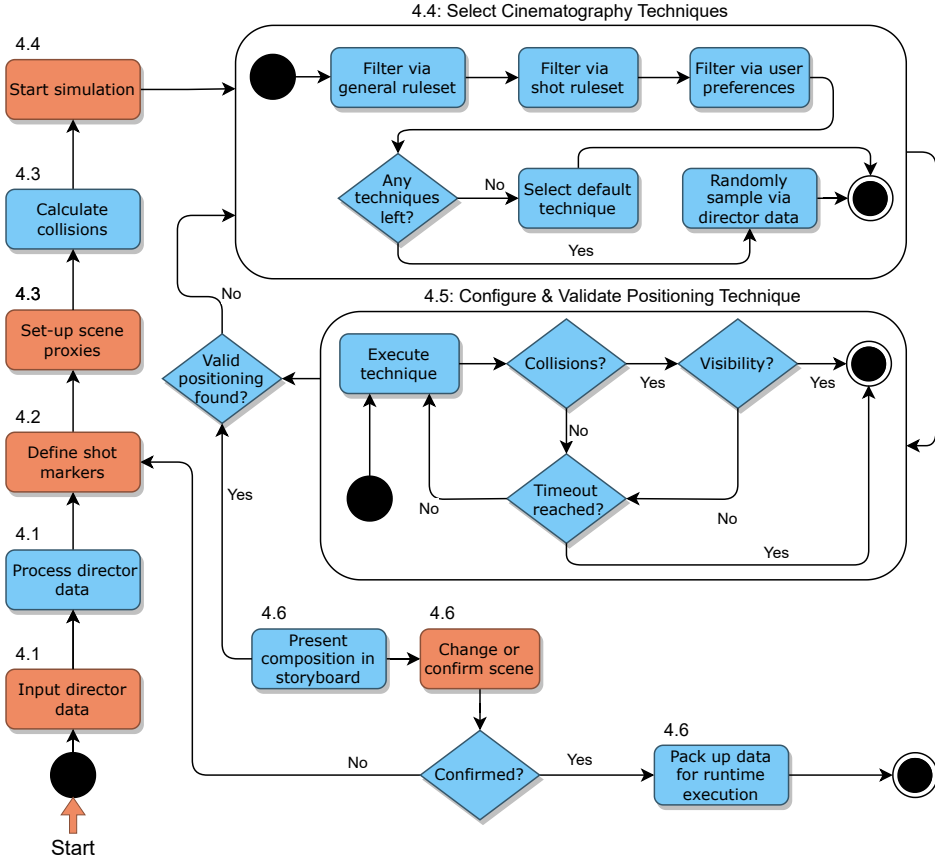
Fig. 2. UML Activity diagram of Cine-AI's design-time (sub-) processes (**rounded rectangles** - actions, **diamonds** - conditions, **circles** - start/end of overall or sub-process). **Orange** components require user interaction, **blue** ones are fully automated. Numbers refer to our subsections with detailed descriptions.

4.1 Director Data Input & Processing

The user initialises Cine-AI by inputting a director imitation dataset, obtained through the process described in Sec. 3.2. As a prerequisite to deciding which cinematography rules to abide by to generate camera behaviour in the style of the encoded director, Cine-AI calculates the conditional probabilities of each technique t to occur within a category C as

$$p(t|C) = \begin{cases} f_t / \sum_{t' \in C} f_{t'} & \text{, if } t \in C \\ 0 & \text{, otherwise.} \end{cases} \quad (1)$$

This processing step relies on f_t, the total frequency of a technique observed over all clips, as comprised in the dataset.

4.2 Defining Shot Markers

The user defines shot markers in the Unity Editor animation timeline (Figure 3a) to indicate at which time Cine-AI should insert cuts and transitions into the cutscene (Supplementary Video,

01:08). For each individual marker, the user can specify dramatisation and pace values, as well as which game object(s) the camera should focus on (Figure 3b) .

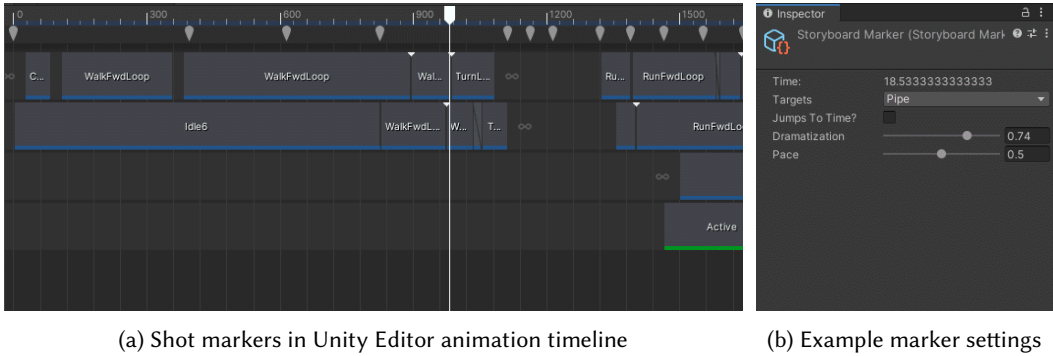(a) Shot markers in Unity Editor animation timeline   (b) Example marker settings

Fig. 3. Shot marker user interface. The user defines markers (**grey drop shapes**) in the Unity Editor animation timeline (a). The timeline comprises multiple tracks (**rows**) with animation and object activation nodes (**grey boxes with blue and green underlines, respectively**). For each marker (b), the user can set the camera target, and specify a desired transition dramatisation and pace.

4.3 Scene Proxies and Collision Information

To prevent camera clipping and realise collision-free camera paths, Cine-AI must obtain 3D collision information from the scene geometry. This calculation is expensive, but a cutscene is typically only set in a small part of the game world (Figure 4a). For increased efficiency, Cine-AI requires the user to set up scene proxies, i.e. 3D volumes that delimit the area in which the cutscene takes place, and for which collision information is required (Supplementary Video, 01:28).

(a) Initial scene   (b) User-defined scene proxy   (c) Collision information in proxy
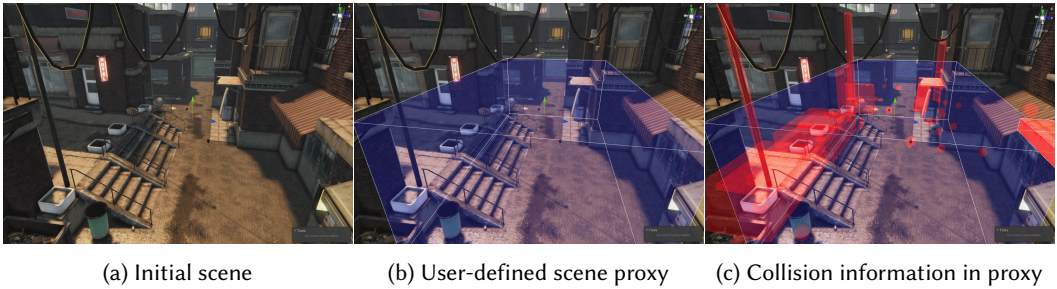
Fig. 4. Definition of scene proxies to obtain collision information. The user delimits in which area of the original game scene (a) the cutscene will unfold via a scene proxy (b, **blue box**). Cine-AI then calculates collision information (c, **red boxes**) within that area.

Once the user has set up the scene proxies (Figure 4b), Cine-AI calculates and serialises the collision data at design-time, enabling efficient collision avoidance during runtime. This calculation is repeated for each marker in the animation timeline, respecting the change of posture, position or orientation of the characters and objects in the scene (Figure 4c).

4.4 Selecting Cinematography Techniques

The user next initiates the simulation of all or specific shot markers (Supplementary Video, 01:40), yielding a set of cinematography techniques to be executed at the marker's time-step which (a)

adhere to general cinematography principles, (b) respect the user's settings, and (c) imitate the target director. Here, we elaborate on the first sub-process in this simulation: the selection of suitable cinematography techniques (Table 1) that satisfy these three requirements. To satisfy (a), the system filters the available techniques based on a general and shot-based ruleset, inspired by Kennedy and Mercer's Shotmaker system [25]. The remaining techniques are then filtered further based on their match with the user-defined dramatisation and pace values (b). To imitate the style of the targer director (c), the system finally samples one technique per category via the conditional probabilities determined from the director dataset (Section 4.1). This process is repeated for each category (Positioning, Look, Tracking, FX; Section 3.4) in order, allowing to select e.g. a special effect based on the preceding tracking style.

4.4.1 General Rule Set Filtering. Cine-AI tries abides by a set of general cinematography rules, e.g.:

Triangle Configuration: If multiple subjects are within view, the camera should be focusing the middle of a triangle determined by the position of each subject [2].

Rule of Thirds: To create a natural balance in the shot composition, the subject is placed on top of the cross-over points between imaginary lines [20]. The visibility of the subject is always prioritised over the rule of thirds.

Leading Subjects: In a continuously moving sequence, a tracking camera should come to rest before its target stops.

As first step in the selection of a set of cinematography techniques, each technique from a specific category is checked against the general rule set and filtered out if in violation. For instance, if the 3D geometry at a particular time, defined by the marker, does not permit a long distance shot, then techniques like long shot or master shot are removed.

Due to the interactive nature of games and the complexity of their 3D scenes, exactly abiding by all rules is often infeasible. To increase flexibility and customisability, Cine-AI allows users to tweak, or "bend", each rule through a set of individual parameters. For instance, users can weaken or intensify the rule of thirds by adjusting minimum and maximum shot distances, obedience thresholds, and visibility checking options.

4.4.2 Shot-based Rule Set Filtering. Cine-AI checks each technique that adheres to the general rule set against another set of rules designed to facilitate good shot compositions and avoid unwanted repetitions. In this second filtering process, the system incorporates information about the techniques decided for previous categories and preceding markers. The shot-based rule set includes the following key restrictions:

- It is not possible to use consecutive fast zoom techniques (quick zoom, dolly zoom).
- It is not possible to transit a master shot into a close-up shot. The camera should not be covering distances larger than a user-defined threshold at a single transition.
- It is not possible to use consecutive slow motion effects.
- It is not possible to apply any tracking technique that affects the camera position if dolly zoom is to be used in the Look category of the current marker.

4.4.3 User Preferences Filtering. The resulting set is further refined by comparing the dramatisation and pace values of each remaining technique, determined from the director's data, to the desired values designated by the user in the shot marker settings. Any technique that does not fit the user requirements is eliminated. The user can affect this process via threshold parameters, determining how strongly the marker's dramatisation and pace values influence the selection. Based on these settings, the user can also skip the definition of dramatisation and pace values entirely.

4.4.4   Director Data Sampling. If the preceding filtering has eliminated all techniques, Cine-AI selects the default technique for the respective category (Table 3). If exactly one technique is left, it is selected and the sub-process stops. If multiple techniques are left, Cine-AI samples one technique at random based on the conditional probabilities calculated from the director data. This ensures that the selection yields techniques that are close to the target director's style, while affording stochastic variation if the user chooses to repeat the simulation.

4.5   Executing, Configuring and Validating the Camera Positioning

The second simulation sub-process consists of configuring, executing and validating the previously selected camera positioning technique at the respective marker. It yields settings for camera placement, angle, path, alignment and lens; the *look*, *tracking* and *FX* techniques do not need further refinement, and are simulated at runtime.

4.5.1   Execute Technique. For each user-designated shot marker, Cine-AI firstly positions the camera according to the selected technique. For instance, the implementation of the close-up shot (Table 1) dictates that the camera must be within a meter from the target, while the master shot requires it to be sufficiently far away to capture the whole scene.

4.5.2   Validate Collisions. After the camera is placed, the collision data from the scene proxies (Section 4.3) is used to determine whether the placement is valid or not. In the latter case, Cine-AI tries a new positioning based on the randomisation properties of the executed technique until deemed valid, or a timeout is reached.

4.5.3   Validate Visibility. As a last step, the camera is oriented towards the subject point to perform visibility checks with a raycasting algorithm: a virtual capsule travels from the camera position towards the target position to check if there are any objects blocking the path in between. If such an occluder is detected, Cine-AI falls back to finding a new camera position and restarts the cycle. In case of a timeout, Cine-AI returns to the technique selection sub-process.

While our implementations of the cinematography techniques in Table 1 follow the most common cinematography rules [2], we anticipate projects in which these rules yield undesirable outcomes. For instance, Cine-AI implements Mascelli's [32] recommendation for close-up shots to depict the subject from the chest to above the head. However, this might not be fitting for non-humanoid characters. To provide more flexibility to its users, Cine-AI allows to customise each cinematography technique through a set of individual parameters.

4.6   Storyboard

Motivated by the prevalence of storyboarding in movie pre-production, Cine-AI provides the users with a storyboard interface as the central control element to review and tweak the scene composition at design-time, without running the game (Supplementary Video, 01:45). Cine-AI's storyboard visualises the composition (Section 4.4 and 4.5) by displaying each shot marker as an individual node in sequential order (Figure 5a), comprising various setting and a preview of the selected camera angles (Figure 5b). Moreover, the storyboard summarises information on, and affords control of the imported director data, scene proxy settings and simulation parameters (Figure 5a, left panel). After tweaking e.g. the individual techniques and cinematography rules, users can re-simulate the results for one or more markers. Each marker has a lock option to protect it from changes while re-simulating.

Once the user is satisfied and confirms the cutscene composition, Cine-AI serialises all shot data in Unity asset files for use at runtime. It comprises e.g. target subjects, locations, the selected
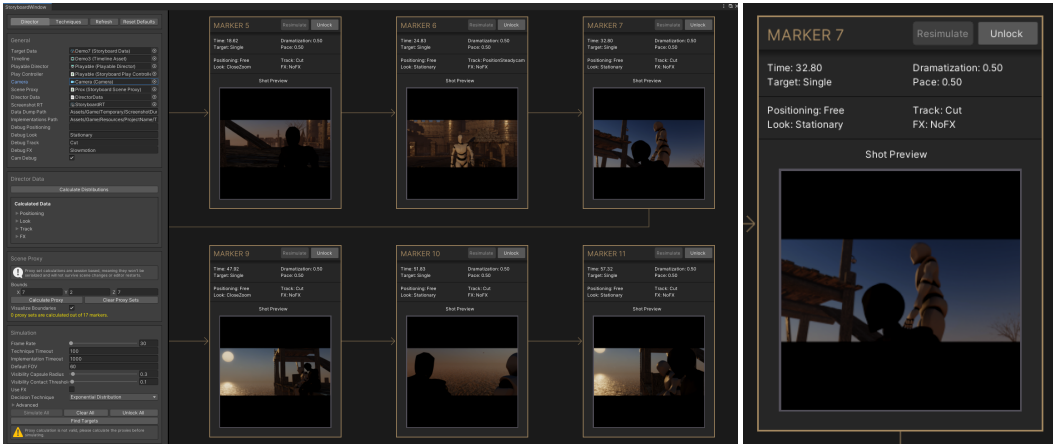
(a) Storyboard overview (b) Example storyboard node

Fig. 5. The storyboard user interface (a) comprises a vertical property window for editing simulation parameters on the left, and a node-based overview of each simulated shot marker on the right. The close-up view of a node (b) highlights a shot marker's parameters, selected cinematography techniques and shot preview.

cinematography techniques with custom settings, and the simulated camera positioning. Our data representation is derived from the FILM language [8], a processing language designed to characterise the behaviour of camera motion within any given 3D world geometry.

4.7 Runtime Execution

The storyboarding completes Cine-AI's design-time workflow; to play the in-game cutscene at runtime, the system listens to events fired as each shot marker is hit over time, and executes the corresponding cinematography techniques (Supplementary Video, 02:30). Although almost all required data is calculated at design-time, the system must still perform some operations at runtime to realise a crucial feature: its applicability to dynamic cutscenes. In such scenes, the behaviour of e.g. objects, animations and non-player characters (NPCs) can vary based on the preceding, player-induced game state; an NPC for instance might walk to different places in the scene, depending on the player's previous choices.

Pre-calculating animations that reach across multiple frames at design-time is only feasible for static cutscenes. This applies to all techniques from the *look*, *tracking* and *FX* categories, and Cine-AI consequently executes those at runtime. The potentially dynamic nature of the cutscene also implies that the 3D geometry might change in ways unanticipated during the design-time collision detection via scene proxies, thus necessitating runtime collision avoidance. If the camera penetrates an object, Cine-AI performs Haigh-Hutchinson's algorithm [17] to move the camera along the object's surface. These runtime calculations require some additional computational resources, but enable the user to create only a single cutscene composition through Cine-AI, which can then adapt flexibly to the game at runtime.

4.8 Summary of User Interaction

We briefly summarise which steps (in **bold**) the user has to or can optionally perform in the interaction with Cine-AI, in which order, and to which end. As prerequisites, the user must perform the following steps in any order:

Selection of Director Dataset (4.1): The user must select a director dataset file, which will enable Cine-AI to imitate the director's style by calculating the corresponding thresholds during the procedural shot generation.

Shot Marker Definition (4.2): Users is expected to have their cutscene animations ready on the timeline and define transition points within by means of shot markers. This is required for Cine-AI to parse the timeline and generate a shot sequence per marker, which will then be combined in the sequential storyboard representation.

Scene Proxy Setup (4.3): The user must create scene proxy objects and setup their boundaries. This is required for Cine-AI to collect collision information from the scene pre-generation.

When everything is ready, users must **hit the generate button (4.4)** to create a shot sequence based on the markers, collision data, director dataset and parameters, which will be presented on the storyboard. Users can **re-generate the whole storyboard or just a collection of shots** if they desire. Users can **fine-tune technique-specific parameters** such as distance limitations, field-of-view limits and movement acceleration. They can moreover **adjust global parameters** such as timeout thresholds to tune Cine-AI's execution speed on specific hardware. The shot sequence data is serialized immediately upon generation, and users can **switch into play-mode** to see their results in runtime.

5 EXPERIMENT 1: DIRECTOR STYLE IMITATION

In our first experiment, we probe Cine-AI's ability to generate cutscenes that people can correctly associate with a specific target director. In a single-session, within-subjects study conducted through videoconferencing and online forms, we asked participants to associate cutscenes generated by Cine-AI with either director, Guy Ritchie or Quentin Tarantino. We then assessed quantitatively (a) whether the generated cutscenes for one director were consistently perceived as different from the other's, and (b) whether they were associated with the correct director. We complement these insights with qualitative information on our participants' decision-making process.

5.1 Materials

We produced eight different cutscenes with Cine-AI, half of which were shot based on Tarantino's and Ritchie's input data, respectively. Each cutscene is based on a different game scene. In order to adequately represent the defining features of our directors' styles, we chose cutscenes with action-rich content such as shooting, chasing and fighting. As many in-game cutscenes also feature steady dialogue scenes, we additionally produced a few calmer scenes, balanced between the directors. Each cutscene lasts thirty to sixty seconds, and the overall duration is the same for both directors.

   In order to brief participants without pre-existing cinematic knowledge, we moreover prepared two mashup videos of real movie clips demonstrating the directors' iconic cinematographic style. These two reference videos have a length of two and a half minutes each. The reference and stimuli clips were embedded in an online form, starting with the reference clips but shuffled randomly within each group. The mashup videos and eight cutscenes can be viewed online[2].

5.2 Methods

We employed a Binomial test to assess whether the generated cutscenes for one director were consistently perceived as different from the other's. The null-hypothesis is that our participants associate the stimuli cutscenes with one of the two directors by random chance, i.e. $p_0 = 0.5$. The Binomial test tells us whether this null hypothesis should be rejected, by measuring how significantly our participants' associations deviate from this hypothesised distribution. We performed the test

---

[2]YouTube playlist: https://youtube.com/playlist?list=PLm0M-oJsXvHpAu7dFCxTxcnoSVQI7ekzL.

on the pooled answers from all participants, which requires the repeated measures, i.e. associations, obtained from each participant to be independent. We consider this assumption valid as no feedback was given about the correctness of the associations, preventing any learning effects.

This null hypothesis would also be rejected if our participants associated the cutscenes consistently, i.e. beyond random chance, but with the "wrong" director. To validate the correct association of the generated cutscenes with the target director, we calculated the accuracy for each participant.

We also calculated two inter-rater reliability measures to determine the conformity of our participants' individual associations [16]. We firstly calculated Fleiss' Kappa [14], an extension of Cohen's Kappa to more than two raters. We backed this up with a two-way mixed, average score ICC(3,k) Intraclass Correlation (ICC) analysis [39]. This model requires that each subject is measured by k fixed raters, and that the measures are averaged for each subject. This holds, as each cutscene was assessed by each participant, who were the only raters of interest.

We complement these quantitative measures with qualitative data from optional questionnaire responses and free-form discussion to learn about the participant's confidence in deciding on the directors, and how it was affected by our procedure. It comprised the following statements on 5-point Likert (Strongly Agree - Strongly Disagree) scales:

- It was really hard for me to decide throughout the process.
- I felt like I was answering randomly.
- I needed longer clips/more time to accurately decide.
- The reference clips for the directors were helpful for me while deciding.

The questionnaire was provided as an online form and is included in our Supplementary Material.

5.3 Participants

We recruited volunteer adult participants via call-outs and advertisements by the research group through Telegram groups and Discord. Participants were deemed eligible if expressing some affiliation with movies and video games. Exclusion criteria included lack of sleep, being under the influence of drugs or alcohol, as well as experiencing any digestive, muscle or organ pain, or emotional distress. Participants were not incentivised.

In total, 18 participants took part in the study. Three identified as female (16%), 14 as male (77%) and one chose to not provide demographics information. The reported age ranged from 22 to 41 years ($Median = 27, SD = 5.85$). 18 participants are sufficient as we do not need to divide the population for comparing multiple experimental conditions.

5.4 Procedure

The study was performed via individual online videoconferencing sessions between one participant and experimenter. We asked each participant to provide informed consent to participate in the study. They were then briefed about the Cine-AI system and its purpose. The goal of the next step was to familiarise the participant with the styles of our two directors. As preparation, we introduced them to basic camera manipulation techniques and the notion of directing style. The participants were then given access to an online form, in which they could watch the mashup clips for Quentin Tarantino and Guy Ritchie, respectively. They were asked to consider the main differences between shooting styles and camera management between the clips, and to pay attention to potential signature techniques. The participants were informed that they could watch these reference clips again at any time during the experiment.

Once they were ready to proceed, our participants answered two optional demographics questions about their age and gender in the same form. This allowed them to rest their minds briefly before

Scene	Description	Director	Correct %
Escape	Fight action	Tarantino	88.8
Meeting	Thriller	Tarantino	77.7
Facility	Spy action	Tarantino	77.7
Village	Drama	Tarantino	66.6
Bridge	Driving action	Ritchie	94.4
Alley	Foot chase	Ritchie	100
Night	Murder mystery	Ritchie	72.2
Port	Drama	Ritchie	55.5

Table 4. Cutscene information and percentage of correct predictions.

commencing with the main part of the experiment, for which they were asked to watch each stimulus cutscene and associate it with one of our two directors.

The study concluded with a questionnaire and free-form discussion about our participants' decision-making, as described earlier. We provide the briefing material and the main experiment form with one randomisation of the reference and stimulus clips as Supplementary Material.

5.5 Results

Pooling the associations of 18 participants on eight stimuli each, we obtained an effective sample size of $n = 144$. The Binomial test on these measurements yielded $p = 9.93 \times 10^{-13}$ ($x = 114$) with a 95% confidence interval $[0.716, 0.855]$, indicating that the null hypothesis of the associations being random should be rejected. This supports that Cine-AI can produce cutscenes that people consistently associate with a certain director beyond random chance.

The individual accuracy scores, reported in Table 4, moreover support that the cutscenes were correctly associated with the target director. In total, 79% of the 144 associations were correct. Further analysis of the individual scores highlights that some cutscenes were associated with considerably lower (*Village*, *Port*) or higher (*Bridge*, *Alley*) success than others. Strikingly, all 18 participants correctly recognised that the *Alley* scene was shot with Guy Ritchie's data.

This is complemented with Fleiss $\kappa = 0.382$ ($z = 13.4, p = 0$), indicating *fair* inter-rater agreement [27] between our participants. We moreover obtained an ICC value of $r = 0.93$ ($p = 5 \times 10^{-13}$) with a 95% confidence interval $[0.85, 0.98]$. Based on this upper and lower confidence bound, our ICC inter-rater reliability is between *good* and *excellent* [26].

The semi-structured interview revealed more nuanced information on our procedure's reliability. Most participants agreed to having experienced difficulty in associating some scenes due to their slow pace, as this was uncharacteristic of both directors. The general opinion was that the *Port* scene was hardest to assess, "due to the lack of shooting and fighting". The majority of participants agreed that the reference clips were extremely helpful for their decision-making, with the exception of three participants who criticised that the clips made it too easy to decide on a few cutscenes. Beyond the fixed questions, the interviews yielded that participants generally enjoyed the cutscenes generated for the study and found them suitable to tell a story. Most stated that they can spot the effect of the Cine-AI and how it tries to mimic a particular director. However, three participants mentioned that the absence of more realistic characters, lip-sync and eye animations, as well as a well-mixed sound design made it harder to associate the scenes, because they tried to infer the director based on characters, emotions and music, rather than through the camera manipulation.

6 EXPERIMENT 2: USABILITY

While our first experiment was dedicated to testing Cine-AI's functional value, we conducted this second experiment to probe the toolset's usability for game designers as required for its professional adoption. We wanted to (a) determine Cine-AI's overall usability, (b) identify any major flaws affecting usability and user experience, and (c) learn how the toolset could be improved further. In individual videoconferencing sessions, we asked our participants to interact with Cine-AI, installed on their own computer, on a predefined sequence of tasks. In each of these single sessions, we assessed usability by means of standardised quantitative, and with qualitative methods.

6.1 Materials

The main material for this usability study is the Cine-AI software. Implemented as a plug-in for the Unity game engine, its installation is straight-forward, but requires Unity, e.g. in form of the free Personal Edition, to be pre-installed.

Fig. 6. The "Living room" 3D scene utilised in our usability experiment. Three characters, identified with grey, orange and red colors, have been animated to walk around and enact a conversation.

We moreover developed a simple 3D scene (Figure 6, in the GitHub repository) to provide each participant with a consistent basis for their individual interaction with Cine-AI's user interfaces. It includes various characters, engaged in a dialogue and walking around in a living room environment. To avoid distraction, the scene does not contain any gameplay elements, but only the cutscene animations ready to be played, with no camera motion attached to it.

6.2 Methods

We assessed usability qualitatively and quantitatively. We performed Concurrent Think-Alouds (CTAs) [13] to gain qualitative insights into our participants' thoughts while interacting with Cine-AI, potentially uncovering flaws and future improvements (b, c). This method has been well-vetted for the accurate collection of usability data [11], and requires participants to verbalise their thoughts while performing a variety of tasks.

We moreover employed the System Usability Scale (SUS) [6, 28] to obtain a standardised, quantitative measure of Cine-AI's usability (a). Initially developed as a quick means to assess software usability – comprising only ten items on 5-point Likert (Strongly Agree - Strongly Disagree) scales – it has proven a reliable instrument for the end-of-test subjective assessments of usability, yielding an easily interpretable score from 0 to 100% [28].

We finally conducted a a semi-structured interview to collect more targeted, qualitative information on flaws (b) and potential improvements (c) in the system's usability, and to receive feedback on the study and procedure.

6.3 Participants

Since Cine-AI is implemented in Unity, and we wanted to keep interference with general, experience-related usability issues minimal, we required our participants to have prior practice in using the game engine. They had to be capable of navigating the Unity user interface, using the timeline tool, and test their work both in the editor and game modes. To accommodate this requirement, we recruited our study participants through the Unity online forum and a Telegram group for [removed for anonymisation] Game & Media students. The eligibility and exclusion criteria were otherwise identical to our first experiment. We did not provide incentives for participation.

In total, twelve adult participants took part in the experiment, of which three identified as female (25%) and nine as male (75%). Participants included professional animators, 3D artists, and Unity developers. We consider this sufficient, since we do not compare experimental conditions and complement usability scores with qualitative data.

6.4 Procedure

The study was performed in individual videoconferencing sessions between one participant and the experimenter. The participant was provided the Cine-AI plug-in for installation with their local copy of Unity ahead of the study session (https://github.com/inanevin/Cine-AI). In the beginning of the live session, they received a briefing on the capabilities and purpose of the Cine-AI toolset (Supplementary Material). As a consistent primer to the toolset, they were shown the windows and asset files required to set-up and use the system.

For the main part of the study, the participants were asked to think aloud while performing various tasks, thus performing a CTA. The tasks consisted of the initial system set-up, importing the sample director data, defining scene proxies, adding shot markers in the timeline, running the simulation, and tweaking the results through the storyboard interface. The experimenter took notes of any thoughts and comments related to Cine-AI's usability.

Upon completion of the demo scene simulation, participants were asked to fill in the SUS questionnaire and provide optional demographics information on their age and gender. This was followed by the semi-structured interview to identify additional usability flaws and means of improvement. The main questionnaire with the demographics and SUS items, as well as the optional, semi-structured interview questions are comprised in the Supplementary Material.

6.5 Results

Across all participants, Cine-AI received a mean SUS score of 74.36% ($Median = 77.5, SD = 17.50$). Compared against a reference dataset [37] from 446 studies and over 5000 individual SUS responses ($Mean = 68, SD = 12.50$), this corresponds to a percentile rank of 70% [38], indicating that Cine-AI's usability is above average and better than seven out of ten reference systems. On a curved grading scale from A to F [38], Cine-AI receives a usability grade B.

The CTA did not uncover any major usability issues; all participants were able to work through the given tasks without severe problems. Regarding minor issues, some participants lamented the lack of tooltips with more information on the adjustable parameters. The storyboard interface was generally well received, with participants highlighting its professional look and usefulness in visualising their composition. Some participants suggested that the storyboard could be improved by offering better layout options and the ability to use multiple aspect ratios for shot previews. Three participants complained about the set-up procedure of the toolset and one requested a wizard to automate it further. Additional feature requests concerned Cine-AI's embedding into the Unity game engine. Two participants expressed a preference for completely integrating the toolset within Unity's own tools, in particular runtime editing support and animation baking in Unity's timeline.

7 DISCUSSION

Our first experiment (Section 5) demonstrates Cine-AI's capability to generate in-game cutscenes in a target director's style that is recognisable and distinguishable by its users. The participants in our second experiment (Section 6) did not identify any fundamental flaws to the system's usability, and indicated that Cine-AI has potential to be adopted in professional cutscene production. We next discuss present and permanent constraints to Cine-AI's functionality and usability, critically reflect on limitations to our experiments, and propose how these can be overcome in future work.

7.1 Director Style Imitation

For any selection of directors, especially if working in a similar genre, individual style can rarely be perfectly distinguished. This is because style evolves over time and typically manifests only in specific types of scenes. The latter observation can explain the outliers in our first experiment: our participants likely experienced strong difficulty in correctly associating the *Village* and *Port* scenes with their target directors, because both scenes contained more dialogue and less action, while our directors' iconic style is mostly expressed in action and fast-paced content. We explain our participants' strong success rate in correctly associating the *Alley* scene with Guy Ritchie by the presence of a chase sequence that was extremely similar to a scene in Ritchie's reference clips. These properties of style impose fundamental limitations to the accuracy by which Cine-AI or any similar system could imitate a particular director.

We also see room for improvement, based on the observation that not all directors solely focus on camera work to express their style. Michael Bay for instance notoriously employs many post-processing effects such as lens flares and god rays. To increase Cine-AI's fidelity in distinguishing and reproducing a director's unique style, we thus suggest to dedicate some future work to implementing and annotating additional cinematography techniques of different types, including post-processing effects. Moreover, we recommend to encode and interpret more information about the ground truth clips beyond the use of specific techniques, e.g. concerning the characters and scene objects, to additionally boost Cine-AI imitation capacity. While such additional features only have to be implemented once, they crucially require more review and annotation work in the creation of every new director imitation dataset.

We consider our style imitation experiment to be presently limited by the choice of Quentin Tarantino and Guy Ritchie as target directors. While allowing for a solid proof-of-concept due to the characteristics discussed in Section 3.1, both directors focus on the action genre and exhibit distinctive and iconic styles that can be recognised with only moderate difficulty. To better assess Cine-AI's fidelity and universal applicability, future research must evaluate its capacity to imitate additional directors working in other genres and with less clearly distinguishable styles.

Both directors are moreover white and western men, and future efforts in extending the dataset should focus on increasing director diversity. We would be intrigued to see how well Cine-AI can for instance reproduce the style of the Indonesian director Timo Tjahjanto, who works on horror and action movies and employs superior camera work in extremely fast-paced scenes without creating visual discomfort. The American director Spike Lee focuses on color and race relations and is well known for his frequent use of dolly shots to let characters "float" through their surroundings. Finally, the Belgian-born French director Agnes Varda has been highly praised for her unique style of using the camera "as a pen"; this provides a worthwhile challenge to procedural cinematography.

We finally note that our dataset is limited by the selection of ground-truth clips and their annotation being dependent on the authors' cinematography knowledge. The dataset creation process affects Cine-AI's performance and, in comparison against human judgement, its imitation accuracy scores. As a straight-forward means to reduce subjective bias in the future, we propose

employing more annotators, and validating the reliability of the resulting dataset with inter-rater agreement measures. As a more ambitious plan, we envision to overcome such bias, as well as avoiding an increase in labour due to more features, by automating the annotation process via machine learning. This is ambitious, in that the respective algorithms must be capable of differentiating cinematography techniques from image sequences, which requires advances to the current state of the art.

7.2 Usability

Our usability study is limited in that participants were subjected to a Cine-AI version that was not tightly integrated into the Unity game engine and editor, potentially impeding usability. While this decoupling was intentional to allow for the straight-forward migration to other systems such as the Unreal Engine, we aim to accommodate our participant's engine-specific feature requests and leverage more intuitive and accessible native UI components in future work.

We moreover propose to improve Cine-AI's usability through modifications to the timeline and storyboard, our two core UI components. At present, Cine-AI simulates all markers once and presents them in the storyboard, through which the user can re-simulate all or individual markers for fine-tuning. In future versions, we want to generate multiple options for each marker for the user to select from in the storyboard, thus alleviating effort in re-simulating markers.

We plan to improve the usability and functionality of our storyboard by allowing users to define their own cinematography rules based on existing, specialised declarative languages [9]. Combined with a constraint solver to prevent rule conflicts, such user-defined rulesets would complement presently hard-coded rules such as Arijon [2]'s idioms (Sec. 4.4), increasing Cine-AI's expressive power and customisability.

7.3 Performance and Automation

We highlight two avenues for increasing Cine-AI's performance and automation, drawing on related work (Sec. 2). In order to achieve faster and more accurate camera placement, future work should be dedicated to implementing Lino and Christie's Toric Space [30]. This novel camera space representation uses a triplet of Euler angles to define a camera viewpoint around a pair of targets, enabling automated viewport computation. It represents a fast, uniform way of determining possible camera placements given any target in 3D space. At present, each cinematography technique in Cine-AI must define its own camera placement rules. We suggest to overcome this by integrating Toric Space in Cine-AI, and only exposing the calculation parameters. This would make it considerably easier to include new cinematography techniques: a new instance would be defined by a set of manipulation parameters alone.

The use of Toric Space opens up a second avenue for future work. Cine-AI's scene proxies can be slow to compute in large, complex 3d scenes and on lower-end hardware as they rely on checking all collision information within the volume and serializing the data for cinematography technique calculations. One option for future work is to turn these into runtime calculations with an efficient and precise algorithm such as Burg et al.'s occlusion maps [7], which relies on Lino and Christie's Toric Space [30]. Such an algorithm would ensure high-rates of subject visibility while obeying the cinematography rules. Moreover, it would eliminate the need to improve the present runtime camera helpers, thus providing better responses in fully dynamic scenarios.

7.4 System Adoption

We hope that Cine-AI's capacity to imitate a given director's style, its usability, and its implementation in the popular and freely available game engine Unity will foster adoption in real-life game projects. Crucially, we do not expect automation through Cine-AI to take away jobs in game

industry. We consider our research and the developed tool to be particularly valuable for independent game developers, who can typically not afford a dedicated production team for creating in-game cinematography in the first place. We do not expect Cine-AI to completely replace the process of manual directing, but rather see its potential to co-create an initial cutscene in the consistent and uniform style of a certain director without specialised cinematography expertise. The result can then be tweaked further by the independent game designer to fit their game's unique style. The cinematography requirements of AAA games tend to get extremely sophisticated in terms of style, scene duration and size. While AAA companies will thus likely continue relying on dedicated production teams to achieve the desired level of cinematographic quality, they can utilise Cine-AI to create prototypes for their cinematography design, automatically generate shots to inspire new ideas, or use the storyboard feature to quickly iterate on possible shots, following a specific directorial style.

8 CONCLUSION

We have presented Cine-AI, a novel, semi-automated cinematography toolset capable of procedurally generating in-game cutscenes in the style of a specific movie director. The system combines a highly configurable design-time workflow with runtime support, allowing for cutscenes to unfold dynamically based on the game state. Through extensive tools, including a novel storyboard interface, the user can adjust simulation properties and tweak the final scene composition. Via two user studies, we provided evidence that Cine-AI can generate cutscenes that people correctly associate with a specific target director, and that the system provides above-average usability. We point out present limitations of our dataset and study, and discuss potential means of adopting Cine-AI for production in game industry studios of different sizes. We highlight opportunities for future work to improve style imitation fidelity, increase design- and runtime efficiency, foster extensibility, and improve automation further.

To support Cine-AI's adoption and future extension, we made our proof-of-concept dataset and the source code publicly available under an open-source license. We invite researchers, (game) developers and film enthusiasts to join us in taking this project further.

9 ACKNOWLEDGMENTS

REFERENCES

[1] Dan Ablan. 2002. *Digital Cinematography & Directing.* Pearson Education.
[2] Daniel Arijon. 1991. *Grammar of the Film Language.* Silman-James Press.
[3] Alok Baikadi, Julius Goth, Christopher Mitchell, Eun Ha, Bradford Mott, and James Lester. 2011. Towards a Computational Model of Narrative Visualization. In *Proc. Conference on Artificial Intelligence and Interactive Digital Entertainment.* AAAI, 2–9.
[4] Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning.* Springer.
[5] Rogerio Bonatti, Cherie Ho, Wenshan Wang, Sanjiban Choudhury, and Sebastian A. Scherer. 2019. Towards a robust aerial cinematography platform: Localizing and tracking moving targets in unstructured environments. In *Proc. Conference on Intelligent Robots and Systems.* IEEE, 229–236.
[6] John Brooke. 1996. SUS: A Quick and Dirty Usability Scale. In *Usability Evaluation in Industry.* CRC Press, 189–195.
[7] Ludovic Burg, Christophe Lino, and Marc Christie. 2020. Real-time Anticipation of Occlusions for Automated Camera Control in Toric Space. *Computer Graphics Forum* 39, 2 (2020), 523–533.
[8] Fred Charles, Jean-Luc Lugrin, Marc Cavazza, and Steven J. Mead. 2002. Real-Time Camera Control for Interactive Storytelling. In *Proc. GAME-ON.* 73–76.
[9] David Christianson, Sean Anderson, Li-wei He, David Salesin, Daniel Weld, and Michael Cohen. 1996. Declarative Camera Control for Automatic Cinematography. *Proc. National Conference on Artificial Intelligence* 1 (1996), 148–155.

[10] Marc Christie and Patrick Olivier. 2009. Camera Control in Computer Graphics: Models, Techniques and Applications. In *SIGGRAPH ASIA Courses*. ACM, Article 3, 197 pages.

[11] Lynne Cooke. 2010. Assessing Concurrent Think-Aloud Protocol as a Usability Test Method: A Technical Communication Approach. *Trans. Professional Communication* 53, 3 (2010), 202–215.

[12] Steven M. Drucker and David Zeltzer. 1994. Intelligent Camera Control in a Virtual Environment. In *Proc. Graphics Interface*. 1–9.

[13] K. Anders Ericsson and Herbert A. Simon. 1984. *Protocol Analysis: Verbal Reports as Data*. MIT Press.

[14] Joseph L. Fleiss and Jacob Cohen. 1973. The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educational and Psychological Measurement* 33, 3 (1973), 613–619.

[15] Marco Furini, Filippo Geraci, Manuela Montangero, and Marco Pellegrini. 2010. STIMO: STIll and MOving video storyboard for the web scenario. *Multimedia Tools And Applications* 46 (2010), 47–69.

[16] Natasa Gisev, J. Simon Bell, and Timothy F. Chen. 2013. Interrater Agreement and Interrater Reliability: Key Concepts, Approaches, and Applications. *Research in Social and Administrative Pharmacy* 9, 3 (2013), 330–338.

[17] Mark Haigh-Hutchinson. 2009. *Real-Time Cameras: A Guide for Game Designers and Developers*. Chapter 9 - Motion and Collision, 313–364.

[18] Shakil A. Hannan, Daisuke Endou, and Masanobu Yamamoto. 2005. Director-Oriented Shot Assessment and Evaluation in Virtual Cinematography. *The Journal of the Institute of Image Information and Television Engineers* 59, 4 (2005), 592–603.

[19] Li-wei He, Michael Cohen, and David Salesin. 1996. The Virtual Cinematographer: A Paradigm for Automatic Real-Time Camera Control and Directing. *Proc. SIGGRAPH Conference on Computer Graphics*, 217–224.

[20] Timothy Heiderich. 2012. Cinematography Techniques: The Different Types of Shots in Film. *Videomakers* 22 (2012).

[21] Arnav Jhala and R. Young. 2005. A Discourse Planning Approach to Cinematic Camera Control for Narratives in Virtual Environments. *Proc. National Conference on Artificial Intelligence* 1, 307–312.

[22] Hongda Jiang, Bin Wang, Xi Wang, Marc Christie, and Baoquan Chen. 2020. Example-driven virtual cinematography by learning camera behaviors. *Trans. on Graphics* 39, 4 (2020), 45–1.

[23] Anna Kantosalo and Hannu Toivonen. 2016. Modes for creative human-computer collaboration: Alternating and task-divided co-creativity. In *Proc. Conference on Computational Creativity*. 77–84.

[24] Peter Karp and Steven Feiner. 1993. Automated Presentation Planning of Animation Using Task Decomposition With Heuristic Reasoning. In *Proc. Graphics Interface*. 118–127.

[25] Kevin Kennedy and Robert Mercer. 2002. Planning Animation Cinematography and Shot Structure to Communicate Theme and Mood. In *Proc. Symposium on Smart Graphics*. 1–8.

[26] Terry K. Koo and Mae Y. Li. 2016. A Guideline of Selecting and Reporting Intraclass Correlation Coefficients for Reliability Research. *Journal of Chiropractic Medicine* 15, 2 (2016), 155–163.

[27] J. Richard Landis and Gary G. Koch. 1977. The Measurement of Observer Agreement for Categorical Data. *Biometrics* (1977), 159–174.

[28] James R. Lewis. 2018. The System Usability Scale: Past, Present, and Future. *International Journal of Human−Computer Interaction* 34, 7 (2018), 577–590.

[29] Christophe Lino and Marc Christie. 2012. Efficient Composition for Virtual Camera Control. In *SIGGRAPH/Eurographics Symposium on Computer Animation*. ACM, 65–70.

[30] Christophe Lino and Marc Christie. 2015. Intuitive and Efficient Camera Control with the Toric Space. *Trans. on Graphics* 34, 4 (2015), 1–12.

[31] Christophe Lino, Marc Christie, Fabrice Lamarche, Schofield Guy, and Patrick Olivier. 2010. A Real-time Cinematography System for Interactive 3D Environments. In *Proc. SIGGRAPH/Eurographics Symposium on Computer Animation*. ACM, 139–148.

[32] Joseph V. Mascelli. 1965. *The Five C's of Cinematography: Motion Picture Filming Techniques*. Cine-Graphic Publications.

[33] Thomas Oskam, Robert W Sumner, Nils Thuerey, and Markus Gross. 2009. Visibility Transition Planning for Dynamic Camera Control. In *Proc. SIGGRAPH/Eurographics Symposium on Computer Animation*. ACM, 55–65.

[34] David Pizzi, Marc Cavazza, Alex Whittaker, and Jean-Luc Lugrin. 2008. Automatic generation of game level solutions as storyboards. In *Proc. Conference on Artificial Intelligence and Interactive Digital Entertainment*. AAAI, 96–101.

[35] Rémi Ronfard, Vineet Gandhi, Laurent Boiron, and Vaishnavi Ameya Murukutla. 2013. The Prose Storyboard Language: A Tool for Annotating and Directing Movies. In *Proc. Workshop on Intelligent Cinematography and Editing at the Foundations of Digital Games Conference*. 1–8.

[36] James Rucks and Nikolaos Katzakis. 2021. CamerAI: Chase Camera in a Dense Environment using a Proximal Policy Optimization-trained Neural Network. In *Proc. Conference on Games*. IEEE, 1–8.

[37] Jeff Sauro. 2011. *A Practical Guide to the System Usability Scale: Background, Benchmarks & Best Practices*. Measuring Usability LLC.

[38] Jeff Sauro and James R. Lewis. 2012. *Quantifying the User Experience.* Morgan Kaufmann, Chapter 8 - Standardized Usability Questionnaires, 185–240.

[39] Patrick E. Shrout and Joseph L. Fleiss. 1979. Intraclass Correlations: Uses in Assessing Rater Reliability. *Psychological Bulletin* 86, 2 (1979), 420.

[40] James Skorupski and Michael Mateas. 2009. Interactive Story Generation for Writers: Lessons Learned from the Wide Ruled Authoring Tool. In *Proc. Digital Arts and Culture.* 1–10.

[41] Edirlei Soares de Lima, Cesar Pozzer, Marcos d'Ornellas, Angelo Ciarlini, Bruno Feijó, and Antonio Furtado. 2009. Virtual Cinematography Director for Interactive Storytelling. In *Proc. Conference on Advances in Computer Entertainment Technology.* 263–270.

[42] The Editors of Encyclopaedia Britannica. 2021. "Cinematography". In *Encyclopedia Britannica.* https://www.britannica.com/topic/cinematography

[43] Paul Wheeler. 2005. *Practical Cinematography.* Taylor & Francis.

[44] Ke Xie, Hao Yang, Shengqiu Huang, Dani Lischinski, Marc Christie, Kai Xu, Minglun Gong, Daniel Cohen-Or, and Hui Huang. 2018. Creating and Chaining Camera Moves for Quadrotor Videography. *Trans. on Graphics* 37 (2018), 1–14.

[45] R. Young and Johanna Moore. 1994. DPOCL: A Principled Approach to Discourse Planning. In *Proc. Workshop on Natural Language Generation.* 1–8.