



US 20250265426A1

(19) **United States**

(12) **Patent Application Publication**
GAGLIANO et al.

(10) **Pub. No.: US 2025/0265426 A1**

(43) **Pub. Date: Aug. 21, 2025**

(54) **AUTOMATED NARRATIVE PRODUCTION
SYSTEM AND SCRIPT PRODUCTION
METHOD WITH REAL-TIME INTERACTIVE
CHARACTERS**

H04N 21/44 (2011.01)

H04N 21/84 (2011.01)

(52) **U.S. CL.**

CPC **G06F 40/56** (2020.01); **G06F 40/103**
(2020.01); **H04N 21/44** (2013.01); **H04N**
21/84 (2013.01)

(71) Applicant: **TRANSITIONAL FORMS INC.**,
Toronto (CA)

(72) Inventors: **Pietro GAGLIANO**, Toronto (CA);
Dante Camarena JIMÉNEZ, Toronto
(CA); **Nicholas COUNTER**, Toronto
(CA); **Liam Atticus CLARKE**, Toronto
(CA)

(57)

ABSTRACT

A script production method and a show production system, in which a user has a script seed input that the user wishes to be the starting narrative or starting point for a video. In an example, the script production method includes: receiving the script seed input; generating, using the script seed input and a script writer module: a script which continues the narrative; generating, using the script and a script analysis module: respective segment metadata for the script; and generating, using the script, the respective segment metadata and one or more rendering modules: one or more video segments. An interface screen on a device can be used to add, edit, delete, or approve proposed script segments of the script in real-time after the video segments are generated, in which those approved proposed script segment are also generated into video segments which further continue the narrative.

(21) Appl. No.: **19/199,758**

(22) Filed: **May 6, 2025**

Related U.S. Application Data

(63) Continuation of application No. 17/861,870, filed on Jul. 11, 2022, now Pat. No. 12,314,679.

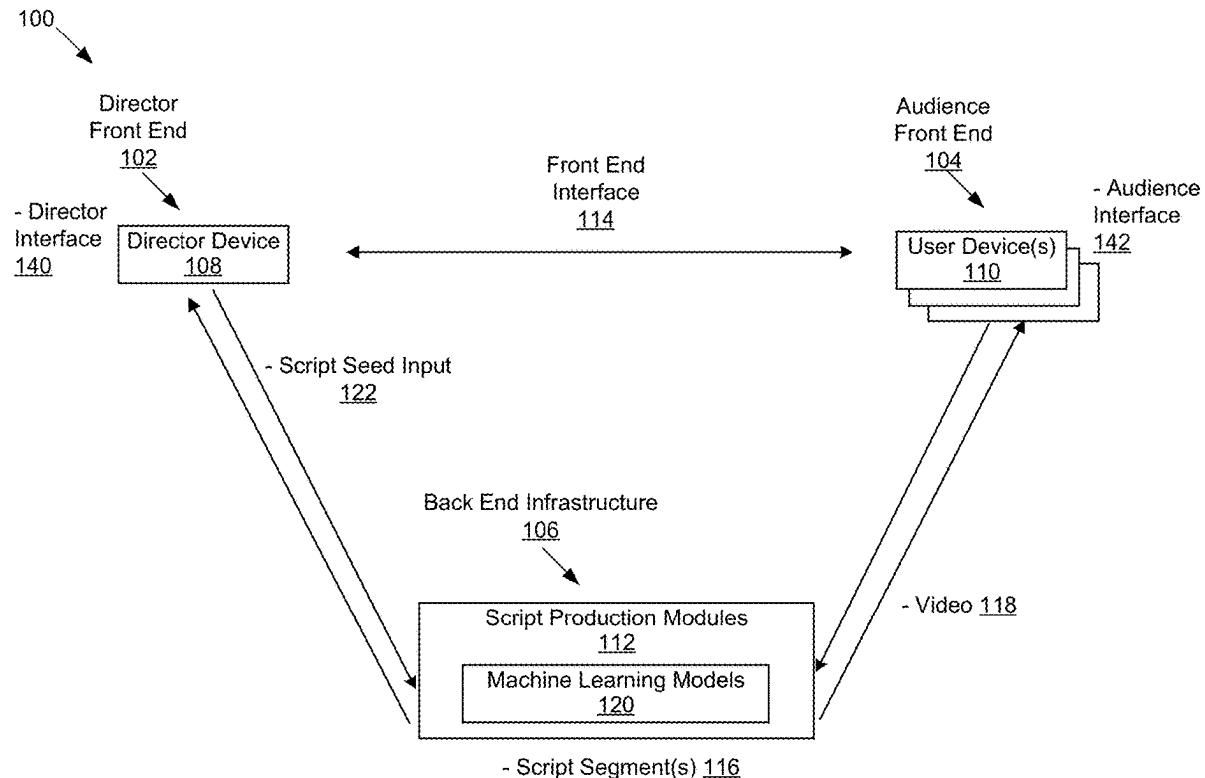
(60) Provisional application No. 63/220,442, filed on Jul. 9, 2021.

Publication Classification

(51) **Int. CL.**

G06F 40/56 (2020.01)

G06F 40/103 (2020.01)



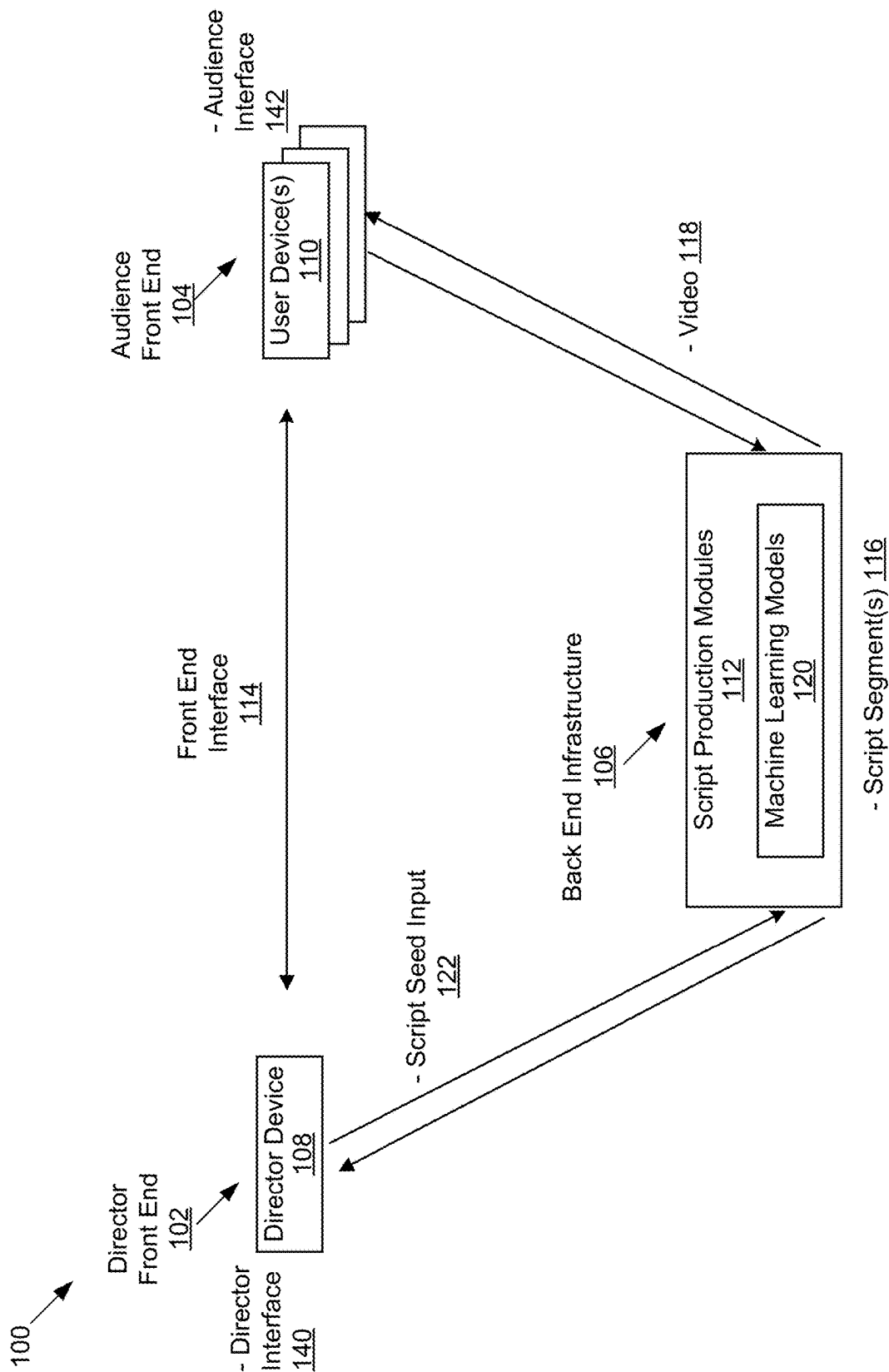


FIG. 1

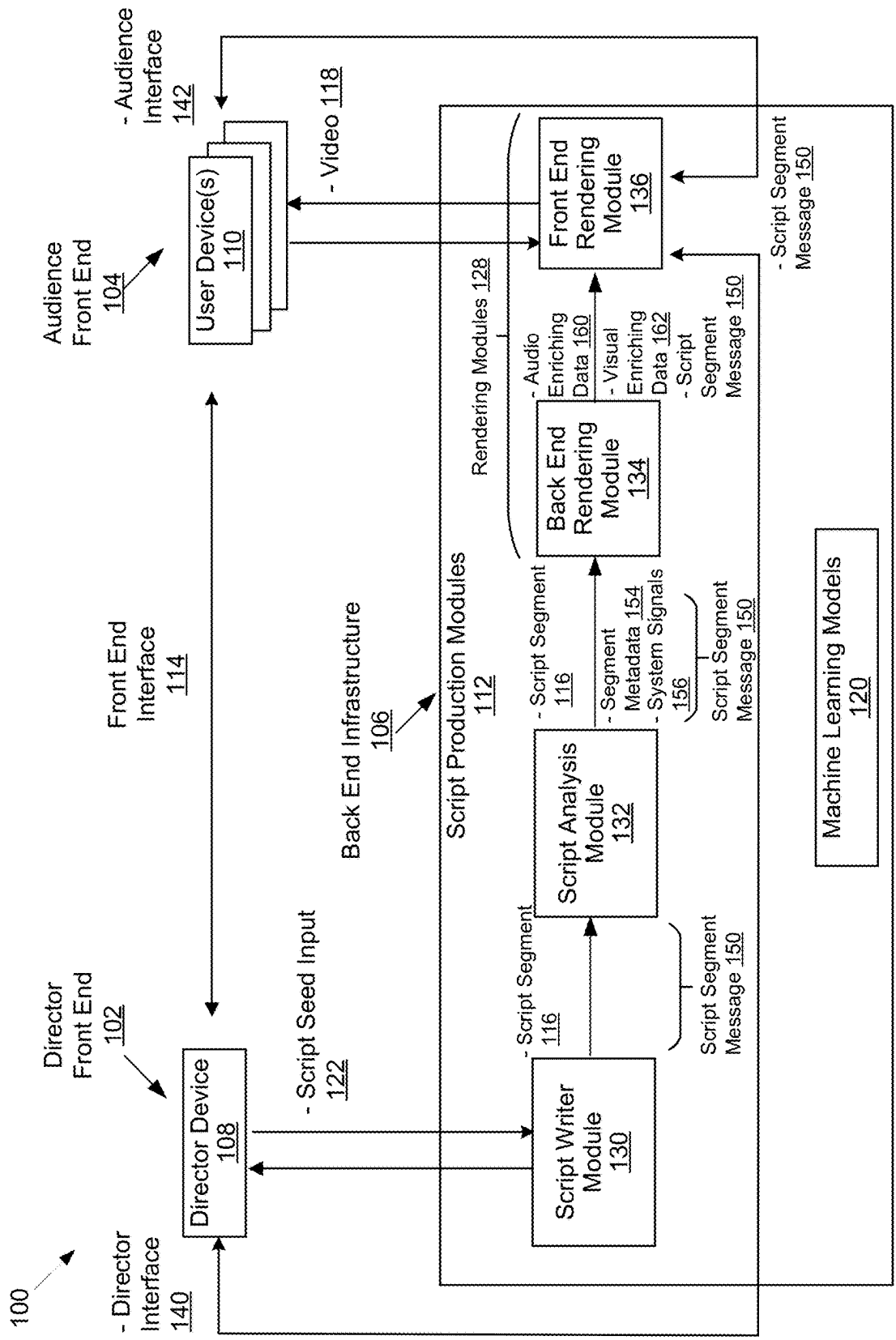


FIG. 2

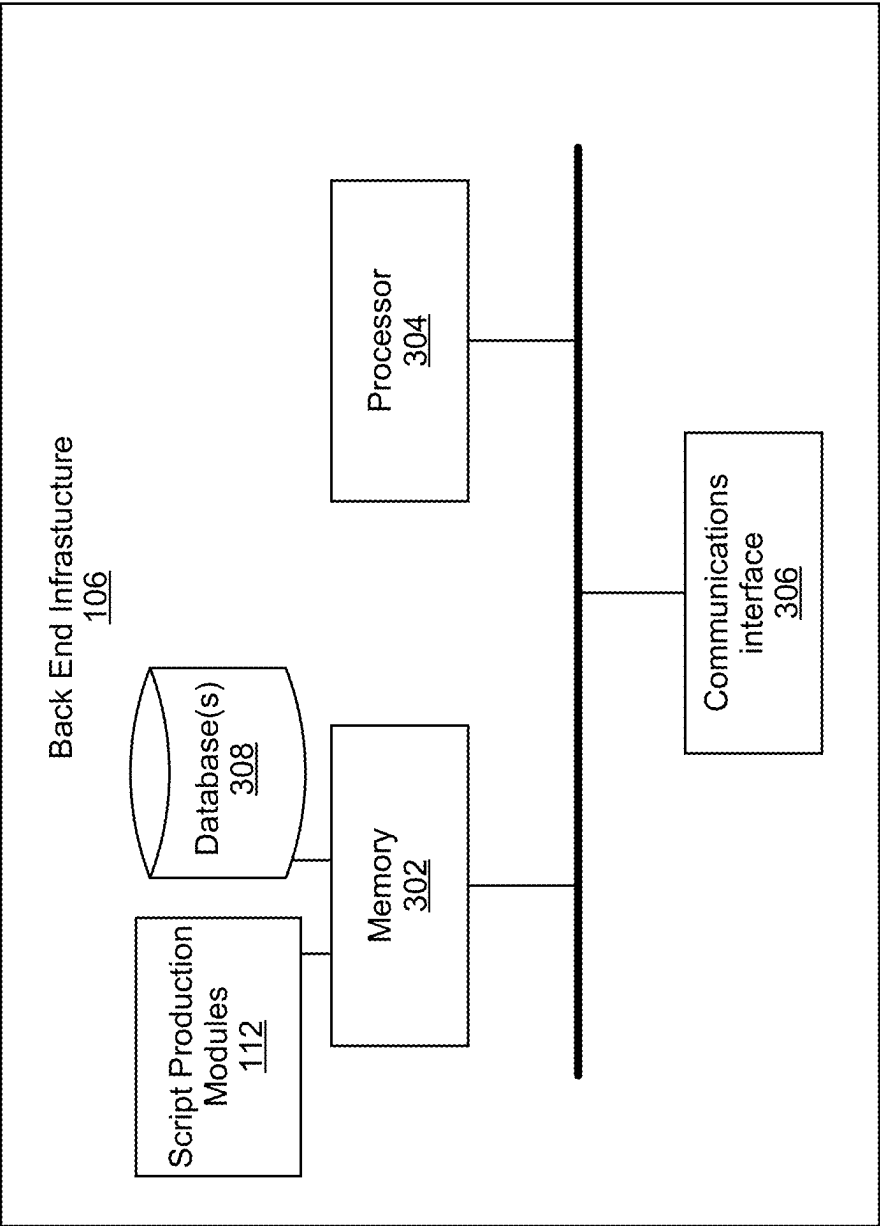


FIG. 3A

106

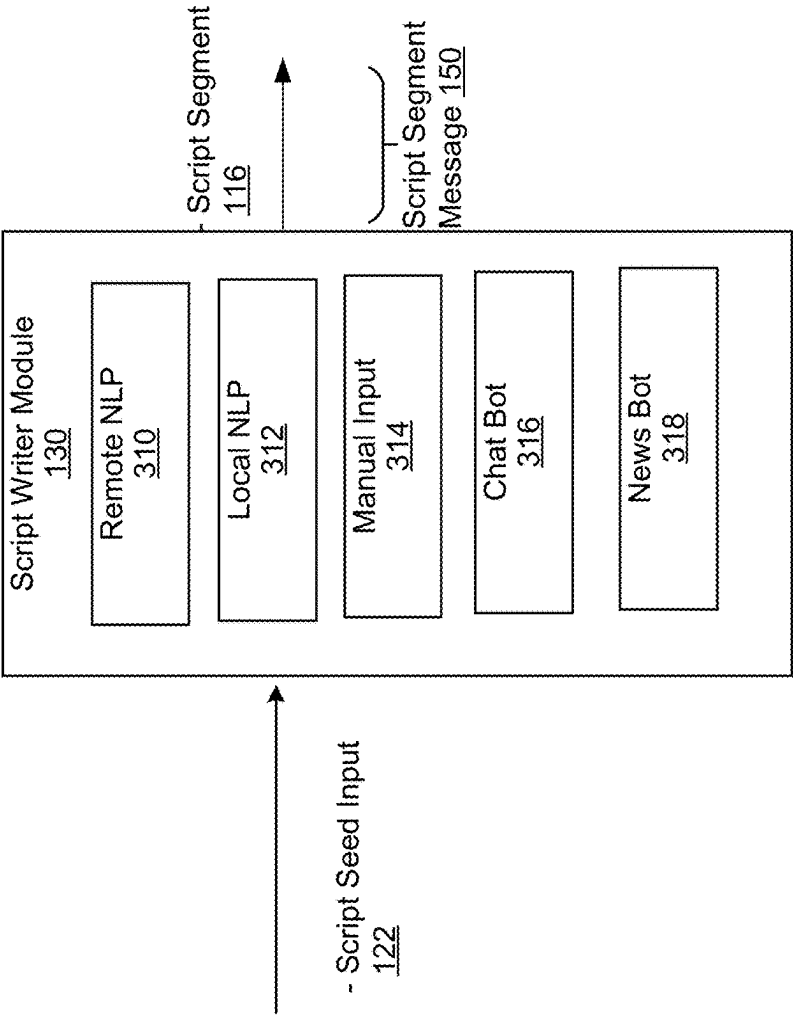


FIG. 3B

106

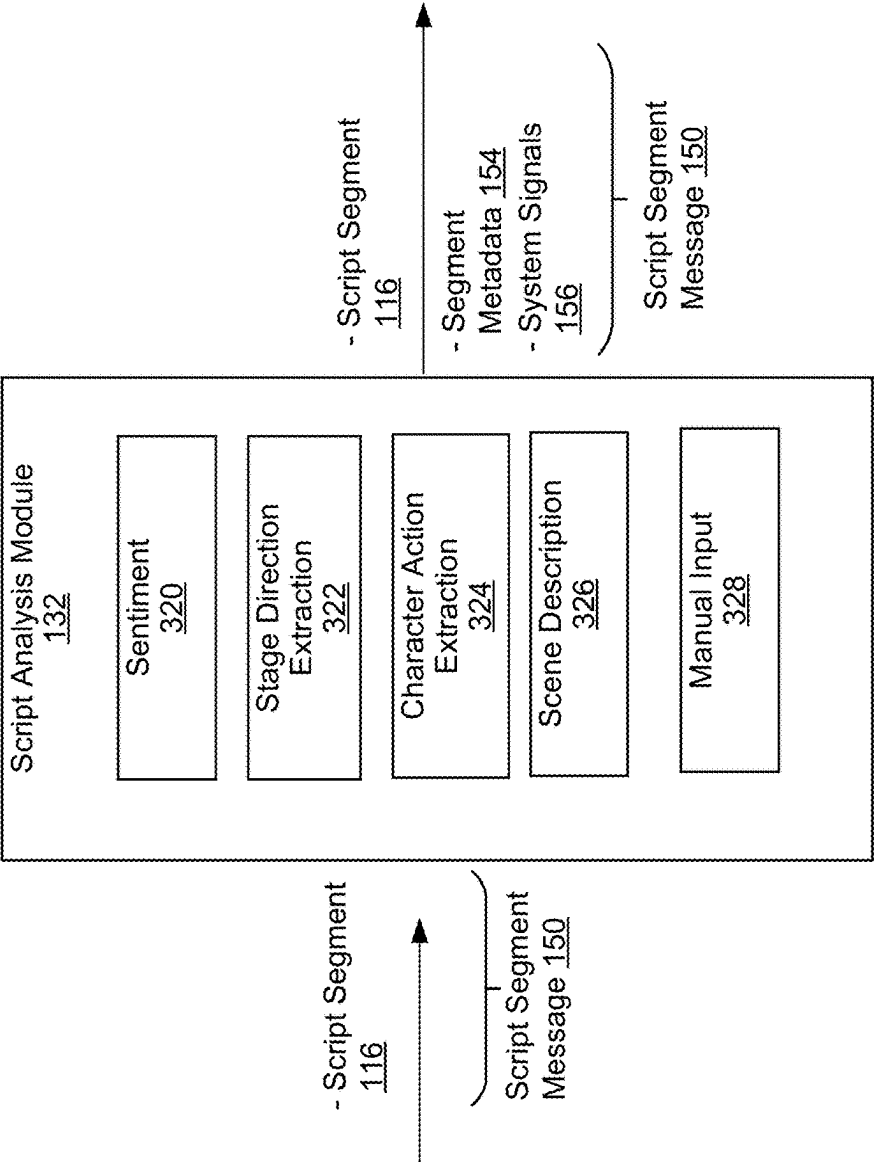


FIG. 3C

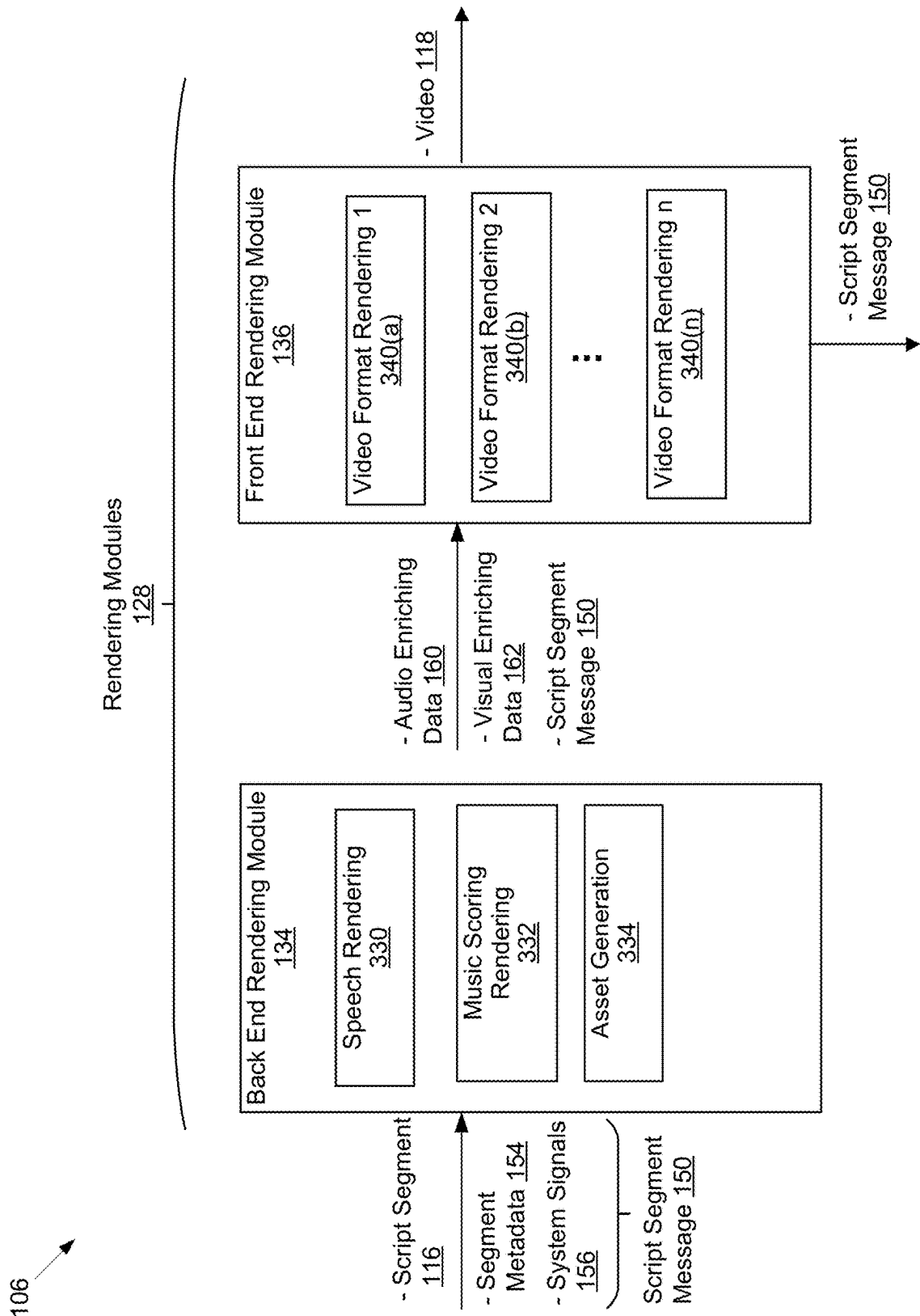


FIG. 3D

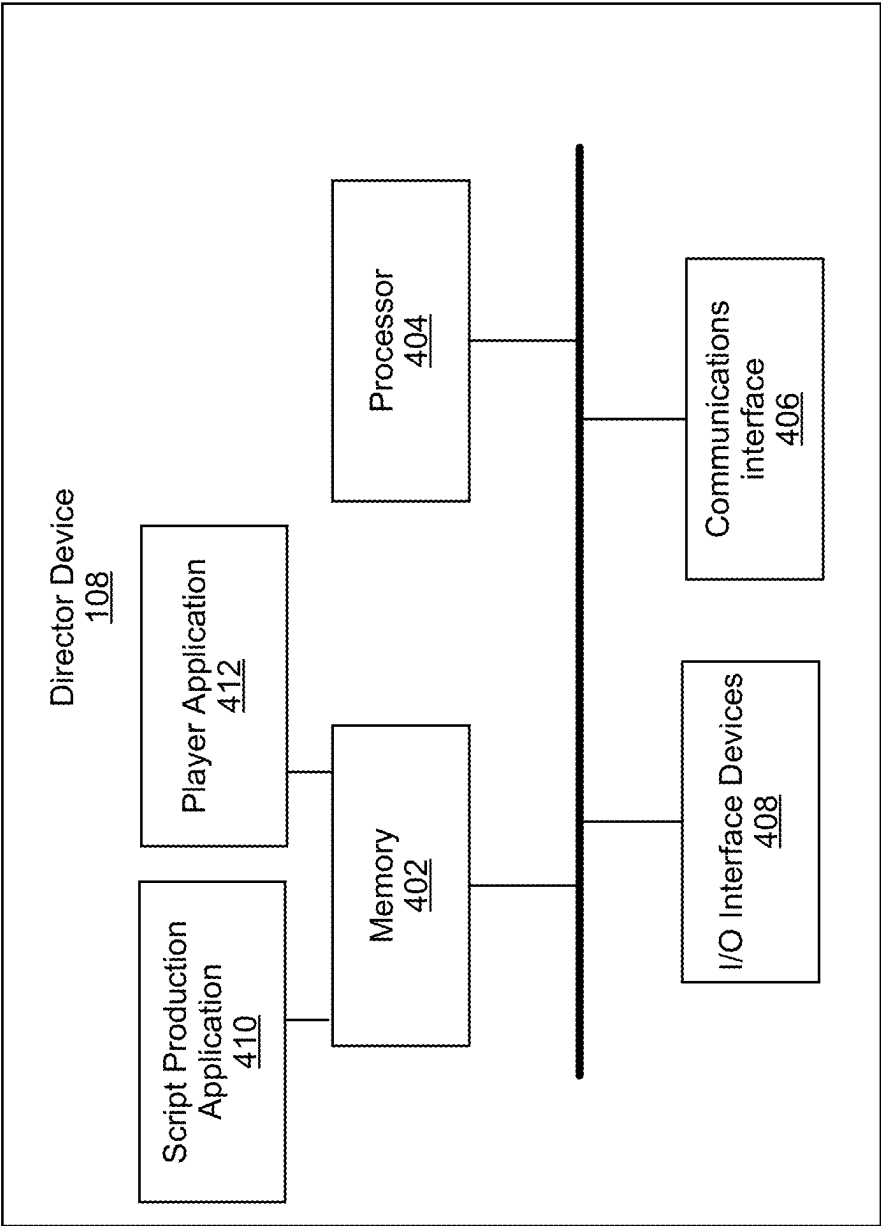


FIG. 4

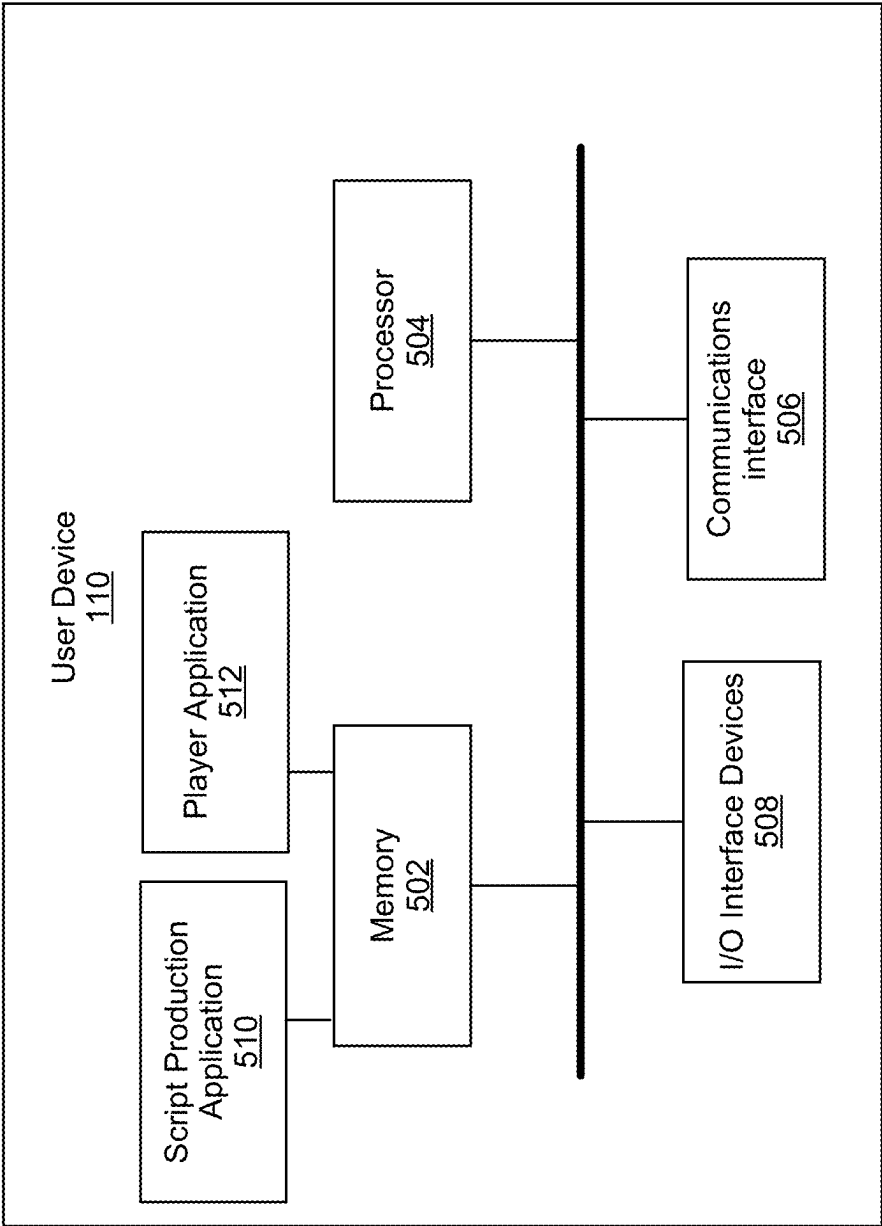


FIG. 5

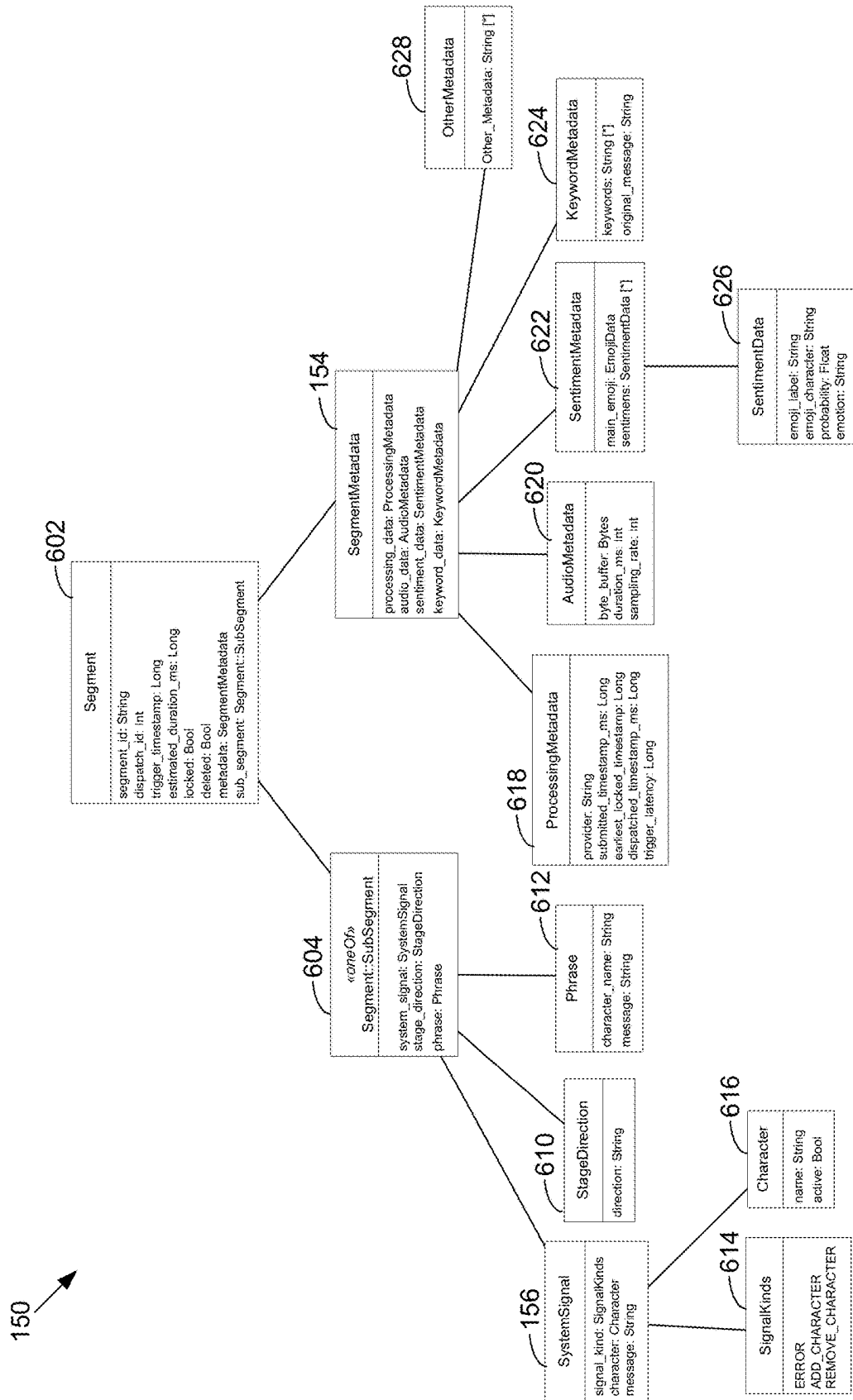


FIG. 6

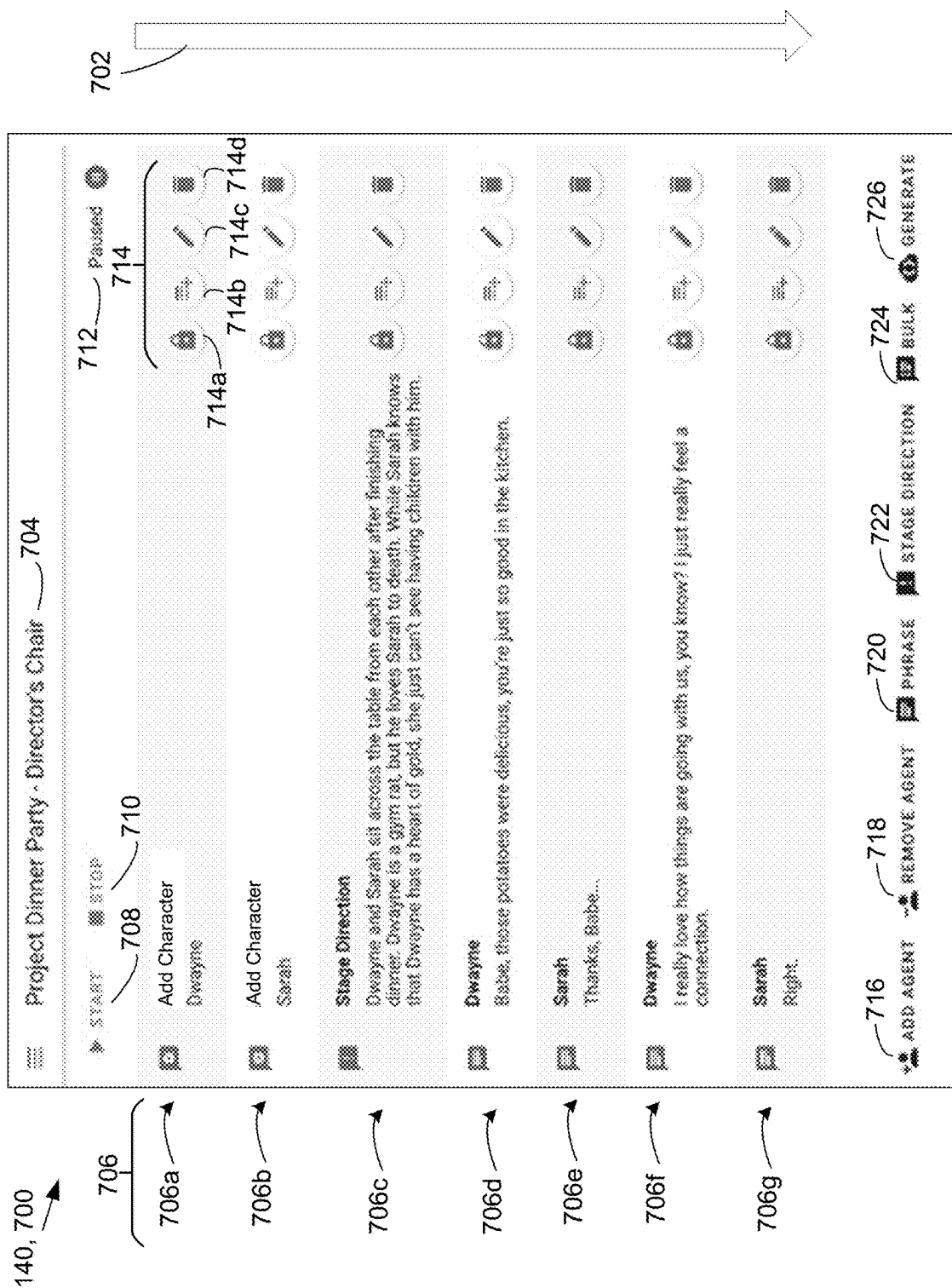


FIG. 7A

140,700

706

Project Dinner Party - Director's Chair

▶ 5/25/21 ⌵ STOP

Being there yesterday was incredible, we're just so glad to see you.

706e

706f

706g

706h

706i

706j

706k

706l

706m

706n

706o

706p

706q

706r

706s

706t

706u

706v

706w

706x

706y

706z

706aa

706ab

706ac

706ad

706ae

706af

706ag

706ah

706ai

706aj

706ak

706al

706am

706an

706ao

706ap

706aq

706ar

706as

706at

706au

706av

706aw

706ax

706ay

706az

706ba

706bb

706bc

706bd

706be

706bf

706bg

706bh

706bi

706bj

706bk

706bl

706bm

706bn

706bo

706bp

706bq

706br

706bs

706bt

706bu

706bv

706bw

706bx

706by

706bz

706ca

706cb

706cc

706cd

706ce

706cf

706cg

706ch

706ci

706cj

706ck

706cl

706cm

706cn

706co

706cp

706cq

706cr

706cs

706ct

706cu

706cv

706cw

706cx

706cy

706cz

706da

706db

706dc

706dd

706de

706df

706dg

706dh

706di

706dj

706dk

706dl

706dm

706dn

706do

706dp

706dq

706dr

706ds

706dt

706du

706dv

706dw

706dx

706dy

706dz

706ea

706eb

706ec

706ed

706ee

706ef

706eg

706eh

706ei

706ej

706ek

706el

706em

706en

706eo

706ep

706eq

706er

706es

706et

706eu

706ev

706ew

706ex

706ey

706ez

706fa

706fb

706fc

706fd

706fe

706ff

706fg

706fh

706fi

706fj

706fk

706fl

706fm

706fn

706fo

706fp

706fq

706fr

706fs

706ft

706fu

706fv

706fw

706fx

706fy

706fz

706ga

706gb

706gc

706gd

706ge

706gf

706gg

706gh

706gi

706gj

706gk

706gl

706gm

706gn

706go

706gp

706gq

706gr

706gs

706gt

706gu

706gv

706gw

706gx

706gy

706gz

706ha

706hb

706hc

706hd

706he

706hf

706hg

706hh

706hi

706hj

706hk

706hl

706hm

706hn

706ho

706hp

706hq

706hr

706hs

706ht

706hu

706hv

706hw

706hx

706hy

706hz

706ia

706ib

706ic

706id

706ie

706if

706ig

706ih

706ii

706ij

706ik

706il

706im

706in

706io

706ip

706iq

706ir

706is

706it

706iu

706iv

706iw

706ix

706iy

706iz

706ja

706jb

706jc

706jd

706je

706jf

706jg

706jh

706ji

706jj

706jk

706jl

706jm

706jn

706jo

706jp

706jq

706jr

706js

706jt

706ju

706jv

706jw

706jx

706jy

706jz

706ka

706kb

706kc

706kd

706ke

706kf

706kg

706kh

706ki

706kj

706kk

706kl

706km

706kn

706ko

706kp

706kq

706kr

706ks

706kt

706ku

706kv

706kw

706kx

706ky

706kz

706la

706lb

706lc

706ld

706le

706lf

706lg

706lh

706li

706lj

706lk

706ll

706lm

706ln

706lo

706lp

706lq

706lr

706ls

706lt

706lu

706lv

706lw

706lx

706ly

706lz

706ma

706mb

706mc

706md

706me

706mf

706mg

706mh

706mi

706mj

706mk

706ml

706mm

706mn

706mo

706mp

706mq

706mr

706ms

706mt

706mu

706mv

706mw

706mx

706my

706mz

706na

706nb

706nc

706nd

706ne

706nf

706ng

706nh

706ni

706nj

706nk

706nl

706nm

706nn

706no

706np

706nq

706nr

706ns

706nt

706nu

706nv

706nw

706nx

706ny

706nz

706oa

706ob

706oc

706od

706oe

706of

706og

706oh

706oi

706oj

706ok

706ol

706om

706on

706oo

706op

706oq

706or

706os

706ot

706ou

706ov

706ow

706ox

706oy

706oz

706pa

706pb

706pc

706pd

706pe

706pf

706pg

706ph

706pi

706pj

706pk

706pl

706pm

706pn

706po

706pp

706pq

706pr

706ps

706pt

706pu

706pv

706pw

706px

706py

706pz

706qa

706qb

706qc

706qd

706qe

706qf

706qg

706qh

706qi

706qj

706qk

706ql

706qm

706qn

706qo

706qp

706qq

706qr

706qs

706qt

706qu

706qv

706qw

706qx

706qy

706qz

706ra

706rb

706rc

706rd

706re

706rf

706rg

706rh

706ri

706rj

706rk

706rl

706rm

706rn

706ro

706rp

706rq

706rr

706rs

706rt

706ru

706rv

706rw

706rx

706ry

706rz

706sa

706sb

706sc

706sd

706se

706sf

706sg

706sh

706si

706sj

706sk

706sl

706sm

706sn

706so

706sp

706sq

706sr

706ss

706st

706su

706sv

706sw

706sx

706sy

706sz

706ta

706tb

706tc

706td

706te

706tf

706tg

706th

706ti

706tj

706tk

706tl

706tm

706tn

706to

706tp

706tq

706tr

706ts

706tt

706tu

706tv

706tw

706tx

706ty

706tz

706ua

706ub

706uc

706ud

706ue

706uf

706ug

706uh

706ui

706uj

706uk

706ul

706um

706un

706uo

706up

706uq

706ur

706us

706ut

706uu

706uv

706uw

706ux

706uy

706uz

706va

706vb

706vc

706vd

706ve

706vf

706vg

706vh

706vi

706vj

706vk

706vl

706vm

706vn

706vo

706vp

706vq

706vr

706vs

706vt

706vu

706vv

706vw

706vx

706vy

706vz

706wa

706wb

706wc

706wd

706we

706wf

706wg

706wh

706wi

706wj

706wk

706wl

706wm

706wn

706wo

706wp

706wq

706wr

706ws

706wt

706wu

706wv

706ww

706wx

706wy

706wz

706xa

706xb

706xc

706xd

706xe

706xf

706xg

706xh

706xi

706xj

706xk

706xl

706xm

706xn

706xo

706xp

706xq

706xr

706xs

706xt

706xu

706xv

706xw

706xx

706xy

706xz

706ya

706yb

706yc

706yd

706ye

706yf

706yg

706yh

706yi

706yj

706yk

706yl

706ym

706yn

706yo

706yp

706yq

706yr

706ys

706yt

706yu

706yv

706yw

706yx

706yy

706yz

706za

706zb

706zc

706zd

706ze

706zf

706zg

706zh

706zi

706zj

706zk

706zl

706zm

706zn

706zo

706zp

706zq

706zr

706zs

706zt

706zu

706zv

706zw

706zx

706zy

706zz

730

Director

I know I was just there yesterday, but now that I have a bigger room, everything is different.

730a

730b

730c

730d

730e

730f

730g

730h

730i

730j

730k

730l

730m

730n

730o

730p

730q

730r

730s

730t

730u

730v

730w

730x

730y

730z

730aa

730ab

730ac

730ad

730ae

730af

730ag

730ah

730ai

730aj

730ak

730al

730am

730an

730ao

730ap

730aq

730ar

730as

730at

730au

730av

730aw

730ax

730ay

730az

730ba

730bb

730bc

730bd

730be

730bf

730bg

730bh

730bi

730bj

730bk

730bl

730bm

730bn

730bo

730bp

730bq

730br

730bs

730bt

730bu

730bv

730bw

730bx

730by

730bz

730ca

730cb

730cc

730cd

730ce

730cf

730cg

730ch

730ci

730cj

730ck

730cl

730cm

730cn

730co

730cp

730cq

730cr

730cs

730ct

730cu

730cv

730cw

730cx

730cy

730cz

730da

730db

730dc

730dd

730de

730df

730dg

730dh

730di

730dj

730dk

730dl

730dm

730dn

730do

730dp

730dq

730dr

730ds

730dt

730du

730dv

730dw

730dx

730dy

730dz

730ea

730eb

730ec

730ed

730ee

730ef

730eg

730eh

730ei

730ej

730ek

730el

730em

730en

730eo

730ep

730eq

730er

730es

730et

730eu

730ev

730ew

730ex

730ey

730ez

730fa

730fb

730fc

730fd

730fe

730ff

730fg

730fh

730fi

730fj

730fk

730fl

730fm

730fn

730fo

730fp

730fq

730fr

730fs

730ft

730fu

730fv

730fw

730fx

730fy

730fz

730ga

730gb

730gc

730gd

730ge

730gf

730gg

730gh

730gi

730gj

730gk

730gl

730gm

730gn

730go

730gp

730gq

730gr

730gs

730gt

730gu

730gv

730gw

730gx

730gy

730gz

730ha

730hb

730hc

730hd

730he

730hf

730hg

730hh

730hi

730hj

730hk

730hl

730hm

730hn

730ho

730hp

730hq

730hr

730hs

730ht

730hu

730hv

730hw

730hx

730hy

730hz

730ia

730ib

730ic

730id

730ie

730if

730ig

730ih

730ii

730ij

730ik

730il

730im

730in

730io

730ip

730iq

730ir

730is

730it

730iu

730iv

730iw

730ix

730iy

730iz

730ja

730jb

730jc

730jd

730je

730jf

730jg

730jh

730ji

730jj

730jk

730jl

730jm

730jn

730jo

730jp

730jq

730jr

730js

730jt

730ju

730jv

730jw

730jx

730jy

730jz

730ka

730kb

730kc

730kd

730ke

730kf

730kg

730kh

730ki

730kj

730kk

730kl

730km

730kn

730ko

730kp

730kq

730kr

730ks

730kt

730ku

730kv

730kw

730kx

730ky

730kz

730la

730lb

730lc

730ld

730le

730lf

730lg

730lh

730li

730lj

730lk

730ll

730lm

730ln

730lo

730lp

730lq

730lr

730ls

730lt

730lu

730lv

730lw

730lx

730ly

730lz

730ma

730mb

730mc

730md

730me

730mf

730mg

730mh

730mi

730mj

730mk

730ml

730mm

730mn

730mo

730mp

730mq

730mr

730ms

730mt

730mu

730mv

730mw

730mx

730my

730mz

730na

730nb

730nc

730nd

730ne

730nf

730ng

730nh

730ni

730nj

730nk

730nl

730nm

730nn

730no

730np

730nq

730nr

730ns

730nt

730nu

730nv

730nw

730nx

730ny

730nz

730oa

730ob

730oc

730od

730oe

730of

730og

730oh

730oi

730oj

730ok

730ol

730om

730on

730oo

730op

730oq

730or

730os

730ot

730ou

730ov

730ow

730ox

730oy

730oz

730pa

730pb

730pc

730pd

730pe

730pf

730pg

730ph

730pi

730pj

730pk

730pl

730pm

730pn

730po

730pp

730pq

730pr

730ps

730pt

730pu

730pv

730pw

730px

730py

730pz

730qa

730qb

730qc

730qd

730qe

730qf

730qg

730qh

730qi

730qj

730qk

730ql

730qm

730qn

730qo

730qp

730qq

730qr

730qs

730qt

730qu

730qv

730qw

730qx

730qy

730qz

730ra

730rb

730rc

730rd

730re

730rf

730rg

730rh

730ri

730rj

730rk

730rl

730rm

730rn

730ro

730rp

730rq

730rr

730rs

730rt

730ru

730rv

730rw

730rx

730ry

730rz

730sa

730sb

730sc

730sd

730se

730sf

730sg

730sh

730si

730sj

730sk

730sl

730sm

730sn

730so

730sp

730sq

730sr

730ss

730st

730su

730sv

730sw

730sx

730sy

730sz

730ta

730tb

730tc

730td

730te

730tf

730tg

730th

730ti

730tj

730tk

730tl

730tm

730tn

730to

730tp

730tq

730tr

730ts

730tt

730tu

730tv

730tw

730tx

730ty

730tz

730ua

730ub

730uc

730ud

730ue

730uf

730ug

730uh

730ui

730uj

730uk

730ul

730um

730un

730uo

730up

730uq

730ur

730us

730ut

730uu

730uv

730uw

730ux

730uy

730uz

730va

730vb

730vc

730vd

730ve

730vf

730vg

730vh

730vi

730vj

730vk

730vl

730vm

730vn

730vo

730vp

730vq

730vr

730vs

730vt

730vu

730vv

730vw

730vx

730vy

730vz

730wa

730wb

730wc

730wd

730we

730wf

730wg

730wh

730wi

730wj

730wk

730wl

730wm

730wn

730wo

730wp

730wq

730wr

730ws

730wt

730wu

730wv

730ww

730wx

730wy

730wz

730xa

730xb

730xc

730xd

730xe

730xf

730xg

730xh

730xi

730xj

730xk

730xl

730xm

730xn

730xo

730xp

730xq

730xr

730xs

730xt

730

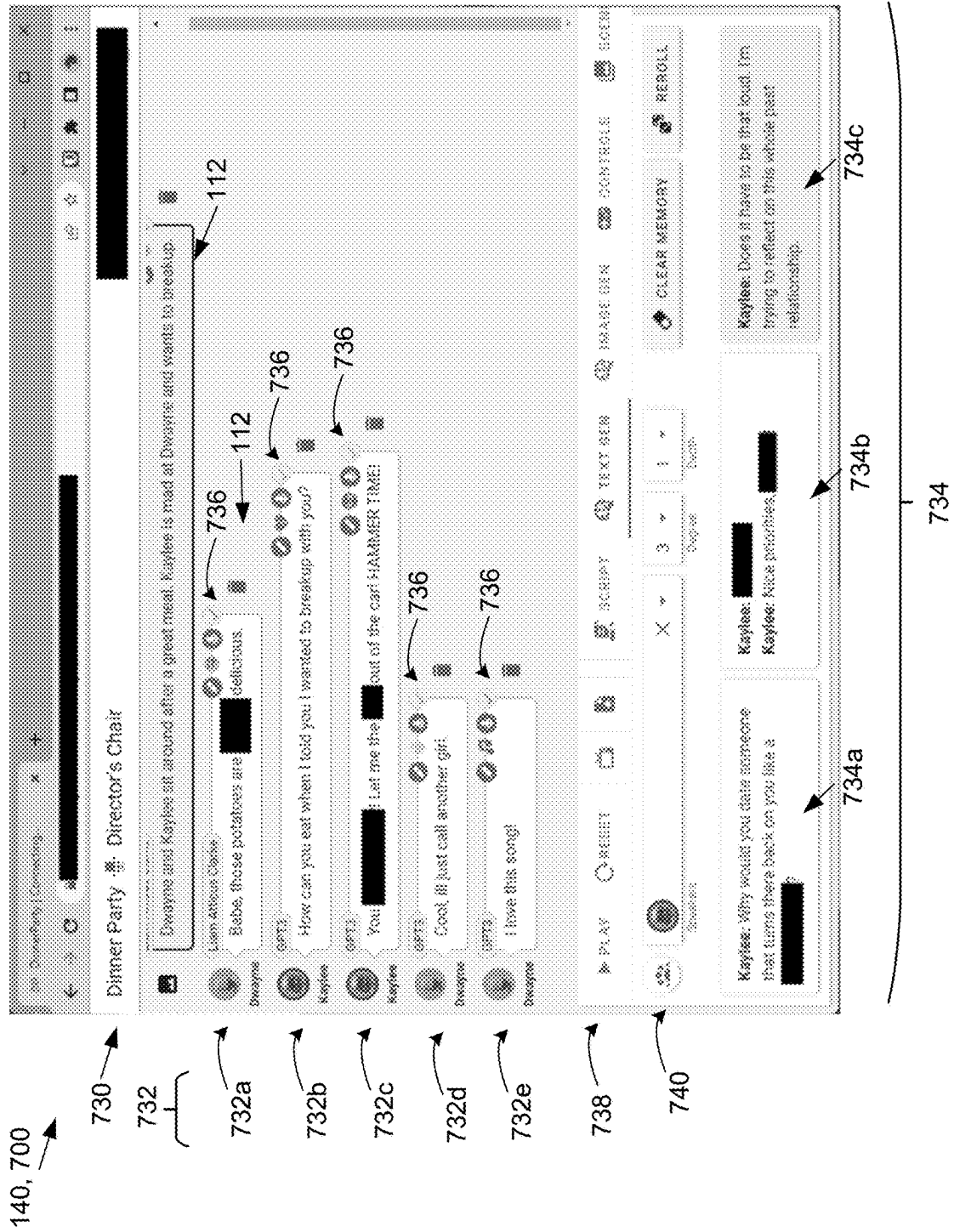


FIG. 7C

118, 800



806

FIG. 8

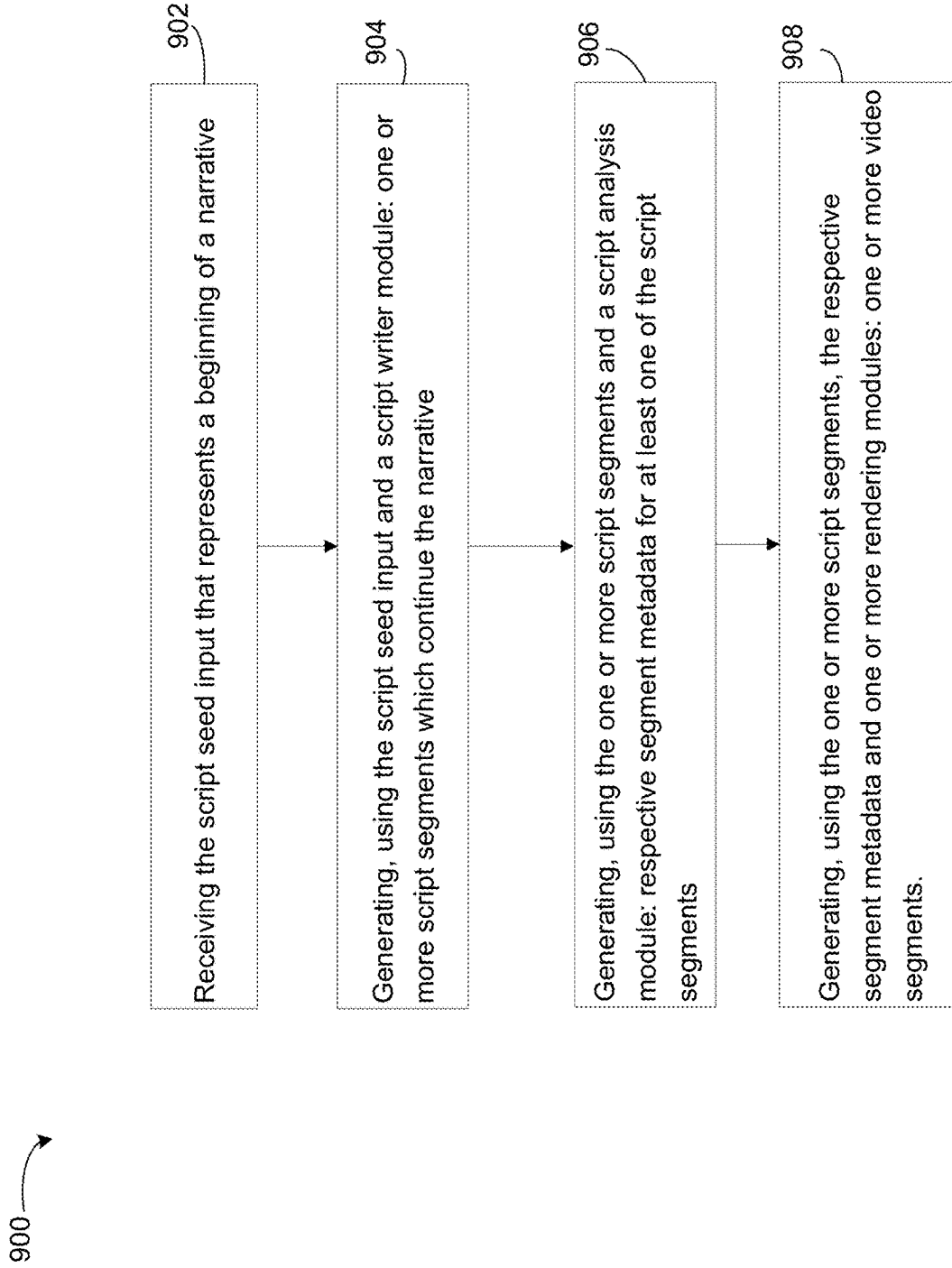


FIG. 9A

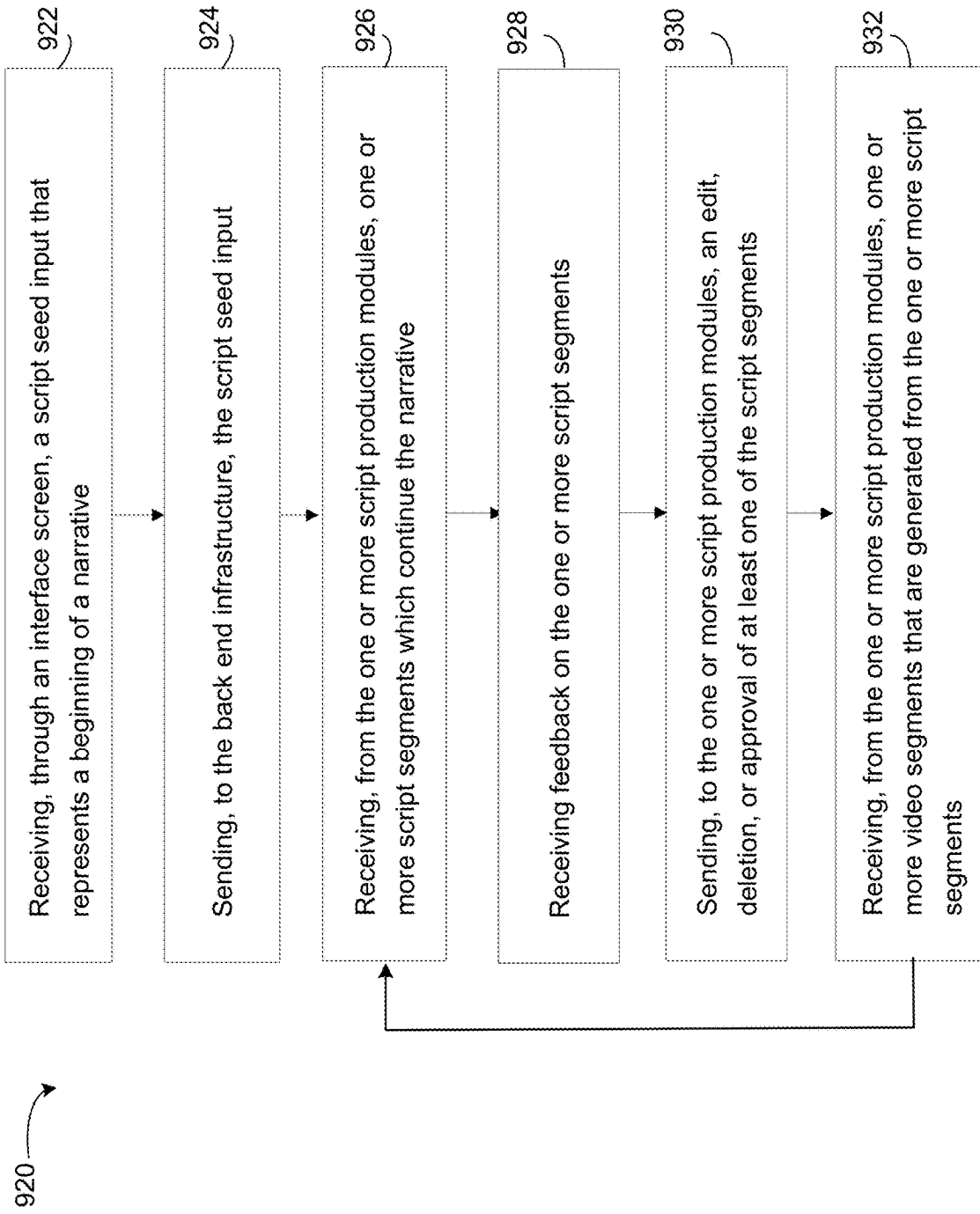


FIG. 9B

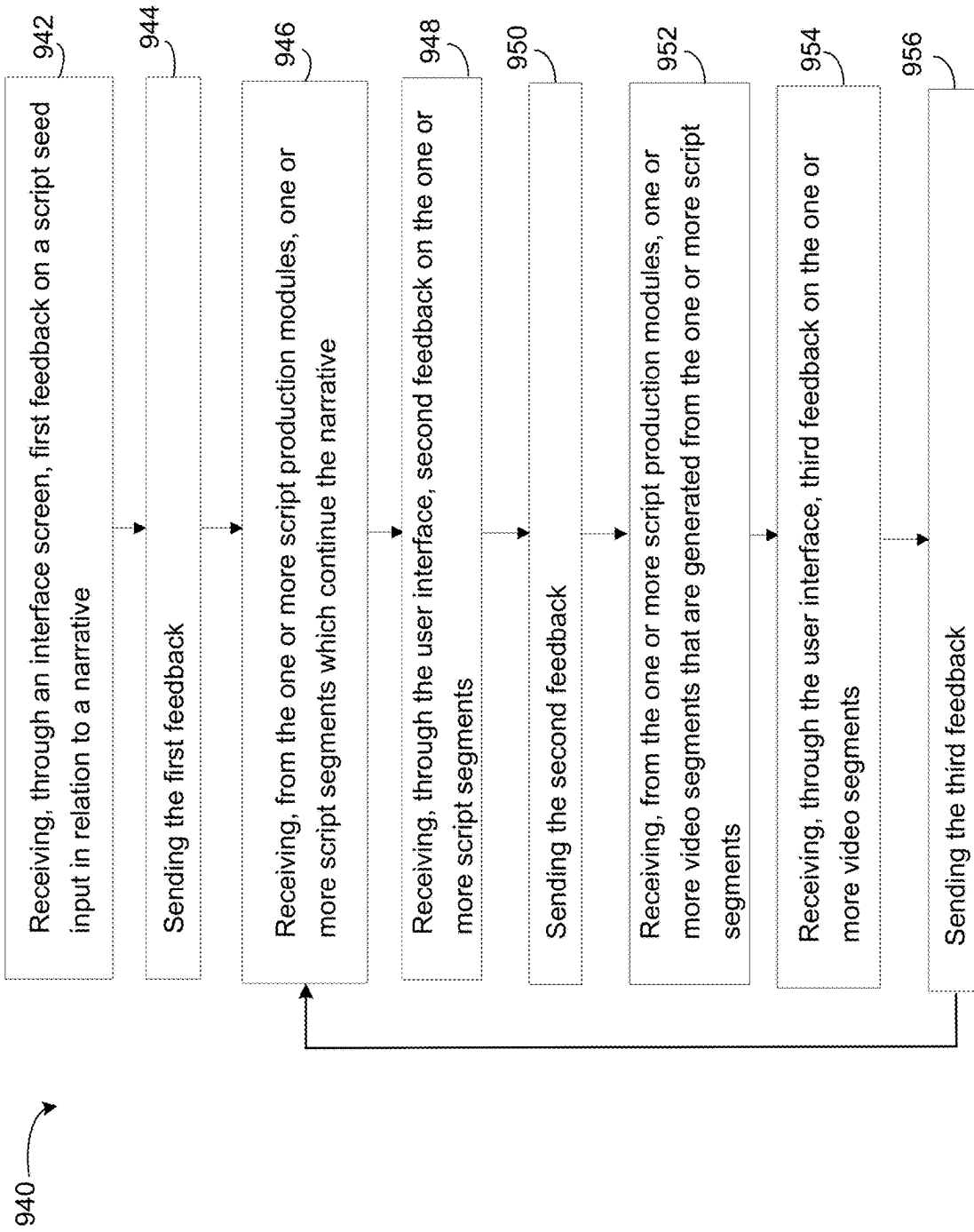


FIG. 9C

**AUTOMATED NARRATIVE PRODUCTION
SYSTEM AND SCRIPT PRODUCTION
METHOD WITH REAL-TIME INTERACTIVE
CHARACTERS**

CROSS-REFERENCE

[0001] This application is a continuation application of U.S. patent application Ser. No. 17/861,870 filed Jul. 11, 2022 entitled AUTOMATED NARRATIVE PRODUCTION SYSTEM AND SCRIPT PRODUCTION METHOD WITH REAL-TIME INTERACTIVE CHARACTERS, which claims the benefit of U.S. Provisional Patent Application No. 63/220,442 filed Jul. 9, 2021, the entire contents of all of these references are herein incorporated by reference.

TECHNICAL FIELD

[0002] Example embodiments relate to automated generation of real-time scripts and animation videos.

BACKGROUND

[0003] Video shows such as movies, television shows, online shows, narratives and animations can require large production studios. The production resources for a video show can include scriptwriters, directors, actors, crew, equipment, location sites, post-processing, and large time and financial resources. For animated video shows, high performance computers and skilled animators are also typically needed.

[0004] After a video show is produced, there may be reviews and audience feedback. In some instances, the video show cannot be easily changed after public release. In other instances, for example when a focus group is used, the same level of production resources may be required to adjust, rewrite and remake particular scenes of the video show. These changes need to be manually performed after receiving the reviews and audience feedback, which is slow and not in real-time. Conventional video production methods are therefore slow to respond to feedback from audiences.

[0005] Additional difficulties may be appreciated in view of the Detailed Description, herein below.

SUMMARY

[0006] It would be advantageous to have a show production system that can generate scripts and video shows in real-time.

[0007] It would be advantageous to have the show production system be dynamically adjustable based on director feedback, audience feedback, and machine learning models.

[0008] It would be advantageous to have the show production system that tailors content to the user's (or users') current state of engagement.

[0009] It would be advantageous to have the show production system that optimizes engagement over time and in real time.

[0010] It would be advantageous to have the show production system be dynamically adjustable based on a short feedback loop from user feedback.

[0011] Example embodiments relate to show production systems and script production methods for generating scripts and video shows in real-time. An example of the script production method can be applied to the scenario where a user has a script seed input that the user wishes to be the

starting point for a video show. In response, the script production method uses one or more script production modules to dynamically generate and propose, in real-time, new script segments for a new script from the script seed input. The new script can include the same or different characters having phrases and stage directions that stem from the script seed input. The script production method uses the one or more script production modules to dynamically generate in real-time a new video from the script. The incremental nature of the script generation enables audiences and directors to influence the outcomes of the video show. Other enrichment data, audio enrichment data and visual enrichment data can be generated by the script production modules and incorporated into the new video.

[0012] In some example embodiments, the new video is streamed (live broadcast or live published) to the audience prior to completion of the script. For example, as each script is generated, those scripts can be generated into videos and streamed to the audience. Real-time feedback from the audience can be used for the script production method to generate new scripts. The script production method can, in real-time, propose potential new scripts. In some examples, the user can approve, edit or delete prior to the video for those scripts being generated and streamed. In some examples, the user can provide feedback on the video in real-time which is used to generate future proposed scripts that are used by the script production method to continue the generation of the video. The script production method can also generate multiple potential script segments (different narrative branches that continue the narrative) that can be selected, approved, edited or deleted by one or more of the users.

[0013] An example embodiment is a script production method, comprising: receiving a script seed input that represents a beginning of a narrative; generating, using the script seed input and a script writer module: a queue of one or more script segments of a script which continues the narrative; locking at least one of the script segments in the queue; generating, using one or more of the script segments and a script analysis module: respective script metadata for the one or more script segments of the script; generating, using the at least one of the script segments that are locked, the respective script metadata and one or more rendering modules: one or more video segments; and receiving feedback on one or more of the script segments in the queue that have not been locked.

[0014] In an example embodiment of any of the above script production methods, the script seed input includes: i) a phrase and a character to speak the phrase; and ii) a stage direction.

[0015] In an example embodiment of any of the above script production methods, the script seed input includes: i) a second phrase and a second character to speak the second phrase.

[0016] In an example embodiment of any of the above script production methods, the script seed input includes: a system signal.

[0017] In an example embodiment of any of the above script production methods, the system signal includes: adding a character, removing a character, an error signal, a scene change, an animation trigger, a sound effect trigger, ending a scene, a lighting cue, a camera cues, or ending a show.

[0018] In an example embodiment of any of the above script production methods, the script seed input includes: an input text, an input image, an input video, or an input audio of a spoken narrative.

[0019] In an example embodiment of any of the above script production methods, at least one of the script segments includes: a phrase and a character speaking the phrase; and wherein at least one of the script segments includes: a stage direction.

[0020] In an example embodiment of any of the above script production methods, at least one of the script segments includes: a system signal.

[0021] In an example embodiment of any of the above script production methods, the system signal includes: adding a character, removing a character, an error signal, a scene change, an animation trigger, a sound effect trigger, ending a scene, a lighting cue, a camera cues, or ending a show.

[0022] In an example embodiment of any of the above script production methods, the one or more script segments are each in a common script segment message format.

[0023] In an example embodiment of any of the above script production methods, the script writer module includes: a remote natural language processing (NLP) module; a local NLP module; a chat bot module; and/or a news bot module.

[0024] In an example embodiment of any of the above script production methods, the script analysis module includes: a sentiment module for generating sentiment metadata for the respective script metadata; a stage direction extraction module for generating a stage direction or stage direction metadata of the stage direction for the respective script metadata; a character extraction module for generating character metadata of a character action for the respective script metadata; and/or a scene direction module for generating scene direction metadata for the respective script metadata.

[0025] In an example embodiment of any of the above script production methods, the generating the respective script metadata includes generating processing metadata using the script and the script analysis module, wherein the processing metadata is for technical introspection of one or more states of the script production method, managing performance of the script production method, and managing timing of message delivery between the script writer module, the script analysis module, and the one or more rendering modules.

[0026] In an example embodiment of any of the above script production methods, the generating of the respective script metadata includes generating audio metadata using the script and the script analysis module, wherein the audio metadata includes audio duration and audio sampling rate, wherein the generating the one or more video segments uses the audio metadata.

[0027] In an example embodiment of any of the above script production methods, at least one of the rendering modules includes a speech rendering module which uses the script and the audio metadata for generating audio enriching data of speech for the generating the one or more video segments.

[0028] In an example embodiment of any of the above script production methods, at least one of the rendering modules includes a music scoring module which uses the

script and the audio metadata for generating audio enriching data of music for the generating the one or more video segments.

[0029] In an example embodiment of any of the above script production methods, the generating the respective script metadata includes generating sentiment metadata using the script and the script analysis module, wherein the sentiment metadata includes sentiment data or emotion data, wherein the generating the one or more video segments uses the sentiment metadata.

[0030] In an example embodiment of any of the above script production methods, the generating the respective script metadata includes generating keyword metadata using the script and the script analysis module, wherein the keyword metadata includes at least one keyword extracted from a phrase of the script, wherein the at least one keyword is given higher weight by the script analysis module or one of the rendering modules, wherein the generating the one or more video segments uses the keyword metadata.

[0031] In an example embodiment of any of the above script production methods, the queue is generated in real-time as the at least one of the script segments are generated into the one or more video segments.

[0032] In an example embodiment of any of the above script production methods, the feedback includes an approval, an edit, or a deletion of at least one of the script segments in the queue.

[0033] In an example embodiment of any of the above script production methods, the feedback includes audience feedback from at least one device, and wherein the feedback includes director feedback from at least one of the devices.

[0034] In an example embodiment of any of the above script production methods, the generating the one or more video segments is generated from three way authorship of the script writer module, the director feedback, and the audience feedback.

[0035] In an example embodiment of any of the above script production methods, the method further includes storing the feedback, wherein the script writer module is configured to learn from the feedback.

[0036] In an example embodiment of any of the above script production methods, the one or more rendering modules include a back end rendering module and a front end rendering module, wherein the generating the one or more video segments includes: generating, using the script, the respective script metadata, and the back end rendering module: audio enriching data and visual enriching data; generating, using the audio enriching data, the visual enriching data, and the front end rendering module: the one or more video segments; and formatting, using the front end rendering module, the one or more video segments to a video format that is particular to a respective platform of one or more user devices.

[0037] In an example embodiment of any of the above script production methods, the back end rendering module includes an asset generation module.

[0038] In an example embodiment of any of the above script production methods, the asset generation module is configured to generate a character and/or an object in at least one the video segments.

[0039] In an example embodiment of any of the above script production methods, the method further includes receiving second feedback on the script metadata.

[0040] In an example embodiment of any of the above script production methods, the generating the script, the generating the respective script metadata, and the generating the one or more video segments are performed in real-time.

[0041] In an example embodiment of any of the above script production methods, the at least one of the script segments that have been generated into the one or more video segments are live published, wherein the receiving feedback is on one or more of the script segments in the queue that have not been live published.

[0042] In an example embodiment of any of the above script production methods, the at least one of the script segments that are locked are earlier in the queue and the one or more of the script segments in the queue that have not been locked are later in the queue.

[0043] In an example embodiment of any of the above script production methods, the generating the queue includes generating a plurality of different parallel queues each respectively having one or more of the script segments; further comprising receiving a selection of one of the parallel queues.

[0044] In an example embodiment of any of the above script production methods, the receiving feedback is on the queue that is selected.

[0045] In an example embodiment of any of the above script production methods, the selection of the queue is performed by an automated selection system

[0046] In an example embodiment of any of the above script production methods, the automated selection system takes into account the script seed input or further script seed inputs in relation to a middle of the narrative for performing the selection.

[0047] In an example embodiment of any of the above script production methods, the automated selection system includes a machine learning model.

[0048] In an example embodiment of any of the above script production methods, the method further includes receiving one or more further script seed inputs in relation to a middle of the narrative, wherein the generating the queue further uses the further script seed inputs.

[0049] In an example embodiment of any of the above script production methods, the one or more further script seed inputs are manually generated, extracted from a show structure document, or automatically generated by the script writer module.

[0050] In an example embodiment of any of the above script production methods, the one or more further script seed inputs include a natural language description of a character or a scene.

[0051] Another example embodiment is a script production method, comprising: sending, to one or more script production modules, a script seed input that represents a beginning of a narrative; receiving, from the one or more script production modules, a queue of one or more script segments of a script which continue the narrative; receiving a locking of at least one of the script segments in the queue; receiving, from the one or more script production modules, one or more video segments that are generated from the at least one of the script segments that have been locked; and sending, to the one or more script production modules, an edit, a deletion, or an approval of at least one of the script segments in the queue that have not been locked.

[0052] In an example embodiment of any of the above script production methods, the receiving the queue includes

receiving a plurality of different parallel queues each respectively having one or more of the script segments; further comprising receiving a selection of the queue as one of the parallel queues and sending the selection of the queue.

[0053] In an example embodiment of any of the above script production methods, the method further includes sending one or more further script seed inputs in relation to a middle of the narrative, wherein the queue is generated by the one or more script production modules using the further script seed inputs.

[0054] In an example embodiment of any of the above script production methods, the method further includes receiving, from the one or more script production modules, respective script metadata for at least one of the script segments; and sending, to the one or more script production modules, feedback on the respective script metadata for one or more of the script segments in the queue that have not been locked.

[0055] In an example embodiment of any of the above script production methods, the one or more video segments are generated from three way authorship of the one or more script production modules, director feedback, and audience feedback from one or more user devices.

[0056] In an example embodiment of any of the above script production methods, the method further includes displaying an interface screen which includes the one or more script segments in the queue, where the interface screen is configured to receive the edit, the deletion, the approval, or the addition of at least one of the script segments.

[0057] In an example embodiment of any of the above script production methods, the queue is displayed in real-time through the interface screen as further of the script segments are received from the one or more script production modules.

[0058] In an example embodiment of any of the above script production methods, the receiving the one or more video segments of the at least one of the script segments are live published, wherein the sending the edit, the deletion, or the approval is on one or more of the script segments in the queue that have not been live published.

[0059] In an example embodiment of any of the above script production methods, the at least one of the script segments that are locked are earlier in the queue and the one or more of the script segments in the queue that have not been locked are later in the queue.

[0060] An advantage of the script production method is that videos can be automatically dynamically generated in real-time from the script seed input.

[0061] An advantage of the script production method is that costly production studios are not required and production time is reduced.

[0062] An advantage of the script production method is that the videos can be automatically dynamically generated in real-time based on real-time feedback from users.

[0063] An advantage of the script production method is that machine learning can be used to automatically generate videos from the script seed input.

[0064] Another example embodiment is a show production system, including: at least one processor; and memory containing instructions which, when executed by the at least one processor, cause the processor to perform the script production method of any of the above.

[0065] Another example embodiment is a non-transitory memory containing instructions which, when executed by at least one processor, cause the at least one processor to perform the script production method of any of the above.

BRIEF DESCRIPTION OF THE DRAWINGS

[0066] Reference will now be made, by way of example, to the accompanying drawings which show example embodiments, and in which:

[0067] FIG. 1 illustrates a block diagram of an example show production system and script production method, in accordance with an example embodiment;

[0068] FIG. 2 illustrates a detailed block diagram of the show production system of FIG. 1, in accordance with an example embodiment;

[0069] FIG. 3A illustrates a block diagram of a back end infrastructure for the show production system of FIG. 1, in accordance with an example embodiment;

[0070] FIG. 3B illustrates a detailed block diagram of a script writer module of the back end infrastructure of FIG. 3A, in accordance with an example embodiment;

[0071] FIG. 3C illustrates a detailed block diagram of a script analysis module of the back end infrastructure of FIG. 3A, in accordance with an example embodiment;

[0072] FIG. 3D illustrates a detailed block diagram of rendering modules of the back end infrastructure of FIG. 3A, in accordance with an example embodiment;

[0073] FIG. 4 illustrates a block diagram of a director device for the show production system of FIG. 1, in accordance with an example embodiment;

[0074] FIG. 5 illustrates a block diagram of a user device for the show production system of FIG. 1, in accordance with an example embodiment;

[0075] FIG. 6 illustrates an example message format of a script segment message for the show production system of FIG. 1, in accordance with an example embodiment;

[0076] FIG. 7A illustrates an example interface screen for the director device of FIG. 4 to display and edit script segments, in accordance with an example embodiment;

[0077] FIG. 7B illustrates another example interface screen continued from the interface screen of FIG. 7A;

[0078] FIG. 7C illustrates another example interface screen for the director device of FIG. 4 to display and edit different narrative branches of script segments, in accordance with an example embodiment;

[0079] FIG. 8 illustrates an example video generated from the example interface screen of FIGS. 7A and 7B, in accordance with an example embodiment.

[0080] FIG. 9A illustrates an example flow diagram of the script production method of FIG. 1, in accordance with an example embodiment;

[0081] FIG. 9B illustrates another example flow diagram of the script production method of FIG. 1, in accordance with an example embodiment; and

[0082] FIG. 9C illustrates another example flow diagram of the script production method of FIG. 1, in accordance with an example embodiment.

[0083] Similar reference numerals may have been used in different figures to denote similar components.

DETAILED DESCRIPTION

[0084] Example embodiments relate to an end-to-end show production system and script production method for

the automatic generation of a character-driven narrative, powered by machine learning (ML), artificial intelligence (AI), and natural language processing (NLP). Audiences can interact with these narratives in real-time through a variety of media, including audio and visual channels. In doing so, the show production system enables co-creation between machine and users for dynamic authorship and storytelling, resulting in increased audience engagement and uniquely produced video shows. The show production system can be used to generate content rich video that goes beyond a simple static character with a moving mouth.

[0085] Example embodiments relate to show production systems and script production methods for generating scripts and video shows in real-time. An example of the script production method can be applied to the scenario where a user has a script seed input that the user wishes to be the starting point for a video show. In response, the script production method uses one or more script production modules to dynamically generate in real-time new script segments for a new script from the script seed input. The new script can include the same or different characters having phrases and stage directions that stem from the script seed input. The script production method uses the one or more script production modules to dynamically generate in real-time a new video from the script. The incremental nature of the script generation enables audiences and directors to influence the outcomes of the video show. Other enrichment data, audio enrichment data and visual enrichment data can be generated by the script production modules and incorporated into the new video.

[0086] In some example embodiments, the new video is streamed (live broadcast or live published) to the audience prior to completion of the script. For example, as each script is generated, those scripts can be generated into videos and streamed to the audience. Real-time feedback from the audience can be used for the script production method to generate new scripts. The script production method can, in real-time, propose potential new scripts. In some examples, the user can approve, edit or delete prior to the video for those scripts being generated and streamed. In some examples, the user can provide feedback on the video in real-time which is used to generate future proposed scripts that are used by the script production method to continue the generation of the video. The script production method can also generate multiple potential script segments (different narrative branches that continue the narrative) that can be selected, approved, edited or deleted by one or more of the users.

[0087] An example embodiment is a script production method, comprising: receiving a script seed input that represents a beginning of a narrative; generating, using the script seed input and a script writer module: a queue of one or more script segments of a script which continues the narrative; locking at least one of the script segments in the queue; generating, using one or more of the script segments and a script analysis module: respective script metadata for the one or more script segments of the script; generating, using the at least one of the script segments that are locked, the respective script metadata and one or more rendering modules: one or more video segments; and receiving feedback on one or more of the script segments in the queue that have not been locked.

[0088] Another example embodiment is a script production method, comprising: sending, to one or more script

production modules, a script seed input that represents a beginning of a narrative; receiving, from the one or more script production modules, a queue of one or more script segments of a script which continue the narrative; receiving a locking of at least one of the script segments in the queue; receiving, from the one or more script production modules, one or more video segments that are generated from the at least one of the script segments that have been locked; and sending, to the one or more script production modules, an edit, a deletion, or an approval of at least one of the script segments in the queue that have not been locked.

[0089] FIG. 1 illustrates an example show production system 100 (also called narrative production system) and script production method, in accordance with an example embodiment. An example of the show production system 100 can be applied to the scenario where a user has a script seed input 122 that the user wishes to be the starting narrative or starting point for a video show (video 118). In response, the script production method uses one or more script production modules 112 to dynamically generate in real-time a new script from the script seed input 122. The new script can include different characters having phrases and stage directions that stem from the script seed input 122. A phrase is what is to be spoken by a character. A stage direction is a description of what is to happen with one or more of the characters (e.g. movement, position, or tone), objects, background, or other enrichment of the video 118. The script production method uses the one or more script production modules 112 to dynamically, incrementally, and iteratively generate in real-time the new video 118 from the new script. Other video enriching data, such as audio enrichment data and visual enrichment data can be generated by the script production modules 112 and incorporated into the new video 118.

[0090] As shown in FIG. 1, the show production system 100 can include: a director front end 102 including a director device 108 having a director interface 140; an audience front end 104 including one or more user devices 110 (each or collectively 110) each having an audience interface 142; and a back end infrastructure 106. The director device 108 is operated by a director in some examples. In other examples, the functions of the director device 108 are automatically performed and a director is not needed. A user device 110 is operated by an audience member in some examples. The back end infrastructure 106 includes script production modules 112. The back end infrastructure 106 and the script production modules 112 can include machine learning models 120 (each or collectively 120), as well as manual input functions or rules based functions. The script production modules 112 are configured to perform script production functions. At least one of the script production modules 112 are configured to generate, from the script seed input 122, one or more script segments 116 (each or collectively 116). For example, new dialogue (phrases) are generated by the script production modules 112 for the characters to speak. For example, new stage directions are generated by the script production modules 112. The script includes one or more script segments 116, which can be sequential. The script can also be referred to as a screenplay.

[0091] At least one of the script production modules 112 are configured to generate, from each of the script segments 116: the video 118. The back end infrastructure 106 sends the video 118 to the user devices 110, who can provide real-time feedback to the script production modules 112. The

back end infrastructure 106 also sends the video 118 to the director devices 108 who can provide real-time feedback to the script production modules 112. The real-time feedback can be used by the script production modules 112 to generate new script segments 116, which can include new dialogue and/or scene directions and in turn generate new video 118.

[0092] In some examples, at least one of the script production modules 112 are configured to generate, from the script segments 116, metadata containing video enriching data so that the characters and setting are more dynamic. The metadata can also be rules based or manually input. The video 118 is generated taking into account the additional metadata and the video enriching data.

[0093] In an example, the director device 108 and the user devices 110 communicate over a front end interface 114. The director device 108, the user devices 110, and the script production modules 112 therefore collaborate to generate the script segments 116 and the video 118, which can be denoted 3-way authorship. In an example, the director device 108 can receive feedback from the user devices 110, and then the director device 108 is the ultimate decider on whether a script segment 116 on the back end infrastructure 106 is approved to be generated into a video 118.

[0094] In other examples (not shown), the director device 108 and the user devices 110 communicate through the back end infrastructure 106 or through other platforms or servers. In some examples, the director device 108 and the user devices 110 communicate through messaging platforms, audioconference platforms or videoconference platforms.

[0095] In some example embodiments, the generated video 118 is streamed (live broadcast or live published) to the audience front end 104 prior to generating a completed script. For example, one or more script segments 116 are generated at a time. As script segments 116 are streamed to the audience front end 104, real-time feedback from the user devices 110 and the director device 108 can be used for the script production method to generate new script segments 116. The script production method can, in real-time, propose potential new script segments 116, to which the director device 108 can approve, edit or delete prior to the video 118 for those script segments 116 being generated and streamed. In some examples, the script and the video 118 do not necessarily end to completion, and can be continuously streaming indefinitely, e.g. autonomously or based on continuous real-time engagement with audience members and/or the director.

[0096] In some examples, not shown, there is more than one director device 108 whom can collaborate on directing the video 118. In some examples, there is no director device 108. For example, the script production modules 112 can be left to run autonomously and automatically after receiving the script seed input 122. In some examples, the script production modules 112 can receive feedback directly from the user devices 110 of the audience front end 104 in the absence of the director device 108, and the video 118 is generated from audience feedback, voting, consensus, one or a particular subset of user devices 110, etc. In some examples, any one the user devices 110 can be designated as the director device 108, and the user of that user device 110 is the director.

[0097] FIG. 2 illustrates the show production system 100 in greater detail. The show production system 100 is configured to perform the script production method, and incrementally and iteratively generate a video show based on

real-time user feedback. The script production modules 112 include a script writer module 130, a script analysis module 132, and one or more rendering modules 128 including a back end rendering module 134 and a front end rendering module 136.

[0098] The script production modules 112 can communicate using script segment messages 150 (each or collectively 150). A detailed example message format of the script segment messages 150 is illustrated in FIG. 6. Each script segment message 150 represents a script segment 116, such as a scene direction or a phrase (including the character who is speaking the phrase). Each script segment message 150 can be used to generate a respective video segment of the video 118. The script segment message 150 can be transmitted or otherwise accessed between the script production modules 112. In some examples, the script segment message 150 contains information that can be used by the rendering modules 128 to generate the video segment of the video 118.

[0099] In an example scenario of the script production method, the director device 108 transmits a script seed input 122 to the script writer module 130. The script seed input 122 can be based on feedback from the director (through the director interface 140) and the user devices 110. The script seed input 122 represents a beginning of a narrative, for example, the script seed input 122 starts a narrative or show that is desired by the director to be the starting point for generating a new video 118. The narrative is a human-understandable scene in some examples, which have a narrative structure familiar to the audience that stems from the script seed input.

[0100] In some examples, the script seed input 122 is already in the format of the script segment message 150 from the director device 108 and includes script segments 116. In some examples, the script writer module 130 pre-processes and converts the received script seed input 122 into one or more script segments 116. The script segment 116 can include a stage direction and/or a phrase (including the character speaking the phrase). In an example of the script segment message 150, the script segment 116 can be one of: a stage direction, a phrase (including character) or a system signal 156.

[0101] The script writer module 130 generates, using the script seed input 122, one or more new script segments 116 that continue the narrative and stem from the script seed input 122. As more script segments 116 are generated, the script writer module 130 generates more screen segments 116 based on the script seed input 122 and all previous pending or approved screen segments 116.

[0102] The script analysis module 132 generates, using the script segments 116 from the script writer module 130, segment metadata 154 (also referred to as a script metadata) and system signals 156. The segment metadata 154 is content and video enriching data based on text analysis of the script segments 116, and can include sentiment data, music data, voice data, and other data. The system signals 156 can be data extracted from the text of the script segments 116 or can be received from the director device 108. The script analysis module 132 can generate a new script segment message 150 which contains the system signal 156. In some examples, the system signals 156 are based on manual input from the director interface 140 or the audience interface 142. The system signals 156 can include adding a character, removing a character, or error. In examples, the system signals 156 can include scene change,

animation triggers, sound effect triggers, ending a scene, lighting cues, camera cues, ending a show, etc.

[0103] The back end rendering module 134 receives the script segment messages 150, which can include the script segment 116, the segment metadata 154, and/or the system signals 156. The back end rendering module 134 generates, from one or more of the script segment messages 150, video enriching data which can include audio enriching data 160 and/or visual enriching data 162. The back end rendering module 134 can also pass through the script segment message 150 to the front end rendering module 136.

[0104] The front end rendering module 136 generates (renders), using the audio enriching data 160 and the visual enriching data 162 in relation to one or more script segment messages 150, the video 118 (including one or more video segments). The front end rendering module 136 renders the video 118. In an example, the front end rendering module 136 uses conventional gaming engine rendering software and does not use machine learning. In examples, the video 118 can be in a variety of formats. In examples, the front end rendering module 136 formats the video 118 into a particular platform or video player that is being used by the user devices 110 and the director device 108. The front end rendering module 136 can transmit a customized format of the video 118 to one or a group of the user devices 110. Different customized formats of the video 118 can be generated at the same time by the front end rendering module 136. The front end rendering module 136 can also customize the particular resolution depending on the user device 110, for example standard definition, High Definition (HD), or 4K, etc.

[0105] In some examples, the front end rendering module 136 is in the user device 110 and can render the video 118 based on the video enrichment data generated by the back end rendering module 134. The user device 110 renders the video 118 to a format suitable for the particular device capabilities, platform or video player of that user device 110.

[0106] As shown in FIG. 2, the script segment message 150 can be sent (passed through) from the front end rendering module 136 to both the director interface 140 of the director device 108 and to the audience interface 142 of the user device 110. The script segment message 150 can be a proposed script segment message 150 that contains proposed scene direction, proposed phrase (including proposed character to speak the proposed phrase) or proposed system signals 156. Based on feedback received through the director interface 140 or the audience interface 142, the director device 108 can edit, add, or approve any of the fields of the proposed script segment message 150 to the script writer module 130. Therefore, creation of the script and the video show is performed incrementally from real-time feedback from the users (director and audience).

[0107] In some examples, any of the script production modules 112 in the back end infrastructure 106 can run irrespective of the presence of other script production modules 112. Without the other script production modules 112, the back end infrastructure 106 can serve as a simple message passing server to pass the script segment messages 150 between the script production modules 112 with suitable constraints.

[0108] In some examples, not shown, a moderator is assigned the role of moderator functions such as filtering unwanted or explicit content. The moderator can be an audience member of the user device 110 or the director

device **108**, who performs manual filtering, censoring, commenting, blocking users, and/or editing of user feedback and proposed script segments **116**. In some examples, not shown, the moderator function is performed autonomously by one or more machine learning models **120**. In some examples, not shown, the moderator function is performed autonomously by moderator rules such as those set by gaming rating systems or media rating systems. The moderator function can include blacklist of explicit content, such as phrases, screen directions, narratives, and the visual enriching data **162**. In an example, the director is the moderator. The moderator function can include whitelist of permitted content, phrases, visual enriching data **162**. In an example, at least one of the audience members in the audience front end **104** is the moderator.

[0109] Details of the back end infrastructure **106** are illustrated in FIGS. 3A-3D. FIG. 3B illustrates a detailed block diagram of the script writer module **130**, in accordance with an example embodiment. In an example, the script writer module **130** is a script writer model that includes one or more neural networks. In an example, the script segment **116** can be generated by the script writer module **130** using the script seed input **122** and all previous screen segments **116** that are either in progress or those that were finalized (e.g., approved by the director or rendered into the video **118**). The script writer module **130** can be a stand alone module in which the final output is a script (or script segments **116**) that is used for other purposes such as conventional show production, script publication, etc.

[0110] In an example, the script writer module **130** is a text generator that includes a NLP model. The script writer module **130** is configured to render the current scene, phrases and actions in a format amenable to the script analysis module **132** or the other script production modules **112**, for example each phrase or stage direction is generated in a message format of the script segment message **150**. The script writer module **130** can be configured to append, correct or remove script segments **116** from the scene. The script writer module **130** uses director inputs and can continue generating script segments **116** if left unattended (if so desired). In an example, the back end infrastructure includes multiple script writer modules **130** connected at the same time, allowing each to represent a specific character.

[0111] NLP refers to a branch of computer science and AI for giving computers the ability to understand text and spoken words in much the same way human beings can. NLP combines computational linguistics, rule-based modeling of human language, with statistical, machine learning, and deep learning models. NLP includes identifying language features such as grammar, sentences, nouns, adjectives, and verbs. NLP includes normalization of whitespace, punctuation, and spelling. NLP includes splitting long messages into sentences so sentiment analysis is more granular. NLP enables computers to process human language in the form of text or voice data and to understand the full meaning, to generate the intent and sentiment for the script or character. NLP drives computer programs that translate text from one language to another, respond to spoken commands, and summarize large volumes of text rapidly.

[0112] An example of the script writer module **130** includes remote NLP **310** (e.g. GPT-3, Lamda) which uses a remote application programming interface (API) call to call into for example, OpenAI's API to generate text in the given format. An example of the script writer module **130**

includes local NLP **312** (e.g. GPT-2, GPT-J) which uses a local processing device to run inference on a modern language model. An example of the script writer module **130** includes manual input module **314**, e.g. a human script-writer, which is run based on manual input without an automatic text generator. An example of the script writer module **130** includes a chat bot module **316** that includes a character that is driven by programmed logic to have certain responses. An examples of the script writer module **130** includes a news bot module **318** in which a character says phrases that are guided from news (e.g. newswires, Associated Press™ or Google™ News) or social media (e.g. Facebook™ or Twitter™).

[0113] FIG. 3C illustrates a detailed block diagram of the script analysis module **132**, in accordance with an example embodiment. In an example, the script analysis module **132** is a script analysis model that includes one or more neural networks. The script analysis module **132** can be a stand alone module in which the input is any script (or script segments **116**) which can be manually entered by the director or automatically generated by the script writer module **130**. The final output from the script analysis module **132** can be enriched data such as the segment metadata **154** or the system signals **156**.

[0114] In an example, the script analysis module **132** inspects the current text of the script segments **116** and analyzes the text for enriching directorial content. This enriching directorial content can include emotion data (emotional qualities) and action logic that can be used by the other script production modules **112** in their functions. The script analysis module **132** appends segment metadata **154** to the script segments **116**, for example using the message format of the script segment message **150**. The script analysis module **132** can generate one or more system signals **156** based on inspection of the text. In some examples, the system signal **156** is received from the director device **108** or one of the user devices **110**. The system signal **156** can be used as a medium through which the director devices **108**, the user devices **110** and the script production modules **112** communicate the system signal **156** for particular messages.

[0115] An example of the script analysis module **132** includes a sentiment module **320** such as Deepmoji from MIT™, which provides sentiment analysis in the form of emoji tied to specific sections of the text. Sentiment data (or metadata) can be extracted by scanning of the phrases (and sometimes from the stage directions) to be spoken by a character to determine the emotions that the character is most likely feeling at the time. An example of the script analysis module **132** includes a stage direction extraction module **322**, in which script segments **116** contain stage direction **610** (FIG. 6) that is not meant to be spoken, but rather read by a director (or audience). The stage direction **610** contains information that influences the stage direction of the scene (one or more script segments **116**). The stage direction extraction module **322** removes any stage direction that could be mistaken as phrase and adds as segment metadata **154**, or generates a new script segment **116** containing the stage direction **610** (FIG. 6). An example of the script analysis module **132** includes a character action extraction module **324**, which infers possible physical actions that a character may take given context of the script segments **116** or the constraints set up by the director. An example of the script analysis module **132** includes a scene description module **326**, which extracts scene description

about the physical layout of the scene, such as set and environment. The scene description can include location, furniture, props, time of day, blocking, etc. An example of the script analysis module 132 includes a manual input module 328, e.g. a human director, in which the human director provides manual interpretation of the script segments 116.

[0116] FIG. 3D illustrates a detailed block diagram of the rendering modules 128, including the back end rendering module 134 and the front end rendering module 136, in accordance with an example embodiment. In an example, the back end rendering module 134 is a back end rendering model which includes one or more neural networks. In an example, the back end rendering module 134 and the front end rendering module 136 are discrete entities with distinct workflows. In an example, the back end rendering module 134 operates asynchronously while the front end rendering module 136 operates in real-time. The rendering modules 128 can operate independently from the other script production modules 112, in which the input to the back end rendering module 134 is any script (or script segments), segment metadata 154 and/or system signals 156. For example, the back end rendering module 134 can receive a script segment message 150 which includes information to be rendered into the video 118. The script segment message 150 can be manually input (created) by the director device 108 in some examples, or can be generated by the script analysis module 132. In response to receiving the script segment message 150, the back end rendering module 134 generates the audio enriching data 160 and the visual enriching data 162, and the front end rendering module 136 generates the video 118 (one or more video segments corresponding to one or more script segments 116).

[0117] In an example, the back end rendering module 134 includes any or all of: i) one or more speech rendering modules 330, ii) one or more music scoring modules 332; and/or iii) one or more asset generation modules 334.

[0118] The speech rendering module 330 can render voice from text using the emotional cues, for example from the sentiment metadata 622 (FIG. 6). An example speech rendering module includes posterior-sampling style transfer using NVIDIA™ Tacotron2. An example speech rendering module 330 includes online Speech-To-Text APIs such as Replica and Azure.

[0119] An example speech rendering module 330 includes real-time human voice acting through a web application.

[0120] The music scoring rendering module 332 can provide a background track to accentuate the dramatic action in the scene. An example music scoring rendering module 332 includes a music selection engine which searches a library of music to play in view of the segment metadata 154 or with certain segment conditions. An example music scoring rendering module 332 includes an AI music generator which uses emotional parameters in the sentiment metadata 622 (FIG. 6) in the scene to generate music. An example music scoring rendering module 332 includes a human equivalent of a musician or disc jockey (DJ) live-streaming.

[0121] The asset generation module 334 embeds visual elements that can be used by the rendering modules 128 to enhance the video 118 and the overall experience. Examples of the asset generation module 334 include: image generator using neural techniques, an image search plugin, and/or 3D asset generation. The asset can be one of the characters or an object in the scene (script segment 116).

[0122] The asset generation module 334 can be configured to generate 3D objects and 3D avatars (for the characters), and background scenes.

[0123] The asset generation module 334 can include a character builder that is configurable from a set of components or textual descriptions. When the same character name is used (e.g. Dwayne in examples herein), the same asset is used.

[0124] The asset generation module 334 can include a scene builder that is configurable based on background, entities (scene items), and other data. The asset generation module 334 can use textual descriptions of each scene/character component, in which the language models assemble a scene that matches the setting and character components (inside seeds). The asset generation module 334 can receive a scene description to generate the scene, e.g., receiving free-form textual description of a setting in which a story would occur. The asset generation module 334 can receive a character description, e.g. a free-form textual description of a character (humanoid/non-humanoid) that takes part in a narrative. The asset generation module 334 can receive scene or character descriptions which define how each will appear as well as how they will be understood by the language models involved in the narrative construction.

[0125] In an example, a character may change shirt color, hair style, expression, age, etc. In an example, an object may change, grow, shrink, change shape, explode, etc. The asset generation module 334 can also repurpose or revise an existing asset when appropriate.

[0126] In an example, the front end rendering module 136 is an audiovisual front-end plug-in that can be rendered to a particular format for sharing e.g. automated podcast, Unity game engine, web browser. For example, the front end rendering module 136 can include video format rendering module 1 (340 (a)), video format rendering module 2 (340 (b)), . . . , video format rendering module n (340 (n)), which are each used to render different video formats. In an example, the video 118 generated by the front end rendering module 136 can include a plurality of video frames (images). The video 118 can comprise one or more video segments (which is a different concept than a video frame). Each video segment corresponds to a script segment 116. The video segments can be sequential. Example formats of the video 118 include video files, video streaming (e.g. Moving Picture Experts Group 2, MPEG2), PNG sequences, stereoscopic (also known as three dimensional) video, equirectangular video, depth-augmented video, Virtual Reality (VR), Augment Reality (AR), three dimensional (3D) environments, voxel-based environments, gaming engine data, although the system is not limited to these formats. The video 118 can be live broadcast (live published) in real-time as video segments are generated (and locked). For example, the user device 110 can select a particular point of view in a 3D environment in order to view the video 118. In an example, the visual enriching data 162 can include closed captioning for the hearing impaired which displays the particular phrase and/or stage direction from the script segments 116. The audio enriching data 160 can include stereophonic audio data (e.g. left and right, or in some examples further spatial audio data such as Dolby™). The audio enriching data 160 can include descriptive audio for the visually impaired, which can be extracted from the stage direction or other fields of the script segment message 150.

In some examples, an additional “character” indicated in the script segment message **150** can be “voiceover” or “God”, which is rendered as audio enriching data **160** without any visual enriching data **162**. When the voiceover is from a particular character, the same voice and tone as that character is audio rendered by the front end rendering module **136**.

[0127] In an example, the front end rendering module **136** (or another script production module **112**) can inspect the queue of approved script segments **116** that are not yet rendered and pace the contents of the video **118** accordingly. For example, if the number of approved script segments **116** is below a first threshold, the front end rendering module **136** can make the characters speak slower, add pauses, or add intervening character or scene actions. For example, if the number of approved script segments **116** is above a second threshold, the front end rendering module **136** can make the characters speak faster, or speed up character or scene actions. If the number of approved script segments **116** is between the first threshold and the second threshold, the front end rendering module **136** can render the action in the video **118** at normal pace. The number of approved script segments **116** in the queue, or a suitable flag (e.g., fast, normal, slow) can be a system signal **156** or other segment metadata **154**.

[0128] The script seed input **122** will now be described in greater detail. In an example, the script seed input **122** is in the format of one or more script segment messages **150** and sent to the script writer module **130**. In an example, the script seed input **122** can be in the format of a script, which can be converted to one or more script segments **116**. For the script seed input **122**, the director can insert various phrases (with characters), scene directions in natural language, or system signals **156** such as adding or removing characters. An example of the script seed input **122** in the format of a script is shown in FIG. 7A. In other examples, the script seed input **122** can be in the format of a screenplay (in a natural language text format, which is then preprocessed, parsed and converted into one or more script segments **116**). In other examples, the script seed input **122** can be in the format of

seed input text (words, sentences or paragraphs in natural language starting the narrative for the desired video), a seed input image, a seed input video, or a seed input audio in the form of a spoken narrative. The various formats of the script seed input **122** can be converted by the script writer module **130** (or another script production module **112**) into descriptive text or vectors as necessary. The script seed input **122** is the start of the narrative, and can be used as the beginning of the script being generated by the script writer module **130** which continues the narrative.

[0129] There can be multiple script seed inputs **122** that are used by the script writer module **130**. For example, the further script seed inputs **122** can relate to a middle of the narrative. The further script seed inputs **122** can include free form text of scene descriptions, character descriptions, and/or object descriptions. The further script seed inputs **122** can be pre-made or can be received after parts of the script or video have already been generated. The further script seed inputs **122** can input through the director device **108** or the user devices **110**.

[0130] In an example, a chat box through the director device **108** or the user devices **110** is used to input the further script seed inputs **122**.

[0131] In an example, the script seed inputs **122** are contained in a document such as an XML document. In examples, the XML document can be edited by the director device **108** or the user devices **110**, or via 3-way authorship. In examples, the XML document can be manually edited or autonomously edited by the script writer module **130**. The XML can include natural language text as to what is to happen at that particular time, in seconds, of the narrative (script). In an example, the XML document can also contain other trigger events rather than time in seconds. In an example, the XML document is editable in real-time as script segments **118** are approved and locked, with parts of the XML document being locked for editing which correspond to the locked script segments **118** (and already rendered into the video **118** and live published).

[0132] An example XML document for the script seed inputs is as follows:

```
<?xml version="1.0"?>
<Bot name="BranchingTemplate">
  <Execute>
    <Run template="Branching" />
  </Execute>
</Templates>
<Template name="Branching">
  <Scene name="DinnerParty">
    <Delay duration="1s" />
    <SetText name="ShowTitle" value="Potatoes" />
    <!-- Scene Setup -->
    <SetSelect character="Dwayne" name="DPCharacterStage" value="1" />
    <SetSelect character="Sarah" name="DPCharacterStage" value="3" />
    <SetTrigger name="Intro" />
    <Delay duration="7s" />
    <!-- Frame1 -->
    <Bulk>Dwayne and Sarah sit around after a great meal. Sarah is mad at Dwayne and wants
to breakup.
      Dwayne: Babe, those potatoes are (censored) delicious.</Bulk>
    <GenDialog characters="Dwayne, Sarah" duration="30s" />
    <Random>
    <!-- Frame1-1 -->
    <Group>
    <Bulk>Insert stage direction here.</Bulk>
    <GenDialog characters="Dwayne, Sarah" duration="30s" />
    <Random>
    <!-- Frame1-1-1 -->
```

-continued

```
<Group>
  <Bulk>Insert stage direction here.</Bulk>
  <GenDialog characters="Dwayne, Sarah" duration="30s" />
  <Random>
    <!-- Frame1-1-1-1 -->
    <Group>
      <Bulk>Insert stage direction here.</Bulk>
      <GenDialog characters="Dwayne, Sarah" duration="30s" />
    </Group>
    <!-- Frame1-1-1-2 -->
    <Group>
      <Bulk>Insert stage direction here.</Bulk>
      <GenDialog characters="Dwayne, Sarah" duration="30s" />
    </Group>
  </Random>
</Group>
<!-- Frame1-1-2 -->
<Group>
  <Bulk>Insert stage direction here.</Bulk>
  <GenDialog characters="Dwayne, Sarah" duration="30s" />
  <Random>
    <!-- Frame1-1-2-1 -->
    <Group>
      <Bulk>Insert stage direction here.</Bulk>
      <GenDialog characters="Dwayne, Sarah" duration="30s" />
    </Group>
    <!-- Frame1-1-2-2 -->
    <Group>
      <Bulk>Insert stage direction here.</Bulk>
      <GenDialog characters="Dwayne, Sarah" duration="30s" />
    </Group>
  </Random>
</Group>
</Random>
</Group>
<!-- Frame1-2 -->
<Group>
  <Bulk>Insert stage direction here.</Bulk>
  <GenDialog characters="Dwayne, Sarah" duration="30s" />
  <Random>
    <!-- Frame1-2-1 -->
    <Group>
      <Bulk>Insert stage direction here.</Bulk>
      <GenDialog characters="Dwayne, Sarah" duration="30s" />
    </Group>
    <!-- Frame1-2-1-1 -->
    <Group>
      <Bulk>Insert stage direction here.</Bulk>
      <GenDialog characters="Dwayne, Sarah" duration="30s" />
    </Group>
    <!-- Frame1-2-1-2 -->
    <Group>
      <Bulk>Insert stage direction here.</Bulk>
      <GenDialog characters="Dwayne, Sarah" duration="30s" />
    </Group>
  </Random>
</Group>
<!-- Frame1-2-2 -->
<Group>
  <Bulk>Insert stage direction here.</Bulk>
  <GenDialog characters="Dwayne, Sarah" duration="30s" />
  <Random>
    <!-- Frame1-2-2-1 -->
    <Group>
      <Bulk>Insert stage direction here.</Bulk>
      <GenDialog characters="Dwayne, Sarah" duration="30s" />
    </Group>
    <!-- Frame1-2-2-2 -->
    <Group>
      <Bulk>Insert stage direction here.</Bulk>
      <GenDialog characters="Dwayne, Sarah" duration="30s" />
    </Group>
  </Random>
</Group>
</Random>
</Group>
```

-continued

```

</Random>
<!-- Outro -->
<SetTrigger name="Outro" />
<Delay duration="10s" />
</Scene>
</Template>
</Templates>
</Bot>

```

[0133] FIG. 3A illustrates a detailed block diagram of the back end infrastructure 106, in accordance with an example embodiment. The example back end infrastructure 106 shown in FIG. 3A includes at least one memory 302 (one shown), at least one processor 304 (one shown), and at least one communications interface 306 (one shown). A communication connection is implemented between the memory 302, the processor 304, and the communications interface 306, for example using a bus or other communication protocols. The processor 304 is configured to execute, from the script production modules 112 stored in the memory 302, steps of the script production method as detailed in FIG. 2.

[0134] The memory 302 can be a read-only memory (ROM), a static storage device, a dynamic storage device, or a random access memory (RAM). The memory 302 may store programs such as the script production modules 112. The memory 302 can be a non-transitory memory. The memory 302 can store one or more databases 308 (one shown). The database 308 can include all data received or generated by the script production modules 112. For example, the database 308 can include feeds, audience profiles, director profiles, script seed inputs 122 to the show production system 100, repository of scripts and videos, historical user feedback, the script segment messages 150, and any generated script segments 116 and generated videos 118. The database 308 can include asset libraries, which can include 3D objects and 3D avatars (for the characters), and scenes (both generated and pre-made). The same asset is used when the asset returns to a later scene in the instant script or instant video 118 being generated by the script production modules 112. The database 308 can include assets that were originally generated by one of the asset generation modules 334 and stored for re-use by another script or video 118. In an example, when a general description of an asset is manually or automatically provided, the closest asset from the database 308 is selected and used for the particular script segment message 150.

[0135] In some examples, the database 308 can include script segments 116 that were proposed by the script writer module 130 and not accepted (e.g. deleted) by the director device 108 or the user devices 110. The script writer module 130 can learn not to propose (generate) the same or similar script segment 116 that was deleted. For immediate purposes, the script writer module 130 will not cause an infinite loop of repeatedly proposing the same script segment. For long term purposes, the script writer module 130 learns particular trends and preferences of the director and the audience.

[0136] The processor 304 can be a general central processing unit (CPU), a microprocessor, an application-specific integrated circuit (ASIC), a graphics processing unit (GPU), a Tensor Processing Unit (TPU), or one or more integrated circuits. The processor 304 may be an integrated

circuit chip with a signal processing capability. In an implementation process, steps of the script production method as described herein can be performed by an integrated logical circuit in a form of hardware or by an instruction in a form of software in the processor 304. In addition, the processor 304 can be a general purpose processor, a digital signal processor (DSP), an ASIC, a field programmable gate array (FPGA) or another programmable logic device, a discrete gate or a transistor logic device, or a discrete hardware assembly. The processor 304 can implement or execute the methods, steps, and logical block diagrams that are described in example embodiments. The general purpose processor can be a microprocessor, or the processor may be any conventional processor or the like. The steps of the script production method described with reference to the example embodiments may be directly performed by a hardware decoding processor, or may be performed by using a combination of hardware in the decoding processor and a software module. The software module may be located in a mature storage medium in the art, such as a random access memory, a flash memory, a read-only memory, a programmable read-only memory, an electrically erasable programmable memory, or a register. The storage medium is located in the memory 302. The processor 304 reads information from the memory 302, and completes, by using hardware in the processor 304, the steps of the script production method.

[0137] For the script production modules 112, a respective machine learning model 120 can be implemented by a respective neural network running on the back end infrastructure 106. For the script production modules 112, rules based models can implemented in some examples. For the script production modules 112, manual instructions and manual messages can be implemented in some examples.

[0138] The communications interface 306 implements communication between the back end infrastructure 106 and another device or communications network by using wireless or wired communication. In some examples, the training data may be obtained by using the communications interface 306.

[0139] In an example, the processor 304, with the communications interface 306, executes a live data application program interface (API) to communicate with feeds and third party services e.g. third party 3D object databases containing character avatars (characters) or objects.

[0140] It should be noted that, although the memory 302, the processor 304, and the communications interface 306 are shown in the back end infrastructure 106 in FIG. 3A, in a specific implementation process, a person skilled in the art should understand that the back end infrastructure 106 may further include other components that are necessary for implementing normal running. In addition, based on specific needs, a person skilled in the art should understand that the back end infrastructure 106 may further include hardware

components that implement other additional functions. In addition, a person skilled in the art should understand that the back end infrastructure 106 may include only a component required for implementing the embodiments, without a need to include all the components shown in FIG. 3A. In other examples, the back end infrastructure 106 is executed on a local computer, the director device 108 or the user device 110, referred to as edge processing.

[0141] In some examples, the back end infrastructure 106 includes a server or a cloud server. In some examples, the back end infrastructure 106 includes third party machine learning processing and storage services such as Amazon Web Services (AWS)TM, Microsoft AzureTM, and Google CloudTM. In other examples, not shown, one or more of the script production modules 112 are executed by the director device 108, or one of the user devices 110, or other devices.

[0142] In some alternate examples (not shown here), the director device 108 or the user devices 110 include one or more of the script production modules 112 (including machine learning models 120, rules based modules, or manual input modules).

[0143] FIG. 4 illustrates a block diagram of the director device 108, in accordance with an example embodiment. The director device 108 can be an electronic device or user equipment. The director device 108 can be used by the director. The example director device 108 includes a memory 402, a processor 404, and a communications interface 406. The director device 108 can include input or output (I/O) interface devices 408, including but not limited to touch screen, display screen, keyboard, camera, microphone, speaker, mouse, and/or haptic feedback. In another example, the role of the director device 108 (or director) is learned and performed by the script production modules 112.

[0144] The memory 402 can store a script production application 410 for execution by the processor 404. In some examples, a web browser is used to execute the script production application 410. In other examples, the script production application 410 is a dedicated application. The script production application 410 can generate the director interface 140 to receive commands, edits, the script seed input 122, and other functions to or from the director which interact with the back end infrastructure 106. The director device 108 can include a front end interface 114 (FIG. 1) which interacts and communicates with the user devices 110, for example to create, edit and approve script segments 116. The front end interface 114 (FIG. 1) can be integrated with the director interface 140 in some examples. The memory 402 can store a player application 412, for execution by the processor 404, which is used to display the video 118 to the director. In some examples, the player application 412 is executed by a particular platform such as a video platform, social media platform, streaming platform, web platform, gaming platform, application plug-ins, etc. The script production application 410 and the player application 412 can be separate applications, as shown, or can be combined as a single application, not shown.

[0145] In some examples, the script production application 410 receives manually input information for the script seed input 122 from the director through the director interface 140 and forwards to the back end infrastructure 106. In some examples, the script production application 410 formats the information for the script seed input 122 into the same message format as the script segment message 150.

For example, each phrase of stage direction the script seed input 122 is converted into a script segment 116 in the format of the script segment message 150.

[0146] FIG. 5 illustrates a detailed block diagram of the user device 110, in accordance with an example embodiment. The user device 110 can be used by an audience member. The example user device 110 includes a memory 502, a processor 504, a communications interface 506, and I/O interface devices 508. The memory 502 can store a script production application 510 for execution by the processor 504. The script production application 510 can generate the audience interface 142 to receive commands, edits, and other functions to or from the audience which interact with the back end infrastructure 106. The memory 502 can store a player application 512, for execution by the processor 504. The components of the user device 110 can be similar to those described in relation to the director device 108 (FIG. 4). In some examples, the player application 512 is executed by a particular platform such as a video platform, mobile platform, social media platform, streaming platform, web platform, gaming platform, application plug-ins, etc. The user device 110 can include a front end interface 114 (FIG. 1) which interacts and communicates with the director device 108, for example to create, edit and approve script segments 116. The front end interface 114 can be integrated with the audience interface 142 in some examples. In other examples, the front end interface 114 is separate from the audience interface 142 such as the front end interface 114 being a third party messaging platform, third party video or audio conferencing service, or other third party communication service.

[0147] An example platform for the player application 512 is TwitchTM, can be used for live streaming (live broadcast or live published) of the video 118. The player application 512 can also include audio feedback by way of a chat box for sending and receiving public and private messages between users and with the script production modules 112.

[0148] In an example, a live person has live video (camera) and audio (audio voice input or live sounds) through the player application 512, e.g. through TwitchTM. In an example, the live person is treated as a character by the script production modules 112, the audio voice input is converted to text and is treated as a phrase that the character spoke, and the candidate script segments (and other data) are generated by the script production modules 112 that continue the narrative including the audio voice input. As well, image processing is used to interpret the action and sentiment of the live person received through the video (camera).

[0149] In some examples, the script production application 510 receives information such as proposed script segments 116 from the audience through the audience interface 142 and forwards to the director device 108. In some examples, the script production application 410 formats the information into the same message format as the script segment message 150. For examples, each script segment 116 is a script segment message 150.

[0150] The script production application 510 can receive a script segment message 150 from the back end infrastructure 106 or the director device 108, which contains a proposed script segment 116. The script segment message 150 can contain the proposed script segment 116 to which the user device 110 can provide feedback or proposed edits to the director device 108.

[0151] The user device 110 can be an electronic device or user equipment for interacting with the audience. The user device 110 can be a desktop, a laptop, a set top box, or a mobile communication device such as a smart phone or a tablet. In an example, one of the I/O interface devices 508 can be a communal display screen, for example a movie theatre screen. In such an example, the audience can provide feedback through their user devices 110 which causes output of the video 118 that was generated via 3-way authorship, to the communal display screen.

[0152] FIG. 6 illustrates an example message format of the script segment message 150, in accordance with an example embodiment. Each script segment message 150 represents a script segment 116, and each script segment message 150 can be used to generate a respective video segment of the video 118. The script segment message 150 can be transmitted between the script production modules 112. For example, each of the script production modules 112 can be configured with interceptors or state based triggers in order to respond to the script segment message 150 from the database 308. In other examples, one script production module 112 can receive a script segment message 150, generate new data for the script segment message 150, and send the script segment message 150 to the next script production module 112 in the pipeline of the script production method. The script segment message 150 can also be transmitted to and from the director device 108 and the user devices 110.

[0153] In an example, each script segment message 150 represents a script segment 116. The script segment 116 can be, for example, a stage direction or a phrase (including the character who is speaking the phrase). A script can have a number of script elements or fields, including: scene heading; action; character; phrase; parenthetical; extensions; transition; and shot. In an example, these script elements can each be a respective separate script segment 116. In other examples, some of these script elements can be combined into a particular script segment 116.

[0154] A script or proposed script can be pieced together from a number of the script segment messages 150 in sequence. Similarly, the video 118 can be generated (rendered) from a number of the script segment messages 150 in sequence. Using script segment messages 150 as a communication protocol, the script segments 116 can be generated by the script writer module 130 and transmitted to the director interface 140 for approval or editing by the director device 108. Each script segment message 150 can be generated from a stage direction or a phrase that is manually created by the director device 108 or one of the user devices 110.

[0155] The script segment message 150 includes a plurality of fields or metadata. The script segment message 150 includes a script segment field 602 which represents a script segment 116. The script segment field 602 includes segment ID, dispatch ID, trigger timestamp, estimated duration, locked, deleted, segment metadata 154 and subsegment 604. Locked means approved to be rendered into the video 118. Note that “deleted” script segments 116 are still taken into account by the script writer module 130 to avoid infinite loops and to avoid continuously proposing the same script segment 116 over and over. The “deleted” script segments 116 can also be used by the script writer module 130 to learn preferences of the director and the audience.

[0156] The subsegment 604 includes one of: the system signal 156, the stage direction 610, and the phrase 612. The system signal 156 can include the signal kind 614, the character 616, and a message. The signal kind 614 can include error, add character or remove character. In other examples, the signal kind 614 can include scene change, animation triggers, sound effect triggers, ending a scene, lighting cues, camera cues, ending a show, etc. The character 616 field can include the name (name of the character, e.g. Dwayne or Sarah as in FIG. 7A) and additional metadata relating to the character such as whether the character is on-stage, denoted by the “active” field.

[0157] The segment metadata 154 includes processing metadata 618, audio metadata 620, sentiment metadata 622, and keyword metadata 624, and other metadata 628. The processing metadata 618 includes provider, submitted timestamp, earliest locked timestamp, dispatched timestamp, trigger latency. The processing metadata 618 allows a technical introspection of the state of the system and the performance of the various components. Additionally it allows for proper timing of message delivery in video segments of the video 118. In examples, the other metadata 628 can include: search relevancy metadata, offensiveness metadata, and/or 3D asset metadata, or other metadata, etc.

[0158] The audio metadata 620 includes audio byte buffer, audio duration (in ms), and audio sampling rate.

[0159] The sentiment metadata 622 includes emoji data and sentiment data 626. The sentiment data 626 includes emoji label, emoji character, probability, and emotion. The keyword metadata 624 includes the original message and keywords (which can be extracted from the original message). In an example, keywords in the keyword metadata 624 are given a higher priority or weight than other words in the script segment 116 by the script analysis module 132 and/or the back end rendering module 134. For example, “Dwayne enters and drives a car on the street” may have “car” as a keyword, which is given higher priority than “street”.

[0160] In an example, the sentiment metadata 622 is generated by the script analysis module 132 using Deepmoji from MIT, as understood in the art, which is a deep learning model that understands many nuances of how language is used to express emotions. Emotions can, for example, be expressed as text or as emojis.

[0161] In an example, not shown in FIG. 6, set information and environment information is generated by the script analysis module 132, using the script segment 116. The set information can include one or more objects, building infrastructure, ground infrastructure, background, foreground, etc. The foreground does not include the characters. The environment information can include weather, lighting, wind, gravity value, time of day, location, etc. In some examples, the set information and environment information can be generated by the script analysis module 132 from the stage direction 610 and/or the segment metadata 154.

[0162] The set information and environment information can be contained in the stage direction 610 of the script segment message 150. In some examples, the set information and environment information are stand alone fields in the script segment message 150, for example in the segment metadata 154. In some examples, the set information and environment information can be stand alone files or a separate process performed by the script analysis module 132. The generated set information and environment infor-

mation can be proposed to the director device 108 or the user devices 110 for editing, replacement and approval. In some examples, there can be predefined selectable set information and environment information. In some examples, objects for the set information can be dynamically generated, be in the database 308 (FIG. 3A), or obtained from a third party service or library.

[0163] For example, the stage direction 610 in the script segment message 150 may state “Dwayne sits on chair.” In such an example, the script analysis module 132 recognizes that a chair object is required, and can insert a chair object into the segment metadata 154 for insertion into the video 118. The chair object can be in the keyword metadata 624. A chair object from a 3D model can be obtained by the back end rendering module 134. Alternatively, the script analysis module 132 uses a chair object that is already in the script (i.e., a 3D model of a chair already introduced in a previous script segment 116), and the same chair object is used in the visual enriching data 162. The script analysis module 132 can also generate environment data based on analysis of the script segment message 150. The environment data can include visual enriching data 162. The back end rendering module 134 can also generate environmental audio data for the audio enriching data 160, e.g. birds chirping, traffic, wind, water, conversation din, white noise, etc.

[0164] FIG. 7A illustrates an example interface screen 700 for the director interface 140, in accordance with an example embodiment. Some examples of the example interface screen 700 can be applied to non-visual interfaces such as voice, audio, and haptic (touch). In some examples, the interface screen 700 can include more or fewer fields (or options) than those illustrated in FIG. 7A. The interface screen 700 can include any one or all of the fields of the script segment message 150, which can be edited or approved by the director through the interface screen 700. Any one or all of the fields of the script segment message 150 can be populated by the director through the interface screen 700 to create new script segment messages 150 (to generate new script segments 116 and video segments).

[0165] As shown in FIG. 7A, the example interface screen 700 is a sequence or chronology that runs from top to bottom, as illustrated by time arrow 702. The time arrow 702 can be displayed on the interface screen 700 in some examples, or other indicators can be displayed such as fading out (see FIG. 7B), or otherwise locking particular script segments 116 as those script segments 116 become rendered into video 118.

[0166] The title field 704 is the name of the project, e.g. “Project Dinner Party-Director’s chair”. The script segment fields 706 are script segments 116 shown in chronology. The script segment fields 706 can be in a queue of script segments 116 to be rendered into the video 118. Each script segment field 706 can correspond to one of: a system signal 156, a stage direction 610, or a phrase 612 (including the character speaking the phrase 612). In the present example, the script segment fields 706 are proposed script segments 116 that are manually entered by the director and are used as the script seed input 122. The script seed input 122 is used by the script writer module 130 (FIG. 2) to generate new script segments 116 that continue the narrative of the proposed script segment fields 706. As more script segments 116 are generated by the script writer module 130, those become pending script segments 116 which are displayed after the last script segment field 706g as further script

segment fields 706. Those pending script segments 116 can be pending approval by the director device 108. The director can, through the interface screen 700, edit or delete any pending script segment fields 706, manually insert new script segment fields 706, and approve any pending script segment fields 706.

[0167] The start button 708 and stop button 710 are used to instruct the rendering modules 128 to render the pending script segment fields 706 into the video 118. The status field 712 (e.g. “paused”) indicates the present status of the rendering and the outputting of the video 118 to the director device 108 and the user devices 110. In examples, the status field 712 can be “play”, “paused”, “stop”, etc.

[0168] Referring to the present example in FIG. 7A, the script segments 706 relate to a dialogue between character “Dwayne” and character “Sarah”. Script segment field 706a is a system signal 156 to add character “Dwayne”. Script segment field 706b is a system signal 156 to add character “Sarah”. Script segment field 706c is a stage direction 610 of “Dwayne and Sarah sit across the table from each other after finishing dinner. Dwayne is a gym rat, but he loves Sarah to death. While Sarah knows that Dwayne has a heart of gold, she just can’t see having children with him”. The stage direction 610 is part of the seed narrative for the rest of the video 118. Script segment field 706d is a phrase 612 from character Dwayne: “Babe, those potatoes were delicious, you’re just so good in the kitchen”. Script segment field 706e is a phrase 612 from character Sarah: “Thanks, Babe . . .”. Script segment field 706f is a phrase 612 from character Dwayne: “I really love how things are going with us, you know? I just really feel a connection.” Script segment field 706g is a phrase 612 from character Sarah: “Right.”

[0169] The script segment fields 706a-706g relate to a seed or starting narrative that is desired to have the narrative continued by the script writer module 130. The script segment fields 706a-706g are used as the script seed input 122, in which the script writer module 130 generates further script segment fields 706 (new pending script segments 116) which are to be generated and displayed on the interface screen 700 or to the user devices 110. The script analysis module 132 can generate segment metadata 154 for the script segment fields 706a-706g.

[0170] Referring now to FIG. 7B, the script writer module 130 continues the narrative after script segment field 706g by generating new script segment fields 706. For example, script segment field 706h is a phrase 612 from character Sarah: “I really wished I could see a future with you two.” Script segment field 706i is a phrase 612 from character Dwayne: “I mean we could have kids, if you want to.” Script segment field 706j is a phrase 612 from character Sarah: “I’ve thought about our future a lot recently, and I just think we are better off going separate ways instead of focusing on a family now.”, and so forth with further script segment fields 706. The script analysis module 132 can generate segment metadata 154 for the script segment fields 706h-706j, and so forth. In an example, not shown here, the script writer module 130 can also generate one or more stage directions 610 for the script segment fields 706, as new script segment fields. The phrases between Dwayne and Sarah are an original dialogue.

[0171] The set and direction in the video 118 was extracted from the stage direction 706c, including “sit across the table from each other”, “after finishing dinner”, which

can be populated in the keyword metadata 624 (FIG. 6). In other examples, the particular script seed input 122 can be inspired by a particular setting and/or characters that are preconfigured. The video 118 is generated using the script segment fields 706 and the preconfigured setting and/or characters.

[0172] As shown in FIG. 7B, the script segment fields 706 that have been finalized and rendered to the video 118 are faded out. In an example, for example during real-time streaming of the video 118, those script segment fields 706 are also locked and cannot be edited or deleted through the interface screen 700. In an example, those script segment fields 706 that are faded out have also been published live and online, e.g., to the user devices 110 over the Internet, or to other live media such as television. Therefore, those script segment fields 706 are not editable after being locked and published live as video 118 by the front end rendering module 136. While those script segment fields 706 are being published live, yet more script segment fields 706 are generated by the script writer module 130, as well as segment metadata 154, and are displayed on the interface screen 700 as new script segment fields 706 that are later in the queue, for editing, approval and locking through the interface screen 700.

[0173] In another example, those script segment fields 706 that are faded out have been rendered into video segments to form a video, and those video segments, are viewed live by the director device 108 and the user devices 110 in a closed community. The video created by three way authorship can then be stored, reviewed, and/or later published.

[0174] A horizontal time bar 730 (which may be movable) can also be used to divide the locked from the unlocked script segment fields 706. In an example, not shown here, more script segment fields 706 (previous or pending) may be viewed by zooming out of the interface screen 700 or scrolling through the interface screen 700.

[0175] The script segments (script segment field 706) that are locked are earlier in the queue, above the horizontal time bar 730, and are shown as higher on the interface screen 700. The script segments (script segment field 706) in the queue that have not been locked are later in the queue, below the horizontal time bar 730, and are shown as lower on the interface screen 700. The script segments later in the queue (not locked and below the horizontal time bar 730) can still be edited using the 714 script segment toolbar 714 (FIG. 7A).

[0176] The rendering modules 128 are used to render (generate) the video 118 based on the script segment fields 706. The back end rendering module 134 generates audio enriching data and/or visual enriching data for the video 118. As shown in FIG. 7B, an example of the rendered video 800 is displayed side-by-side with the script segment fields 706 (and other options).

[0177] Referring again to FIG. 7A, a script segment toolbar 714 can be used to edit the particular script segment field 706. A lock option 714a is used to approve and lock in a particular pending script segment field 706. An add option 714b is used to insert a new script segment field 706 at that particular position on the queue or timeline (which can be a system signal 156, a stage direction 610 (in natural language), or a phrase 612). An edit option 714c is used to edit a particular pending script segment field 706. A delete option 714d is used to delete a particular pending script segment field 706. In an example, the addition, editing or deletion of

a particular pending script segment 706 causes the generating of the segment metadata 154 by the script analysis module 132 to start over.

[0178] Referring again to FIG. 7A, an add character option 716 can be used to add a character as a system signal 156 in a new script segment field 706, i.e., the character enters the scene in the video 118. A remove character option 718 can be used to remove a character as a system signal 156 in a new script segment field 706, i.e., the character leaves the scene in the video 118. A phrase option 720 can be used to add a new phrase 612 in a new script segment field 706. A stage direction option 722 can be used to add a new stage direction 610 (in natural language) in a new script segment field 706. A bulk option 724 can be used to add more than one new script segment at a time (as one or more new script segment fields 706). The generate option 726 is used to instruct the script writer module 130 to generate new script segments 116 corresponding to new script segment fields 706, based on all of the previous script segment fields 706.

[0179] In some examples, not shown here, the example interface screen 700 can include the respective segment metadata 154 or any of the fields of the script segment message 150 (FIG. 6), which can be added, edited, deleted, or approved.

[0180] In some examples, not shown, the example interface screen 700 can include a message box to receive messages and feedback from the user devices 110.

[0181] In some examples, not shown, a similar interface screen to the example interface screen 700 can be displayed on the audience interface 142 of the user devices 110 for interaction with the audience. The audience interface 142 for the user devices 110 can include a menu of any one or all of the fields of the script segment message 150, which can be edited, added, deleted, etc. The user devices 110 can propose new script segment messages 150, e.g. with phrase, stage direction, or any one or all of the fields of the script segment message 150. Interactions through the audience interface 142 can be sent from the user device 110 to the director device 108 by way of the script segment message 150 or other communication protocols.

[0182] In some examples, the audience interface 142 can be used by the audience to provide feedback to the director device 108 or to the back end infrastructure 106. The feedback can be in the form of edits, deletions, additions to any fields of the script segment message 150. In some examples, the feedback can include text input, which can be input into a comment box or message box. In some examples, the feedback can include emotions, sentiments, emoji, emoticons, text, subscriptions, likes, donations, votes, ratings, viewing time (i.e., is the user device 110 still displaying the video 118), etc. In some examples, not shown here, the audience interface 142 can include a detector, such as a camera or sensor, that is used to detect feedback from the audience, for example heartbeat, electroencephalogram (EEG) signals, dance signals, facial expressions, temperature, or walking or running speed, etc.

[0183] In some examples, not shown here, the audience interface 142 can include a chat box which includes public and private messages to and from the audience. An example of a streaming platform having a chat box is Twitch™, and the messages in the chat box can be parsed by the script production modules 112 to gauge the audience feedback. In an example, stickers or emojis/emoticons from the chat box (e.g. Twitch™) are used to generate the script segment

messages 150. In an example, these stickers or emojis/emoticons can be rules-based, such as a banana or hotdog sticker meaning that such an item is included in the script segment message 150 and is displayed (e.g. placed, thrown or moving) onto the video 118. In another example, these stickers or emoticons are used for general feedback or context, for example happy face or positive stickers (e.g. unicorn) meaning happy sentiment and context for the script segment message 150, and can result in the proposed script segments 116 continuing the narrative (scene) without substantive changes. Unhappy/negative face or unhappy/negative sentiment stickers can mean unhappy sentiment and context for the script segment message 150, and can result in more drastic changes in the proposed script segments 116 such as adding or removing an asset or character, or adding erratic, exciting or action-based script segments 116.

[0184] In response to the feedback, more script segments 116 and more video 118 can be generated taking into account the feedback. The video 118 generated by the script production modules 112, the audience and the director is denoted 3-way authorship.

[0185] FIG. 7C illustrates another example interface screen 700 for the director interface 140 of the director device of FIG. 4 to display and edit different branches of proposed script segments 116 by the script production modules 112, in accordance with an example embodiment. A similar interface can be used for the user devices 110, with more or fewer features. Any blackouts shown here are for censorship purposes and are not intending to be limiting.

[0186] Generally, the script writer module 130 can generate and propose a plurality of parallel candidate queues 734a, 734b, 734c (each or collectively 734) of script segments 116. Any one of the queues 734 can be selected by the director through the director device 108, or by the audience (e.g. voting) through the user devices 110, or selected autonomously by the script writer module 130 having an automated selection system. The automated selection system takes into account the further seed inputs for performing the selection. In an example, the automated selection system includes a machine learning model. The script segments 116 in the selected queue 734 can then be added, edited, deleted, and approved as described in detail herein.

[0187] The title field 730 is the name of the project, e.g. "Dinner Party-Director's chair". The characters in this example are Dwayne and Kaylee. The seed script input 122 describes the scene, e.g. "Dwayne and Kaylee sit around after a great meal . . .". The script segment fields 732 are script segments 116 shown in chronology. In an example, the script segment field 732a is a manually inserted seed script input 112 in script format (a phrase and the character to speak the phrase, Dwayne).

[0188] The script seed input 122 is used by the script writer module 130 (FIG. 2) to generate a plurality of queues 734, each queue 734 containing proposed script segment fields 706 that continue the narrative. The selected queue 734 of script segments 116 are to be rendered into the video 118. Each script segment field 732 can correspond to one of: a system signal, a stage direction, or a phrase (including the character speaking the phrase). As more script segments 116 are generated by the script writer module 130, those become pending script segments 116 which are displayed after the last script segment field 732e as further script segment fields 732. Those pending script segments 116 can be pending approval by the director device 108. The director can,

through the interface screen 700, edit or delete any pending script segment fields 732, manually insert new script segment fields 732, and approve any pending script segment fields 732.

[0189] Upon selection of one of the queues 734, those script segments in the selected queue 734 are used to further populate the script segment fields 732 at the end of the last script segment field (732e in this example). Those script segment fields 732 can be further enriched by the script production modules 112, and edited and approved accordingly.

[0190] Each script segment field 732 includes status icons 736, which can include any of the following:

[0191] Pen: status of Natural Language Processing

[0192] Emoji: status of the emotional analysis module. It displays the top-ranked result (represented as an emoji) when processing is complete.

[0193] Microphone: status of the Text-to-Speech (TTS) module.

[0194] Checkmark: Ready to deliver/delivered status.

[0195] The status icons can be shown in different colors as follows:

[0196] Grey: unprocessed/not ready for delivery.

[0197] Blue: processed/ready for delivery.

[0198] Green (only used for checkmark): delivered.

[0199] The menu 738 can be used to:

[0200] Play: play (send or display) the rendered video,

[0201] Reset: reset the script segments

[0202] TV Icon: Open a separate window containing a WebGL client for rendering the video show in real-time.

[0203] Download File Icon: Download a transcript of the entire show.

[0204] The menu 740 can be used to:

[0205] Speakers: select the character to speak the next phrase of the queue.

[0206] Degree: select the number of parallel queues.

[0207] Depth: select the number of candidate script segments to generate per queue.

[0208] Clear Memory: remove the queues.

[0209] Reroll: Create new queues. Note that different candidate script segments are generated than the previously proposed candidate script segments.

[0210] Further queues 734 are generated or re-generated in an incremental and iterative fashion, as further script segments 116 are approved and audience feedback is received.

[0211] FIG. 8 illustrates an example of the video 800 generated by the script production modules 112 based on the example interface screen 700 in FIGS. 7A and 7B, in accordance with an example embodiment. For example, one or more script segments 116 are generated by the script writer module 130 from the script seed input 122. For example, based on the feedback provided by the director interface 140, the screen segment 116 can be manually edited, for example the phrase 612, stage direction 610, or other elements of the script segment message 150. The video 118 (e.g. video segments) can be generated by the rendering modules 128 from the script segments 116. For example, the video 118 can be transmitted from the front end rendering module 136 to the user devices 110 and the director device 108 for display of the video 118.

[0212] The example video 800 can include the avatar for the characters, Dwayne 802 and Sarah 804. The example video 800 can include closed captioning 806 of the phrases

from the interface screen 700 in FIGS. 7A and 7B. In an example, the closed captioning 806 can include the stage directions in FIGS. 7A and 7B. The example video 800 includes audio which is from the audio enriching data 160. The audio can be the phrases of the characters, with appropriate pitch, emotion, cadence and timing based on the segment metadata 154. For example, if the character has been configured to speak with a child's voice, the audio enriching data 160 may be rendered with a higher pitch. The audio enriching data 160 can take into account the sentiment metadata 622, to convey the sentiment or emotion in the phrase.

[0213] In some examples, the video 118 can be generated for an indefinite period, theoretically forever to infinity. For example, the video 118 is generated and output (streamed or live published) by the front end rendering module 136 for 24 hours a day. The video 118 can be manually stopped or a particular trigger is detected.

[0214] FIG. 9A illustrates an example of the script production method 900 as performed by the back end infrastructure 106, in accordance with an example embodiment. At step 902, the back end infrastructure 106 receives a script seed input 122 that represents a beginning of a narrative. At step 904, the script writer module 130 generates, using the script seed input 122: one or more script segments 116 (i.e. a script) which continue the narrative. At step 906, the script analysis module 132 generates, using the script segments 116: respective segment metadata 154 for at least one of the script segments 116. At step 908, the rendering modules 128 generate, using the script segments 116, the respective segment metadata 154: one or more video segments of the video 118.

[0215] FIG. 9B illustrates an example of the script production method 920 as performed by the director device 108, in accordance with an example embodiment. At step 922, the director device 108 receives, through the interface screen, a script seed input 122 from the director. The script seed input 122 represents a beginning of a narrative. In some examples, the script seed input 122 is proposed by one or more of the user devices 110 and is edited and/or accepted by the director.

[0216] At step 924, the director device 108 sends, to the back end infrastructure 106 which includes the script production modules 112, the script seed input 122. At step 926, the director device 108 receives, from the back end infrastructure 106, one or more script segments 116 (i.e. a script) which continue the narrative. At step 928, optionally, the director device 108 receives feedback on the one or more script segments 116, e.g. from the director through the interface screen or from the user devices 110 (audience members). At step 930, the director device 108 sends, to the one or more script production modules, an edit, deletion, or approval of at least one of the script segments 116. At step 932, the director device 108 receives, from the back end infrastructure 106, one or more video segments that are generated from the one or more script segments 116. The method can loop to step 926, either ad infinitum or until a suitable stop or trigger.

[0217] FIG. 9C illustrates an example of the script production method 940 as performed by the user device 110, in accordance with an example embodiment. At step 942, the user device 110 receives, through a user interface (any of the I/O interface devices 508), first feedback on a script seed input 122 from an audience member. The script seed input

122 represents a beginning of a narrative. In some examples, the first feedback on the script seed input 122 is taken into account, is edited and/or accepted by the director or the script production modules 112.

[0218] At step 944, the user device 110 sends, to either the director device 108 or the back end infrastructure 106, the first feedback on the script seed input 122. At step 946, the user device 110 receives, from the back end infrastructure 106, one or more script segments 116 (i.e. a script) which continue the narrative. At step 948, the user device 110 receives, through the user interface, second feedback on the one or more script segments 116. At step 950, the user device 110 sends, to either the director device 108 or the back end infrastructure 106, the second feedback on the one or more script segments 116. At step 952, the user device 110 receives, from the back end infrastructure 106, one or more video segments that are generated from the one or more script segments 116. The video segments are output to the user interface or a display screen of the user device 110. At step 954, the user device 110 receives, through the user interface, third feedback on the one or more video segments. At step 956, the user device 110 sends, to either the director device 108 or the back end infrastructure 106, the third feedback on the one or more video segments. The method can loop to step 946, either ad infinitum or until a suitable stop or trigger. The method 940 can be performed in parallel from a plurality of user devices 110, so that each respective audience member of the user devices 110 can send their first feedback, second feedback, and/or third feedback.

[0219] As can be appreciated, the back end infrastructure 106 generates the video segments based on the first feedback, second feedback, and/or third feedback from one or more of the audience members.

[0220] The first feedback, the second feedback, and the third feedback can be text inputs or physiological information of the audience detected by a detector of the user device 110. The first feedback can be an addition, edit, or deletion of part of the script seed input 122. The second feedback can be an addition, edit, or deletion of at least one script segment 116. The third feedback can be sentiment information, keyboard inputs (comments), camera captures, sensor information, etc. based on audience reaction to the video segments. In response, the continued narrative of future script segments 116 and video segments generated by the back end infrastructure 106 are affected by audience feedback by way of the first feedback, the second feedback, and the third feedback.

[0221] In the example embodiments, it should be understood that the described show production system 100, the script production method, the back end infrastructure 106, the director device 108, and the user devices 110 may be implemented in other manners. For example, the described back end infrastructure 106 is merely an example embodiment. For example, the unit division is merely logical function division and may be other division in actual implementation. For example, a plurality of units or components may be combined or integrated into another system, or some features may be ignored or not performed. In addition, the displayed or discussed mutual couplings or direct couplings or communication connections may be implemented by using some interfaces. The indirect couplings or communication connections between the units may be implemented in electronic, mechanical, or other forms. In some examples,

the back end infrastructure **106** may be provided on the director device **108** or the user devices **110**.

[0222] The units described as separate parts may or may not be physically separate, and parts displayed as units may or may not be physical units, may be located in one position, or may be distributed on a plurality of network units. Some or all of the units may be selected according to actual requirements to achieve the objectives of the solutions of the embodiments.

[0223] In addition, functional units in the example embodiments may be integrated into one processing unit, or each of the units may exist alone physically, or two or more units are integrated into one unit.

[0224] When the functions are implemented in the form of a software functional unit and sold or used as an independent product, the functions may be stored in a computer-readable storage medium. Based on such an understanding, the technical solutions of example embodiments may be implemented in the form of a software product. The software product is stored in a storage medium, and includes several instructions for instructing a computer device (which may be a personal computer, a server, or a network device) to perform all or some of the steps of the methods described in the example embodiments. The foregoing storage medium includes any medium that can store program code, such as a Universal Serial Bus (USB) flash drive, a removable hard disk, a read-only memory (ROM), a random access memory (RAM), a magnetic disk, or an optical disc. In an example, the software product can be an inference model generated from a machine learning training process.

[0225] In the described methods or block diagrams, the boxes may represent events, steps, functions, processes, modules, messages, and/or state-based operations, etc. While some of the example embodiments have been described as occurring in a particular order, some of the steps or processes may be performed in a different order provided that the result of the changed order of any given step will not prevent or impair the occurrence of subsequent steps. Furthermore, some of the messages or steps described may be removed or combined in other embodiments, and some of the messages or steps described herein may be separated into a number of sub-messages or sub-steps in other embodiments. Even further, some or all of the steps may be repeated, as necessary. Elements described as methods or steps similarly apply to systems or subcomponents, and vice-versa. Reference to such words as “sending” or “receiving” could be interchanged depending on the perspective of the particular device.

[0226] The described embodiments are considered to be illustrative and not restrictive. Example embodiments described as methods would similarly apply to systems or devices, and vice-versa.

[0227] The various example embodiments are merely examples and are in no way meant to limit the scope of the example embodiments. Variations of the innovations described herein will be apparent to persons of ordinary skill in the art, such variations being within the intended scope of the example embodiments. In particular, features from one or more of the example embodiments may be selected to create alternative embodiments comprised of a sub-combination of features which may not be explicitly described. In addition, features from one or more of the described example embodiments may be selected and combined to create alternative example embodiments composed of a

combination of features which may not be explicitly described. Features suitable for such combinations and sub-combinations would be readily apparent to persons skilled in the art. The subject matter described herein intends to cover all suitable changes in technology.

What is claimed is:

1. A script production method, comprising:

receiving a script seed input that represents a beginning of a narrative;

generating, by a script writer module using the script seed input: a queue of one or more script segments of a script which continues the narrative;

locking at least one of the script segments in the queue;

generating, by a script analysis module using one or more of the script segments: respective script metadata for the one or more script segments of the script;

generating, by one or more rendering modules using the at least one of the script segments that are locked and the respective script metadata: one or more video segments; and

wherein the generating the respective script metadata includes generating processing metadata using the script and the script analysis module, wherein the processing metadata is for:

technical introspection of one or more states of the script production method, managing performance of the script production method, or managing timing of message delivery between the script writer module, the script analysis module, and the one or more rendering modules.

2. The script production method of claim 1, further comprising receiving feedback on one or more of the script segments in the queue that have not been locked.

3. The script production method of claim 2, wherein the feedback comprises real-time feedback.

4. The script production method of claim 2, wherein the feedback comprises approval, edit, or deletion of at least one of the script segments in the queue that have not been locked, wherein the script writer module is configured to learn from the feedback.

5. The script production method of claim 2, wherein the generating the queue includes generating a plurality of different parallel queues each respectively having one or more of the script segments; further comprising receiving a selection of one of the different parallel queues.

6. The script production method of claim 5, wherein the receiving feedback is on the queue that is selected.

7. The script production method of claim 5, wherein the selection of the queue is performed by an automated selection system.

8. A non-transitory memory containing instructions which, when executed by at least one processor, cause the at least one processor to perform the script production method as claimed in claim 1.

9. A show production system, comprising:

at least one processor; and

memory containing instructions which, when executed by the at least one processor, cause the at least one processor to perform the script production method as claimed in claim 1.

10. A script production method, comprising:

receiving a script seed input that represents a beginning of a narrative;

generating, by a script writer module using the script seed input: a queue of one or more script segments of a script which continues the narrative;
 locking at least one of the script segments in the queue;
 generating, by a script analysis module using the script and one or more of the script segments:
 respective script metadata for the one or more script segments of the script, wherein the script metadata includes sentiment metadata, wherein the sentiment metadata includes sentiment data or emotion data; and
 generating, by one or more rendering modules using the at least one of the script segments that are locked and the respective script metadata including the sentiment metadata: one or more video segments.

11. The script production method of claim **10**, wherein the generating the one or more video segments uses the sentiment metadata to convey sentiment or emotion within the narrative.

12. A non-transitory memory containing instructions which, when executed by at least one processor, cause the at least one processor to perform the script production method as claimed in claim **10**.

13. A show production system, comprising:

at least one processor; and

memory containing instructions which, when executed by the at least one processor, cause the at least one processor to perform the script production method as claimed in claim **10**.

14. A script production method, comprising: receiving a script seed input that represents a beginning of a narrative;
 generating, by a script writer module using the script seed input: a queue of one or more script segments of a script which continues the narrative;
 locking at least one of the script segments in the queue;
 generating, by a script analysis module using one or more of the script segments: respective script metadata for the one or more script segments of the script;
 generating, by one or more rendering modules using the at least one of the script segments that are locked and the respective script metadata: one or more video segments,

wherein the one or more rendering modules include a back end rendering module and a front end rendering module, wherein the generating the one or more video segments includes:

generating, by the back end rendering module using the script, the respective script metadata:

audio enriching data or visual enriching data; and

generating, by the front end rendering module using the audio enriching data or the visual enriching data: the one or more video segments; and

formatting, using the front end rendering module, the one or more video segments to a video format that is particular to a respective platform of one or more user devices.

15. The script production method of claim **14**, wherein the back end rendering module includes an asset generation module configured to create content for the one or more video segments.

16. The script production method of claim **15**, wherein the content includes a character and/or an object in at least one the video segments.

17. The script production method of claim **15**, wherein the content includes a visual element, audio data, and/or a 3D asset.

18. The script production method of claim **15**, wherein the content is continuous with the narrative.

19. A non-transitory memory containing instructions which, when executed by at least one processor, cause the at least one processor to perform the script production method as claimed in claim **14**.

20. A show production system, comprising:

at least one processor; and

memory containing instructions which, when executed by the at least one processor, cause the at least one processor to perform the script production method as claimed in claim **14**.

* * * * *