

Story ARTist

Marie Kegeleers and Rafael Bidarra
Computer Graphics and Visualization Group
Delft University of Technology
Delft, The Netherlands

M.Kegeleers@student.tudelft.nl/R.Bidarra@tudelft.nl

I. INTRODUCTION

Most content creation applications currently in use are conventional PC applications with visualisation on a 2D screen and indirect interaction, e.g. through mouse and keyboard. Augmented Reality (AR) is a medium that can provide actual 3D visualisation and more hands-on interaction for these applications, due to its technology adding virtual elements to a real-world environment. We explored how AR can be used for story authoring, a specific type of content creation, and investigated how both types of existing AR interfaces, tangible and touch-less, can be combined in a useful way in that context [1]. The Story ARTist application was developed to evaluate the designed interactions and AR visualisation for story authoring. It features a tabletop environment to dynamically visualise the story authoring elements, augmented by the 3D space that AR provides. Story authoring is kept simple, with a linear plot point structure focused on core story elements like actions, characters and objects.

II. THE STORY ARTIST APPLICATION

To explore how AR visualisation and interaction can be used for story authoring, we developed Story ARTist, a prototype AR application for creating simple linear stories. To author a story, the author can create (and edit) plot points, and (for each plot point) select actions and assign characters, objects and environments to markers.

A. Interaction

Story ARTist has a unique interface combining the two categories of AR interaction that currently exist: tangible and touch-less [2], [3]. Tangible interfaces require the user to interact with physical elements like blocks or cards with markers to control virtual elements. In this application, markers can get virtual narrative elements assigned to and be used to move them around. In touch-less interfaces, the user interacts only with virtual elements like menu panels with buttons or holograms. In Story ARTist, this is implemented using hand interaction: authors can indicate their selection by pressing buttons in virtual menus using their physical hands.

B. Plot Point Structure

In Story ARTist, a story line consists of a sequence of plot points, each one representing a single action. When a new plot point is created, an action needs to be chosen and the plot point has to be filled with information related to that action. Once

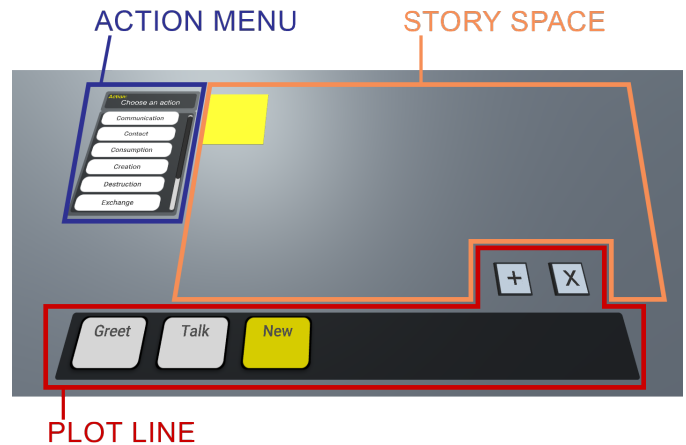


Fig. 1. Overview of the Story ARTist work area and interface

an action is chosen, the interface requests what is needed in the scene to author the chosen action, called its *arguments*. For example, when the author chooses the action *greet*, there need to be two characters in the scene where one character will greet the other. A plot point does not only contain an action and its arguments, but also a *scene*, in which 3D models of narrative elements are present and can be arranged at will using their markers, to visualise what the authored plot point should look like. This is a static representation of the action's arguments that can be seen as a snapshot of the story.

To keep the focus of the Story ARTist application on the AR interaction, only linear plot lines can be authored and no narrative consistency constraints are presently handled [4]. The story is displayed as a linear sequence of plot points called the *plot line*.

C. Interface

Figure 1 displays an overview of the Story ARTist work area, with its three main sectors. In the top left corner, the *action menu* can be found, used to choose an action for each plot point. Once an action is chosen, it is assigned to the plot point that is opened at that moment and the menu changes to display what needs to be added to the scene. This menu implements the touch-less design where the author can use their hands to press virtual buttons for selection.

The *plot line* displayed at the bottom of the application work area contains each authored plot point as a button that

