Prototyping Applications with Tangible User Interfaces in DART, The Designer's Augmented Reality Toolkit

Maribeth Gandy, Steven Dow, Blair MacIntyre,
GVU Center, Georgia Institute of Technology, Atlanta, GA
(maribeth, steven, blair)@cc.gatech.edu

Abstract

In this position statement, we describe our approach to prototyping applications with tangible user interface (TUI) using the Designer's Augmented Reality Toolkit (DART), a system we have created to allow non-technologists to easily create augmented reality (AR) experiences. Over the past year our research has focused on the creation of this toolkit that can be used by technologists, designers, and students alike to rapidly prototype AR applications. This development effort extends to support the configuration of new human interfaces, including tangible user interfaces. Current approaches to AR and TUI development involve extensive programming and content creation as well as knowledge of technical topics involving cameras, trackers, and 3D geometry. It is very difficult, even for technologists, to create AR and TUI experiences. Our goal was to eliminate these obstacles that prevent such users from being able to experiment with AR. The DART system is based on the Macromedia Director multimedia-programming environment, the defacto standard for multimedia content creation. DART uses the familiar Director paradigms of a score, sprites and behaviors to allow a user to visually create complex AR applications. DART also provides low-level support for the management of trackers, sensors, and cameras via a Director plug-in Xtra.

1. Introduction

Over the past few decades, AR researchers have explored a wide variety of domains, but our focus recently is on the use of *dramatic AR experiences* in education and entertainment settings. Unfortunately, the primitive state of AR technology, and the lack of tools to support traditional design activities (such as rapid prototyping and incremental experience testing) has made experiential development difficult.

When working in task-focused domains, such as equipment maintenance and repair, it is feasible for a team of technology experts to work closely with domain experts to understand a particular problem, and build a solution to the problem using widely accepted HCI techniques such as iterative design. Unfortunately, in experiential domains, this separation between domain experts (e.g., designers and artists in this case) and technologists does not work. Designers are most effective when working directly with a medium, and working through an intermediary seriously hinders (or even destroys) the creative process. The primary purpose of our research, therefore, is to advance the state of the art in AR by enabling designers to work directly with AR as a new medium for dramatic experiences.

DART currently focuses on supporting rapid prototyping of AR experiences. DART is built as a collection of extensions to the Macromedia Director multimedia-programming environment, the de-facto standard for multimedia content creation. DART enables designers to rapidly develop and test their AR experiences in the same environment that will be used to deploy the final experience. Designers can gradually evolve their prototypes as they see fit - polished content can be mixed with crude content, elaborate narratives and complex behaviors can be tested as desired, and changes to "complete" experiences can be rapidly prototyped. Our objective is to support all design activities (e.g. storyboarding, informal prototyping, animatic video, designing the agency and interaction of the user, placing graphics in the environment, incorporating physical interfaces) for the creation of dramatic experiences.

In our view of AR, the users' physical environment can be augmented with many sensors and displays in various configurations. TUI's or physical interaction devices naturally complement the visual methods commonly explored in AR (head-mounted displays, ubiquitous projectors). We are currently exploring ideas for information visualization, entertainment, support of design activities, etc. that utilize physical objects as input/output mechanisms with a range of graphical display methods.

Tangible user interfaces can be incorporated into the design of dramatic experiences. The TUI may be a character in a story or a story metaphor manifested in a physical form. Designers will be able to incorporate physical objects

(such as 3D Studio Max models printed on a rapid-prototyping machine) as interface elements in their AR applications by using DART's object model and built-in support for sensors.

2. Motivation

The extensive time required to create content and to write programs to handle all the components of an AR experience often means that there is a large amount of latency between the inception of an idea and its realization. Most instances of successful design using cutting-edge technologies have been slow, painstaking endeavors carried out by close-knit teams or technically sophisticated individual designers. By making AR more accessible to a wider range of designers, we can begin to achieve its potential to create powerful dramatic and educational experiences.

Creating TUI interaction for experiences presents a new set of challenges. Most graphical authoring environments only consider keyboard and mouse interaction, where recent examples of tangible interfaces have taken input from many input features in parallel. Authors may also want to sense and respond to different types of user interaction (positioning vs. rotating vs. touching, etc.) with the physical interface elements. We support the rapid development of TUIs in DART by simplifying the process of attaching physical objects to a sensing infrastructure and by facilitating the visual and interactive design elements based on sensor reports. Physical objects fit into the DART architecture similar to virtual objects, and thus the paradigm is consistent throughout.

3. DART Architecture

The DART system consists of behaviors (extensions to the Director environment written in LINGO) and Xtras (plug-ins for Director written in C++). The DART Xtras provide AR services such as video capture, tracking (via VRPN), and fiducial registration (currently, via the ARToolkit) to Director applications. The user can place behaviors onto the Director score (an arrangement of channels that organize, display and control the application over time). These behaviors allow for the graphical creation of an AR application. Behaviors also represent the basic building blocks of any AR application: virtual objects to be placed in the 3D environment, tracker information, video input (for video mixed AR), and triggers that control the logic (state machine) of the application. The complex services provided by the Xtras are encapsulated by "low level" behaviors that present the user with simple abstractions for services such as VRPN trackers and camera parameters and configurations.

The user can add objects (3D models, 2D storyboards, video, and audio) to her AR environment by placing "Actor" behaviors on the score. These actor behaviors present the user with simple input fields where she can configure such properties as the position and appearance of these objects. These "actors" can have their position linked to tracking information provided by the tracking and fiducial registration behaviors. Objects in the physical environment can also become "actors". For example, in a semi-transparent table-top TUI configuration, a camera below the table can sense the position and rotation of physical objects with attached fiducials on the surface. The "actor" listens to the location reports of the fiducial and updates the associated graphics appropriately. For AR applications, this feature is often used to give physical-world objects presence in the 3D world model – the actor does not have a graphical component, but it occludes graphical elements that should correctly appear behind the physical object. Although fiducial tracking is a simple method for sensing tangible interaction, other sensing technology can be incorporated using VRPN.

DART provides a library of triggers that can be used to create the state machine logic that will control the behavior of the "actors" in the application. These "trigger" behaviors allow the user to specify "when X happens, do Y." For example, the user could place a trigger behavior on an "actor" that would move the position of the actor when a certain sensor value was received, or would start the video on a textured object when a certain time was reached. These simple triggers can be used to build up complex behaviors for an AR application. The key feature of the DART behaviors is that they are not intended to provide comprehensive support for AR experiences: indeed, we believe an attempt to create such a library would be misguided and doomed to failure. Rather, they are designed to provide simple yet complete illustrations of how to create experiences using the low-level AR and TUI services provided by the Xtras, and to integrate with existing Director behaviors and concepts. This allows experienced designers to adapt the behaviors to their own needs.