

CrazyCartoon : the Design and Implementation of a User-defined Strategic Tower Defense System with MobileAR

Yiming Cai, Xiao Li

School of Computer and Information Science, Southwest University
Beibei, Chongqing, P.R.China
835161456@qq.com, lixiaoatswu@139.com

Abstract—Completely different from existing computer games, CrazyCartoon system creates a blending of real (hand-drawn map by a player) and virtual (digital game) scene based on mobileAR technology, which enables players to customize the game strategies freely, such as the level switch and turrets' defensive position. The CrazyCartoon prototype, running on android platform with a combination of Qualcomm Vuforia technologies and Unity Technologies' game engine as well as C# programming, has been proven the robustness of this game system that can track the hand-drawn map in a stable and continual way and overlay virtual objects on the top of the map in a precise manner in real time. The idea, especially importing players' hand-drawn map into computer games, on the other hand is a new challenge to players' imagination and creativity, and on the other hand offers a new approach to the study, design and development of game.

Keywords—user-defined strategies; MobileAR; virtual reality; Tower defense game; implementation and evaluation

I. INTRODUCTION

Tower defense is a subclass of real-time strategy video games, which goal is to try to stop enemies or monsters from destructing camp by building turrets or towers[1]. Augmented Reality (AR) is a kind of technology that overlays virtual objects over the real environment, and its ultimate goal is to enhance the user experience to the real world[2]. Its application on mobile devices can be called mobile augmented reality (MobileAR, or MAR). Apparently different from other systems, the characteristic helps users to keep in contact with the real world, which makes a new window for those who addicted to the digital world to enjoy a lot fun in physical world.

Presently one of the most representative works based on mobileAR is ARDefender (Running on iOS 4.3 or above) [4]. With ARDefender, players have to download and print a PDF file included a pre-define image, a trigger (so-called marker) that can activate the game, and then move the camera in their mobile phones to have a point. Totally different from this system, with CrazyCartoon players can use hand-painted maps by themselves, but also can make a decision to place the defensive tower on their own, which provide them with

challenge to customize game strategy.

The rest of this paper is organized as follows. Section II proposes the design of CrazyCartoon in details, and Section III presents the prototype running on Android and its evaluation. The conclusion section discusses the work for the future as well.

II. SYSTEM DESIGN

A. Game strategies and characteristics

New game strategies are proposed and applied to the CrazyCartoon system, which features are mainly in the following two aspects:

- Customized hand-drawn map levels. A player can draw a map as a level in the video game. Absolutely in this way the player enjoys a lot fun and a great challenge
- Self decision-making the position of the defensive turrets or towers. The players can place the turrets anywhere they want, which makes the game full of change and freedom.

Fig. 1 shows two instances designed by players, a map corresponding to a level in the game.

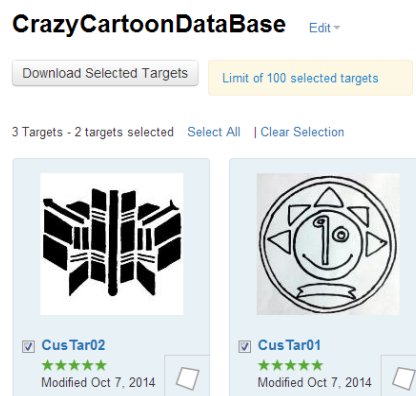


Fig. 1. Two hand-drawn maps in the prototype of CrazyCartoon (a map corresponding to a level in the game)

Xiao Li, corresponding author (lixiaoatswu@139.com)

The rest of the rules in the game are the same as the general tower defense games[5], the goal of CrazyCartoon game system is to try to stop enemies from reaching the center or the camp by building to turrets which shoot at them as they move. Enemies and towers usually have varied abilities, costs, and upgrade prices. When an enemy is defeated, the player earns money or points, which are used to buy or upgrade towers, or upgrade the number of money or points that are earned. Each turret can be placed anywhere, but only the correct strategies can defeat all enemies and win the game.

B. System Architecture

There is the architecture of CrazyCartoon system based on the mobileAR in Fig.2, which is mainly composed of image acquisition, image recognition and tracking, mix scene generation of virtual and actual reality, game logic and human computer interaction, as well as user-defined game strategies modules.

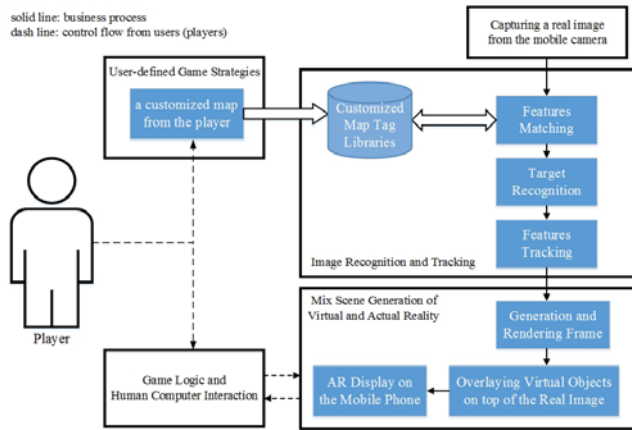


Fig. 2. CrazyCartoon: a User-defined Strategic Tower Defense System with MobileAR

As seen from in Fig. 2, Capturing a real image is work of the mobile camera, and what the player need to do is just that points the camera at the hand-drawn map. Image information can be applied to identify and track as well as the scene generation and blending.

The module of Image Recognition and Tracking addresses to identify the target by comparing data with those in Customized Map Tag Libraries. On the basis of it, the system should keep tracking the identified target during the game playing. Obviously, the robustness of tracking is the key point to bring a perfect game experience for the players.

The main function of Mix Scene Generation module is superimposing the virtual objects on the hand-drawn map accurately and keeping rendering new frame of fusion 3D scene to display on the mobile phone's screen (because the player would change the camera's direction while playing). The player will enjoy the game in this blending scene of reality and virtual with a completely freedom to switch levels or deploy his battlefield.

The function of Game Logic and Human Computer Interaction is not only to support the interaction between game objects and the player but also to give the player ability to

send orders to game. Players can control the game by designing hand-drawn maps and touching operations. While both the interaction between game objects and all the events in game process are disposed by game logic. The interaction between player and system is implemented by three ways below:

- Personal hand-drawn maps are used as the game map instantly, which is also one of the characteristics of this solution.
- The player's view in the game can be controlled by changing the position and orientation of the camera freely.
- Decide the position of the turret. Players can easily build turrets by touching in the 3D Mix Scene.

Finally, the module of user-defined or customized game strategies is mainly to give effective support so that the player can design hand-painted map on his own.

III. PROTOTYPE IMPLEMENTATION AND EVALUATION

A. Development and implementation platform

1) Development Environment

- Hardware: CPU Core i7-3630QM 2.40GHz*4
Memory 8GB Graphic Card NVIDIA GT650M
- OS: Windows7 Ultimate Service Pack 1
- Unity Game Engine
- Qualcomm Vuforia™ SDK
- 3DS MAX and Photoshop

2) Running Environment

- Android phone with a rear camera
- CPU 1.2GHz, Memory 1GB, Memory Card 50M or above
- Android edition 2.3.1 or above

B. Prototype architecture and implementation

1) Prototype architecture

The development and realization of CrazyCartoon is a combination of Qualcomm Vuforia technologies and Unity Technologies game engine[6] [7], and the interaction between them is processed with C# programming.

Based on the ideas discussed in Fig. 2, the CrazyCartoon prototype architecture is shown in Fig. 3.

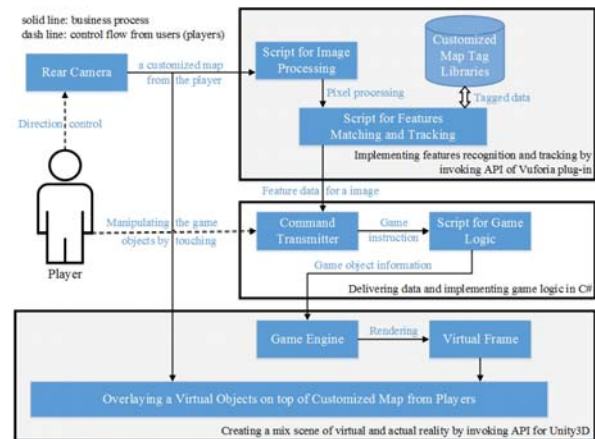


Fig. 3. A Prototype Implementation of CrazyCartoon with MobileAR

By using mobile camera simply to point the hand-drawn map, the image can be acquired and be digitalized by the script for image processing as we as features matching by the script for features matching and tracking with the data established in Customized Map Tag Libraries.

There are lot algorithms for image recognition now. Invoking Vuforia plug-in API is a wise choice for rapid prototype development to generate the tagged data package that can be imported to Unity engine, and finally invoking Unity API to complete virtual objects overlaying and frame rendering.

The data transfer between the Vuforia plug-in and the Unity engine and game logic are established in C # language. Some reusable classes are independently programmed to implement the interaction between the various objects in the game scene as well as the interaction between players and the game.

The following introduces the prototype developed and running on Android platform.

2) prototype implementation

Fig. 4 is the startup screen after CrazyCartoon is loaded.

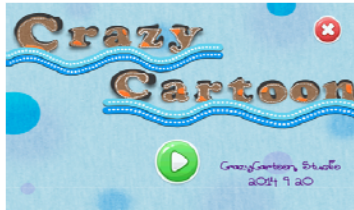


Fig. 4. Startup screen after CrazyCartoon is loaded

If via camera the system did not find any recognizable images, the game enters a paused state (See Fig. 5).



Fig. 5. Game is paused when finds no image

When the system identified the play's map, the game is started and the building needed to be protected is come out at the center (Fig. 6). A box shown in yellow grid on the display screen represents a position that permits the player to built a new turret. It is decided by the player where to build the tower in different situation.



Fig. 6. The yellow line shows when the player is going to build a new turret

Fig. 7 is a screen shot during the game playing. You can observe while the enemies or monsters are approaching in different directions, these turrets which the player has built by himself are automatically attacking them to stop their closing. The interactions between game objects are a typical manifestation of Artificial Intelligent, such as the turret fire at the nearby monsters. By invoking the Vuforia plug-in API, the ongoing tracking of the hand-drawn map is implemented and via command transmitter programmed C# language the data is delivered to Unity engine for game frame's rendering and generation.



Fig. 7. Monsters move toward the house and turrets attack them automatically

When the player defeated all the monsters and the building had not been destroyed, he would be the victor (Fig. 8.). Current game is over at the same time.



Fig. 8. The player win the game after all monsters were defeated

C. Usability testing and evaluation

In the process of software development, the usability testing is an absolutely essential. Of all usability evaluation methods user testing is the most effective one [9][10]. User testing, a type of testing done from an end-user's perspective to determine if the system is easy to use, is applied to the CrazyCartoon prototype. During testing users will try to complete typical tasks while observers watch, listen and takes notes.

Five users [11] joined the test, and one of them is the one who is not interested in computer games at all. A test version of CrazyCartoon, a mobileAR-based tower defense game, is installed on their Android phone with a camera and the two player's hand-drawn maps (shown in Fig. 1.for example) are adopted.

To measure CrazyCartoon system robustness, there are two things we have to do. First of all, whether or not the system can be able to identify two different hand-drawn maps and then switch the corresponding levels. Fig. 9 shows the result when the system recognizes another map different from the one in Fig.6 and a dialog box asking if you want to switch will be displayed.

Secondly, it is crucial to test whether or not the system can be always kept tracking the target in real time while the

camera's changing direction and zooming in or out.



Fig. 9. The dialog box shows a request to switch game-level after identifying another map



a) test one in varied direction



b) test two in varied direction



c) test to zoom out



d) test to zoom in

Fig. 10. CrazyCartoon system keeps tracking in stable and continual way in real time while game playing.

As the four screen shots shown in Fig.10, the system can be always kept tracking the hand-drawn map stably and continually in the whole process of game playing, which proved the system robust.

In short, the results from the end-users have proven that the system can run smoothly on the Android phone and met the preset goals that gave them a great experience, which

verified the correctness and feasibility of the proposed scheme. While the face-to-face talking about the users satisfaction on spot has proven an unique advantage of the MobileAR in supporting game strategy customized. Last but not least the novice loved this game immediately and asked how to make his map.

IV. CONCLUSIONS AND DISCUSSION

This paper presents the design, implementation and evaluation of a tower defense game with mobileAR named CrazyCartoon, which enables players to customize the game strategies freely, such as the level switch and turrets' defensive position by creating a fusion of real (hand-drawn map by a player) and virtual (digital game) scene. The idea, especially importing players' hand-painted map into computer games, on the other hand is a new challenge to players' imagination and creativity, and on the other hand offers a new approach to the study, design and development of game.

To improve the identifiability, players' maps are currently processed in bicolor image (mainly black and white), some algorithms based on computer vision will help to improve the highly freedom of them. It will be the next step to strengthen and expand the function of customized strategy module to make it easy and convenience to use for all players. Furthermore, it is important to design and develop reusable components to optimize the prototype and provide those who are interested in with a practicable solution to develop mobileAR game in a viable and rapid manner as well.

REFERENCES

- [1] Hannu Jaakkola and Bernhard Thalheim, "Architecture-driven modelling methodologies," Proceedings of the 2011 conference on Information Modelling and Knowledge Bases XXII, IOS Press Amsterdam, The Netherlands, pp. 97-116
- [2] Azuma R., "Survey of augmented reality," Teleoperator and Virtual Environments, 1997, 6(4), pp. 355-385
- [3] Tobias H. Hollerer, Steven K. Feiner, "Mobile Augmented Reality," 2004
- [4] ARDefender, <http://www.ardefender.com>, 2014
- [5] Jeannie Novak, "Game Development Essentials: An Introduction", CENGAGE Learning Custom Publishing, 2011
- [6] Vuforia, Qualcomm Incorporated, <https://developer.qualcomm.com/mobile-development/add-advanced-features/augmented-reality-vuforia>, 2014
- [7] Unity, Unity Technologies, <http://unity3d.com/>, 2014
- [8] Karli Watson, Christian Nagel. "Beginning Visual C# 2010," Wiley Publishing, Inc. 2010
- [9] Joseph S. Dumas, Janice C. Redish, "A Practical Guide to Usability Testing," Rev Sub, 1999
- [10] J.L Fu, G Salvendy, L Turley. "Effectiveness of user testing and heuristic evaluation as a function of performance," Behaviour & Information Technology, 2002,21(2), pp. 137-143
- [11] Nielsen, Jakob, and Landauer, Thomas K.: "A mathematical model of the finding of usability problems," Proceedings of ACM INTERCHI'93 Conference (Amsterdam, The Netherlands, 24-29 April 1993), pp. 206-213