Lines of Play: Using Vector Tools to Place Game Objects in Augmented Reality

Jasmine Roberts, John Mernacaj, Suzanne Chambers, Dror Ayalon and Errol King
Google, LLC
New York, New York
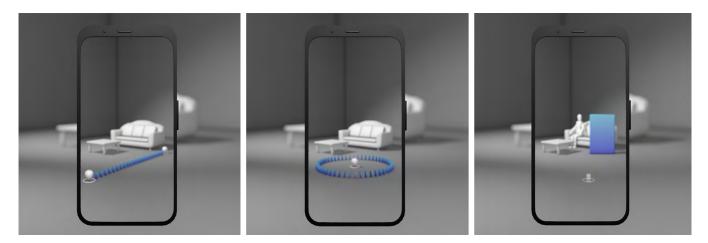{jasmineroberts,mernacaj,schambers,dayalon,errolking}@google.com

Figure 1: Three domino placement modes. *Left.* Line placement, *Middle.* Circle Placement, *Right.* Jumbo Domino placement

ABSTRACT

'Lines of Play' is a domino art application showcasing recent advancements in augmented reality (AR) technology. Unlike previous mobile AR applications that were limited to horizontal and vertical object placement, these game objects interact with real objects in the player's environment with surface-level resolution. Features of the player's environment provide kinematic constraints (on collision) and serve as obstructions both visually (occlusion) and during gameplay. Using single camera depth information, players can use familiar vector drawing tools to rapidly create and modify lines, rings, paths, and turns of domino tiles. We thereby provide a method for placing a plurality of gaming objects by converting raycasted anchor points into paths of game objects. This opens up novel possibilities of authoring tools specially adapted for AR game development or game-integrated level editing.

CCS CONCEPTS

• **Human-centered computing → Mixed / augmented reality**.

KEYWORDS

game design; augmented reality; mobile games; authoring tools

1 INTRODUCTION

Mobile AR benefits from accessibility of hardware like smartphones and tablets. Previously, a primary setback of mobile augmented reality was limited memory and computational capability on-device. Today devices supporting AR are more powerful and less expensive.

Apple's ARKit was released in June 2017; Google's ARCore in March 2018. Both enabled developers to begin creating AR games and applications. These systems allowed for players to place game objects on vertical and horizontal planar surfaces but could not yet distinguish between granular surface-level environmental features thereby restricting the effective gameplay to feature points like specific locations or anchored to an image. Since 2018, developers have been anticipating updates that seamlessly merge real and virtual objects within a player's environment. [1,5]

In December 2019, the ARCore Depth API was announced. Without the use of dedicated hardware like depth sensing cameras (LI-DAR), the ARCore Depth API creates depth maps from motion with

a single camera, and thus works on the multitude of devices that support pre-existing versions of ARCore. [7,10,13]

Furthermore, the 3D depth data assists in surface interaction, path algorithms, realistic physics and occlusion. The depth map determines which object is nearest the player and provides an estimate if the virtual object placed in the same environment will overlap or not, solving an earlier limitation of virtual game objects floating over real objects in the same environment. Depth cues allow for more interactive game mechanics occlusion behind and collision with real-world objects.

2 MOTIVATION AND PREVIOUS WORKS

Even with occlusion handling, there are many technical and design challenges in creating new, usable and useful AR/VR applications. In particular, non-technical designers and end-users encounter a lack of tools to quickly prototype and test new AR/VR user experiences.

Recent solutions like Facebook's Spark AR Studio or Snapchat's Lens Studio allow designers to select from libraries of trackers, 2D/3D objects, and masks interactive to create platform-specific shareable camera effects. [2,11] These both have support for visual and modular programming to add logic, animations, and interactivity. More complex tools like user input and UI elements and explicit interactions are noticeably absent from augmented reality authoring tools. [8]

We aimed to create a world building tool that gamifies real-world environments and allows users to interact with those game objects. We ultimately decided on dominoes because they are primitive objects that can be easily replaced by developers by more complex objects. The difficulty of domino art creation in the real world is players are limited by the number of physical dominoes, precision, dexterity, and patience. 'Lines of Play' allows any player to quickly create domino paths using a select tooling modes. The ability to create layouts and placement of objects from within the space quickly becomes very useful and is extensible to other creative applications.

Examples of AR world-building games like Minecraft Earth rely on location data and GPS to connect the game objects with the real-world. In the game, the player can collect blocks, mobs and items. After collection, the user can build on buildplates– small designated augmented reality areas where players can build, and place blocks like in the traditional Minecraft game. [9] Since 'Lines of Play' is neither marker-based nor tethered to a location, players can build on any detected surface.

3 SYSTEM DESIGN

'Lines of Play' was originally built with Unity 2018.4.17f1 (long-term support release) using the ARCore platform's new Depth API package and released on the Play Store [3,4]. It was later updated in Unity 2019.4.5f1 using AR Foundation– a preview Unity package unifying core features from ARKit, ARCore, Magic Leap and HoloLens to enable multiplatform builds. [12]

We created modes enabling players to rapidly generate paths of dominoes. In vector graphics programs, paths are central to the compositions of various objects. They are the fundamental lines comprising objects. Similarly, in domino art, paths are the base units of constructing domino fields. In our demo, we use analogues to common vector drawing tools (line, circles, rectangle, freeform drawing) to designate a projection path spawning clones of domino game objects. (Figure 1)

There are five different modes of domino path placement:

- **Line placement** Players can draw a path by specifying two points A and B and generate a line of dominoes.
- **Circle placement** Players can generate a ring of dominoes by specifying a center point of the circle and a point on its circumference.
- **Rectangle placement** Players can specify two points A and B, that comprise the diagonal of a rectangle.
- **Draw to place** Players can freeform draw on surfaces and a path of equidistant dominoes will generate.
- **Jumbo domino** Players can place individual 4' tall dominoes.

Both the drawing and shapes functions allow player to preview the design before placing the dominoes. Placement balloons are 3D proxies for anchor points and designate the length, radius, and diagonals of the line, circle and rectangle placement tools respectively.

At application start, once a horizontal plane has been recognized, a phantom domino tile is raycast onto the ground plane to provide a preview of where the domino will appear when the user taps the screen. Players can place multiple domino tiles using this method or can toggle between pre- defined path placement modes in the application's interface that distribute equidistant domino clones along a path.

3.1 Development

A barrier in augmented reality development is there are many system components and functionalities to work properly. A common challenge developers face is staying up-to-date with the latest upgraded libraries and SDKs. Fixing bugs, implementing new functionalities and expanding user capabilities become semiannual. By open-sourcing a complete game sample, we have provided augmented reality game developers sufficient support to reproduce their own applications.

The `MainController.cs` script handles switching between different modes Domino, Jumbo Domino, Remove, Topple and changing colors. It also holds references to most of the objects in the scene. `TwoPointSpawner.cs` is responsible for switching between the mode menu and also generating the shapes. `BalloonPlacing.cs` is responsible for previewing the path and when the user completes all points for the shape it calls function in TwoPointSpawner script to generate dominoes for that particular path.

Similarly, `DominoPlacing.cs` is used for placing dominoes individually on the ground. For previewing and handling large number of dominoes there is a pooling technique which is being used by `ObjectPooler.cs`. As the name suggests `UndoRedoManager.cs` is responsible for Undo and Redo functionality. Drawing freeform shapes tracked surfaces is handled by the `Paintable.cs` script. Lastly, the logic for deleting and toppling dominoes handled in `DeleteDominoes.cs` and `TaptoForce.cs` respectively. [4]

Lines of Play extends baseline demo samples to incorporate physics, UI, sound and interaction methods

4 CONCLUSIONS AND FUTURE WORK

Overcoming the drawbacks of the previous generation of AR software development kits, recent improvements have the capacity to blend the digital and physical worlds on a multitude of mobile devices. We present 'Lines of Play' as template for game developers to start building with the ARCore Depth API. Currently, 'Lines of Play' uses the standard vector functions to generate game object paths of domino tiles.

A technical advancement of "Lines of Play" is the departure from grid-centered 3D augmented world building applications like Minecraft Earth which is expansive yet limited to cuboid structures. Introducing circular paths and spline placement of 3D game objects into augmented reality unlocks what players can experiment with organic path placement.

In future iterations, players could place non-visual and intelligent game objects, have persistent and sharable creations through use of Cloud Anchors.

REFERENCES

[1] Apple Inc. 2020. *ARKit - Augmented Reality*. Retrieved July 2, 2020 from https://developer.apple.com/augmented-reality/arkit/
[2] Facebook. 2020. *Spark AR Updates*. Retrieved July, 6, 2020 from https://sparkar.facebook.com/ar-studio/
[3] Google Creative Lab. 2020. *Lines of Play* (Version 1.0) [Mobile application software]. Retrieved July 15, 2020 from https://play.google.com/store/apps/
[4] Google Creative Lab. 2020. *Lines of Play*. Retrieved July 3, 2020 from https://github.com/googlecreativelab/lines-of-play
[5] Google. 2020. *ARCore Unity SDK*. https://github.com/google-ar/arcore-unity-sdk
[6] Google. 2020. *Build new augmented reality experiences that seamlessly blend the digital and physical worlds*. Retrieved July 2, 2020 from https://developers.google.com/ar
[7] Izadi, Shahram (2019, Dec 9). Blending Realities with the ARCore Depth API. *Google Developer Blog*. https://developers.googleblog.com/2019/12/blending-realities-with-arcore-depth-api.html.
[8] M. Nebeling and M. Speicher, *The Trouble with Augmented Reality/Virtual Reality Authoring Tools* 2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct), Munich, Germany, 2018, pp. 333-337, doi: 10.1109/ISMAR-Adjunct.2018.00098.
[9] Mojang. 2020. *Minecraft Earth* (Early Access) [Mobile application software]. Retrieved July 15, 2020 from https://play.google.com/store/apps/
[10] Ruofei Du, Eric Turner, Maksym Dzitsiuk, Luca Prasso, Ivo Duarte, Jason Dourgarian, Joao Afonso, Jose Pascoal, Josh Gladstone, Nuno Cruces, Shahram Izadi, Adarsh Kowdle, Konstantine Tsotsos, and David Kim. 2020. *DepthLab: Real-Time 3D Interaction With Depth Maps for Mobile Augmented Reality*. Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology, pp. 15.
[11] Snap Inc. 2020. *Snap Lens Studio* 3.1 Retrieved from https://lensstudio.snapchat.com
[12] Unity Technologies. 2020. *AR Foundation*. Retrieved July 15, 2020 from https://unity.com/unity/features/arfoundation
[13] Valentin, J., Kowdle, A., Barron, J.T., Wadhwa, N.,Dzitsiuk, M., Schoenberg, M., Verma, V., Csaszar, A., Turner, E., Dryanovski, I. and Afonso, J., 2018. *Depth from motion for smartphone AR*. ACM Transactions on Graphics (TOG), 37(6), pp.1-19.