Augmented View for Tabletop Miniatures

This application is an augmented reality visual aide for miniature gaming on a tabletop. Some tabletop games are based on users moving miniatures on realistic miniaturized terrain and objects. This is often an expensive option and hobby.

With this app, users would be able to see constructs, objects, background, and effects that are fabricated in this augmented reality visual portal (i.e. phone, tablet, glasses). Players in the same physical room viewing the same set of data would be able to see each other's actions.

Here's the usage scenario: We have a group of players physically in a room playing in the same role-playing tabletop miniature game. There are physical miniatures that are used in the game. There might just be a paper-based map on the table-top. The game play would be exactly like existing RPG games. However, the app add additional visual to the game.

What does this application provide:

1. The users use the app to scan in their miniatures prior to game start. This allows the app to know all the "moving pieces".
2. The reference setups the game by configuring a database on a host server.
3. We use tokens with barcodes to represent monsters, terrains, buildings, objects, etc. The users would be able to see these as augmented objects instead of the real physical tokens with barcode) The app basically identifies an object and display the object in the augmented reality display. The object being displayed would be something pre-loaded from the server that is specific for the current game.
4. The application can be configured for each player uniquely to perform certain "animation" tasks (e.g. cast a spell, shoot an arrow, charge, parry, etc.)
5. Each player can see other player's actions and special effects
6. The reference (game master) can activate object at predefined locations. This is TBD on how this would be done, but a simple way is to allow the reference to access the server using a different interface.

Tell us how you plan on bringing it to life.

This project is a concept stage. We have two part-time developers for this project. We plan to open source this project once it is done. We want finer design and code control for the initial release. This allows the team to learn and keep our own pace. However, we do want to share our framework and results with the development community such that this application can be useful beyond this one single project.

The plan:

1. December-January. Iron out out all the screens and possible user interactions. Setup automated test such that we can use test-driven development for this project. Ensure we have proper unit test framework for continuous integration. Want to have the design completed and reviewed. (Could use Google's help to review our design. For example, we can use ML kit to help with barcode recognition and decoding since tokens are laid flat; miniature face/hand recognition to allow of image-warping for certain easily doable animations. Use ML kit to help track when the physical miniature are moved by a player. Create extendable and mod-capable interface such that users can add on this infrastructure. Could use some feedback and recommendations here.)
2. February-April. Code the application and tests. (Could use Google's help with reviewing the implemented solution.)
3. March-April. Create a server that can support this application. (Could use help with interaction between users and server such as authentication, privacy, etc.)
4. April. Evaluate and refine the product.

Having Google's AI logic in the application would help us on the following:

1. Leverage any Google augmented reality support.
2. Use ML Kit to read slanted barcode.
3. Use the locations of barcodes and their orientation from all users in the same game to stitch a unify map. This would allow a user to still see the who picture even if she might not have visibility to all the pieces on the table top.
4. Use ML Kit to help with tracking object movement.
5. Use ML Kit to help with identifying parts on scanned miniatures to allow the application to do simple animation. For example, a wizard shots lightning from his hand. This might be the default behavior unless the user specifically changes to a different location (e.g. top of staff).

Tell us about you.

I have 20+ years of embedded software development experience for commercial consumer products. I have solely written a simple Android application for a class that I took a few years ago (https://youtu.be/u-eLm8dGd0E). I'm not a complete stranger to Android app development, but I only consider myself as an amatuer Android app developer.

Next steps.

- Be sure to include this cover letter in your GitHub repository
- Your GitHub repository should be tagged #AndroidDevChallenge
- Don't forget to include other items in your GitHub repository to help us evaluate your submission; you can include prior projects you've worked on, sample code you've already built for this project, or anything else you think could be helpful in evaluating your concept and your ability to build it
- **The final step is to fill out this form to officially submit your proposal.**