Team Rocket Project Writeup

Aalhad Patankar, Anuraag Advani, Min Fan, Tianyu Yang.

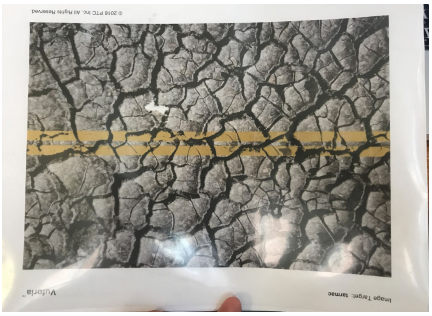
ap3526, ada2161, mf3084, ty2345

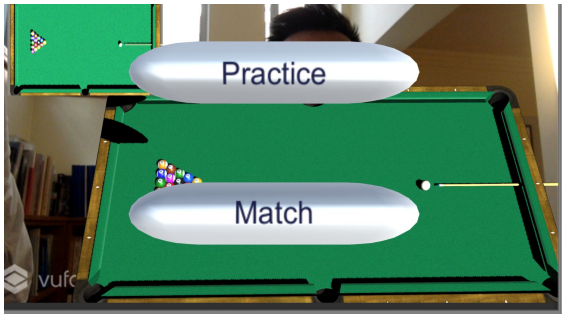
Project: Real Pool

Components

Our game has the following components:

Ground image target: This image target contains the augmented game board object, containing the pool table, pool ball, cue, etc. The image target used is the Tarmac provided in Assignment 3.





Wand image target: This image target contains an augmented cube at its tip which acts as the wand in our game, which can be used for hitting 3D buttons and picking and placing objects. The image target used is the Lego image target provided in Assignment 3.

Toolbar image target: This image target contains virtual buttons that can be clicked for several special actions, such as launching the “ghost ball” (discussed in further sections) and switching between camera views such as cue view and minimap.

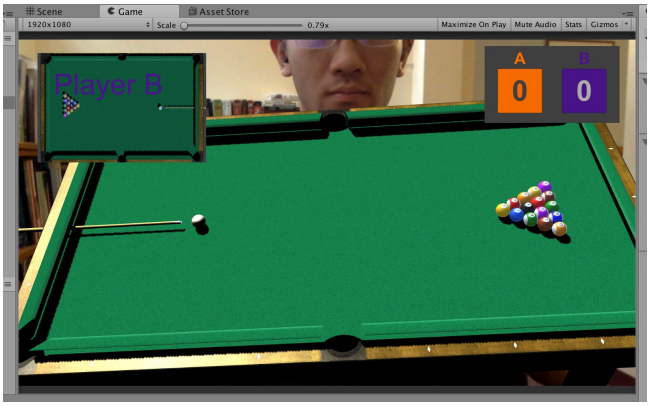
Viewer mobile device + Google Cardboard: This device will be used to view the scene, integrated with Google Cardboard to provide stereoscopic vision. It will not have any buttons associated with it and will be used purely to view the scene.

Controller mobile device: This device will be used as the physical controller to execute shots. It will have a trigger button that can be pressed when executing a rotation or hitting action on the cue, and a button that can be pressed to toggle between rotation and hitting.

Gameplay

Upon launching the game, the user will be presented with options to enter practice mode or play mode through two 3D buttons. The user may then select either of the buttons using the wand to enter the corresponding mode.

In play mode, two players can play a standard game of pool. Turns will be alternated when the white ball stops moving after a hit. Hitting can be achieved using the controller mobile device by toggling to the appropriate mode using the toggle mode button, and by holding the trigger button and rotating the phone along its longitudinal axis to apply desired effect to cue.



The following simplified rules apply to the game:

The score can be seen when looking away from the table in the play mode.

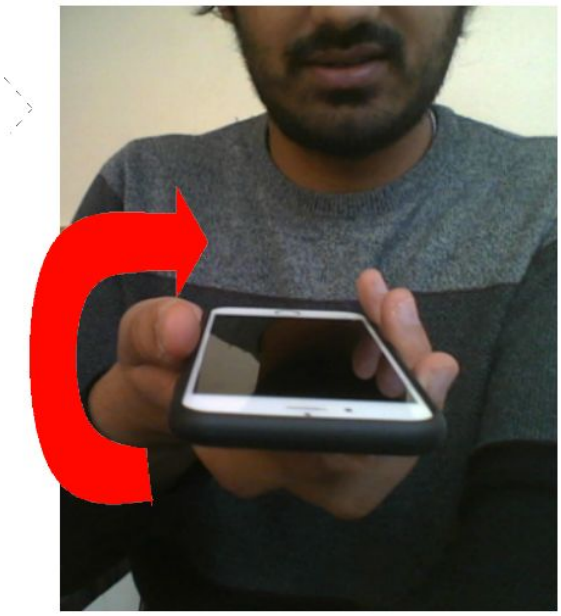
In practice mode, the user may pick and place the balls as the user likes, and execute shots. <Fill in more info here>

Interaction

A large area of focus for us was to determine the interaction technique to hit the ball. At first, we used the wand to perform the hit, where moving the wand up and down would be used for moving the cue. However, we found it was very difficult to hit using this interaction mode as it would move wildly and unrestricted.

To better fulfill the error prevention heuristic, we limited the motion of the cue to angularly around the ball in the movement mode, and radially towards and away from the ball in the hitting mode such that no matter how much the cue is moved, it will always strike the white ball at its center when performing a hit. We found this made it much easier to prevent erroneous hits.

However, the interaction was still far too different from the real haptic feedback from real pool. Thus, to better fulfill the match between system and real world, we moved to using a mobile controller to control the cue. This way, the user could rotate the mobile controller with the force the user wants to apply to the cue, and thus the force of the hit would be directly tied with the force the user applies on his or her rotation of the phone. We attempted to perform the hit by moving the phone controller and mapping its translation force to the hit, but after research, we discovered it is not possible to track this information.

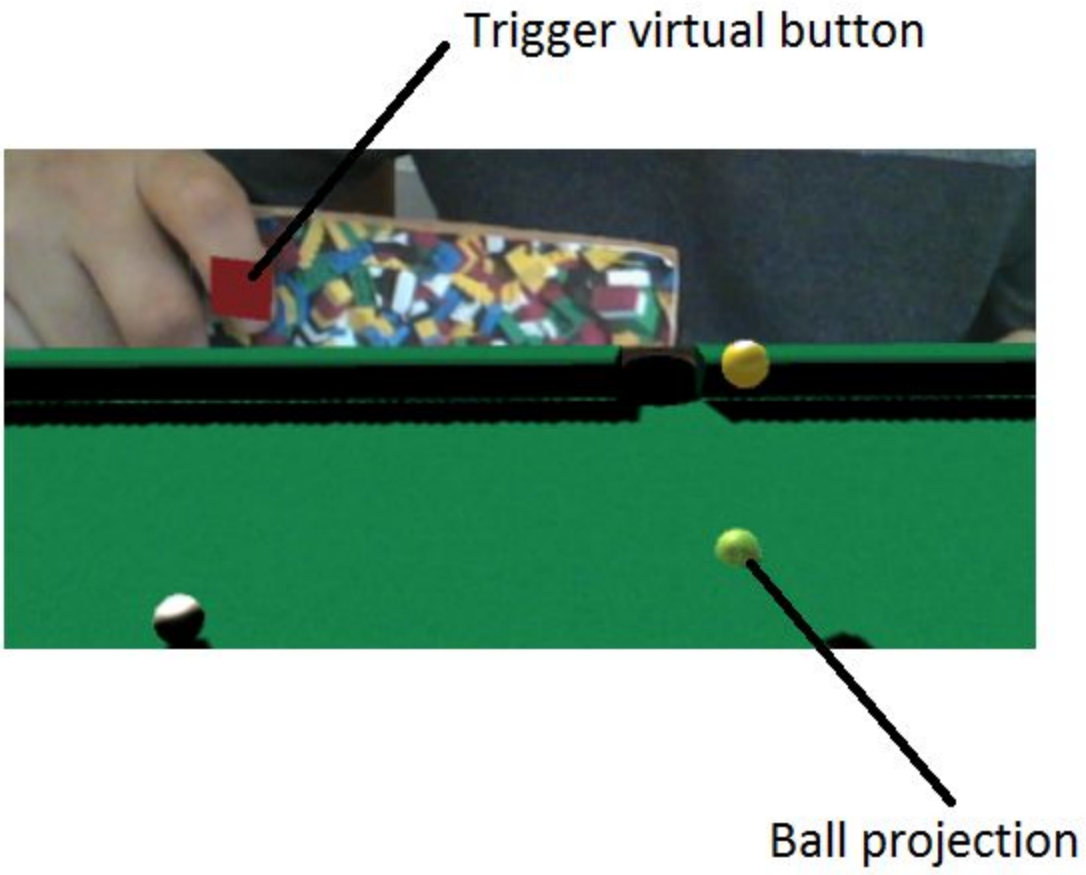


Selection and manipulation

We provided selection and manipulation through the form of allowing users to pick and place any ball in practice mode to arrange the perfect shot. To pick and place, the user can use the augmented cube end of the wand to make contact with the object to move to select it. The user can click the virtual button to begin moving, which will be applied

until the user un-presses the virtual button. To support the visibility of system status heuristic, the virtual button changes color when an object is being actively moved.

Additionally, to support the visibility of system status and error prevention heuristics, we also placed a projection of the ball being moved on the plane of the pool table surface during movement to show users where the ball will be placed upon release of the trigger button. This will help users better envision where on the pool table surface their selected ball will be placed, despite its height from the pool table surface.



Wayfinding and travel

To support wayfinding in our game to allow users to better plan their shots, we added two features. The first is a “ghost ball” functionality, which will show users a projection of their current trajectory if they were to execute their current shot. This method shows the trajectory of a ghost ball, which interacts with the walls and table the same way as the

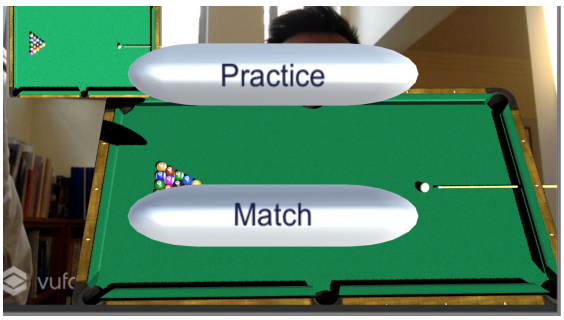
real white ball, but passes through all other balls. The reason we designed the ghost ball to go through other balls is to provide users a challenge, as otherwise, the game would be too easy. The ghost ball was also designed to satisfy the error prevention and system visibility heuristics.

Additionally, we provided a minimap as well as a cue perspective mode, in which the user can switch perspectives to that of the cue stick, and travel with the cue stick.

Menus and controls

Regarding the user interactions, we have 3 types of buttons.

We have 3D augmented buttons for main menus, for example selecting game mode, exit the match and restart it. When user wants to interact the item, he/she would use the wand and collide it with the menu items. Besides the intuitiveness, we use this type of button because it was the most reliable method of controlling the virtual space. The virtual buttons were too dependent on the environmental factors like light condition.



We also have some virtual buttons for auxiliary functionalities. Those options need to be displayed during the game play, so we can't implement all of them as 3D augmented buttons anchored to AR camera as it block too much screen space. Those buttons includes shoot simulating and change minimap view.

We also have a physical button on the controller device. Controller device has 3 mode: inactive, change hitting direction and the actual hitting mode. The button is used to iterate through them. We put the button on the controller itself because they were tied with the interactions.

For panels to inform users the system status, we have a scoreboard which will be displayed only when the user looks away from the table. We made this decision for preserve the most of the display space when the users are playing. We also have a panel that displays which player is playing.

Heuristic Analysis

There are several components of our design intended to support the visibility of system status. Firstly, on the controller phone, the text on the trigger button indicates what mode the user is currently in (e.g. rotation, hitting or neither). Additionally, the trigger button on the wand changes color from white to red when a ball is being moved to indicate that the movement action is occurring. Additionally, when the ball is being moved, a projection of the ball on the table is displayed on the table to indicate where the ball would be placed if the button were released at that position, as discussed above. The minimap also provides further information of the system status by showing a view of the whole table to the user.

The interaction technique we selected was done to make a tight match between our system and the real world, as discussed previously in the interaction section. The size of the game objects, such as cue, balls and the table were designed to reflect a real pool table. Additionally, we added haptic feedback by vibrating the phone upon a successful hit of the cue to the white ball.

To support user control and freedom, we allowed users to exit out of play mode and practice mode and switch between them at any point. Additionally, we provided the user the ability to place a ball wherever he or she likes in the practice mode to practice any shots.

To support consistency and standards, we ensured our game language as well as the app language matches game and phone conventions.

To support error prevention, we restricted the cue motion as discussed earlier to being anchored on the center of the white ball, to make it easier for users to hit the white ball. Additionally, we dynamically rendered UI buttons depending on what mode the user was in to disable users from activating illegal actions. Finally, we provided help features

such as the ghost ball and the minimap view mode as well as the cue view mode to provide users with support in planning their shots.

To minimize recall, we made our menu options available at all times with simple names that indicate their function, and displayed the score and current player at all times when the user looks away from the pool table. We designed this mechanism this way as to not occlude the pool table view.

To provide flexibility and efficiency of use, we abstracted the ghost ball and view switch functionalities to a toolbar image target, such that users who desired that functionality could have that available, but other users need not occlude their screen space with unnecessary options. We also supported a practice mode for more advanced users to design tricky shots and practice.

Our interface design was built keeping minimalism in mind. For this reason, we chose to only display game information such as score and current player when the player looks away from the pool table target, in order to keep the pool table view unoccluded when the player is attempting a hit. Additionally, we reduced the total number of buttons to only necessary buttons, and dynamically rendered only the necessary buttons depending on what state the system is in. Additionally, we used pre-created Unity pool assets for optimal aesthetic touches on our game objects. Also to preserve minimalist design, we minimized the number total touch screen buttons to just the trigger button on the controller device.

To help users recover from errors, we supported a reset and an exit button to allow users a force reset if a mistake was made.