Smart Snooker Survey and Analysis Stage

Douglas O'Hanlon

Supervisor: Phil McMinn

This report is submitted in partial fulfilment of the requirement for the degree of BSc in Computer Science

in the

School of Computer Science

December 9, 2024

Declaration

"All sentences or passages quoted in this report from other people's work have been specifically acknowledged by clear cross-referencing to author, work and page(s). Any illustrations that are not the work of the author of this report have been used with the explicit permission of the originator and are specifically acknowledged. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this project and the degree examination as a whole."

Name: Douglas O'Hanlon

Signature: Douglas O'Hanlon

Date: December 9, 2024

Abstract

Augmented Reality is a technology that can be used to overlay information or objects into a view of the real world. Advances in mobile technology now means that most people have a smartphone in their pocket, capable of seamlessly viewing an augmented realty scene. SmartSnooker aims to explore how a mobile AR system can be used to create an accessible, user friendly system, by helping novice billiard players select different shots and understand successful trajectories.

Billiards is a centuries old group of cue sports, including snooker and pool, commonly found in social venues. SmartSnooker aims to make the game easier to learn for new players, while also giving users an easy introduction to the world of AR.

Contents

Chapter 1

Introduction

1.1 Brief Overview

SmartSnooker aims to investigate and apply augmented reality to help novice players improve at cue sports such as pool and snooker). Augmented reality (AR) is a technology that takes a view of the real world and overlays virtual, aligned, computer-generated content to the video output [23]. This project will look at technologies for AR app development and how cue sports players learn, to create a mobile application novices find useful.

There are existing AR technologies for aiding billiard shots, such as Cassapa [26], IQ Billiards [13] and MagixPool [30], however these technologies require camera and projection system mounted above the table. A novice must visit a venue with this expensive hardware to utilise them. SmartSnooker aims to make this technology much more accessible, by only requiring a smartphone and internet connection. A more comprehensive review of cue sport AR systems can be found in 2.2.4.

1.2 Challenges

The core problem of the app is converting a video feed of a pool table, to a model of a pool table, while also determining the position of the balls on the table. This will require using computer vision techniques to detect and transform balls from image-space to world-space. This is explored in 2.2.3. Once a model of the table and position of balls has been created, the application must use physics and rules of the game to generate potential shot trajectories. The challenge with this is calculating many potential shot trajectories in real-time, as simulating a billiards shot can be much more resource intensive than expected (see 2.3.1). The development of fast, sufficiently accurate shot projections will require experimentation with existing physics toolkits, or creating more simple trajectory models from scratch.

An easy to overlook challenge of creating the app, is making sure the system is actually useful to players and not just a demonstration of AR. This paper looks at the skills a billiards player must learn and how this translates into the app's requirements in 2.6 and 3.1.

The main goal of the application is to produce something that is accessible and easy to use. So the application will also make use of an animated interactive tutor to explain rules and tips.

1.3 Challenges Facing New Billiard Players

The game of Billiards has been played for centuries, with evidence of the game being played on tables indoors dating back to the 17th century [6]. Today the game is a staple of bars and social venues across the world. [29] found that 5% of surveyed adults in Scotland played snooker, billiards or pool in the last four weeks. This number had a strong bias towards men, as 9% of men played within this period compared to just 1% of women.

[32] analyses the gender power-dynamics of billiards. [32] assesses how the aesthetic of a billiards hall itself reinforces the idea of a male space, as well as the nuanced forms of sexism and exclusion that operate in the environment. While it is difficult for a mobile application to change the aesthetic or social dynamic of a venue, it could allow novice women to get into the sport, by offering an alternative to experiencing possible sexism and exclusion while learning with men.

This is just one example of how certain groups of people can be excluded from billiard participation. An accessible app helping novices progress to a higher standard, would allow more people to participate in the sport, without relying on existing players that may not offer encouraging help.

1.4 Justifying Augmented Reality as a Solution

The use of AR to help improve billiard players was studied by [15], where wearable computers were used to suggest shots and improve targeting. This was largely held back due to unnatural, slow smooth motion of the head required for the system to work and delays in the output. A mobile app could improve on some of these shortcomings, as it not bound by the motion constraints of wearable technology. Delays could also be improved upon since today's computers are much more powerful than those of 1997. AR technologies have already been used to improve performance and engagement in other sports. For instance, [16] claims their AR climbing wall was well received and provided users with a greater diversity of challenges than training without the technology.

1.5 Overview of the Report

SmartSnooker is a software engineering project, with a strong experimental aspect. So much of development will include experimenting with different AR, computer vision and physics toolkits/methods. The Literary review in Chapter 2 will provide an overview on the background material for AR, computer vision and the game of billiards, as well as toolkits and methodologies to be experimented with to find a solution to the problem.

In Chapter 3 I put this research into context to create requirements for the application. Finally, Chapter 4 gives a brief overview on progress so far and rough plan of the project.

Chapter 2

Literature Survey

2.1 Introduction

This chapter is a survey of the background literature and concepts/technologies that could be used to make an augmented reality (AR) billiards app. This includes augmented reality (how to overlay graphics into the real world), computer vision (identifying objects and their positions from the real world) and modelling a game of billiards (virtually representing billiard rules and physics). Plus, a review on how billiard players improve at the game. This literary survey and the available technologies in it will inform the requirements and design of SmartSnooker.

2.2 Augmented Reality

2.2.1 reality-virtuality continuum

In 1994, [22] introduced the concept of a reality-virtuality (RV) continuum, which ranges from the real world at one end and a completely virtual environment at the other. Augmented reality (AR) can be related to a point along the RV Continuum, within a bracket of mixed reality [22], the space between completely virtual and completely real. AR achieves this by taking a view of the real world and adding virtual, aligned, computer-generated content [23].

[22] further breaks down AR into two categories: see-through AR displays, where the user can see directly through the display medium to the real world, and monitor-based AR displays. I am most interested in monitor-based AR displays, which overlay graphics onto live or stored video, as this technology can be implemented as a mobile app.

2.2.2 Maker-based and Marker-less systems

An AR system can be decomposed into three key steps: recognition, tracking and mixing [3]. AR systems can be implemented using either a marker-based or marker-less approach [3]. Marker-based systems require a real world reference point, often in the form of a physical QR code, whereas Marker-less systems use other information, such as accelerometer, compass and location data to infer the position of the device within an environment [3]. Marker-less systems can be further broken down into feature-based and model-based [7]. In which model-based systems track 2D or 3D models of edges and points and use a projection of this model to the camera image, to find the camera pose [33]. Feature-based systems try to find a camera pose by finding a projection between 2D features and 3D coordinates and finding a minimum distance measurement between the observed 2D image and the projection of the 3D coordinates [33].

[9] found that Marker-based systems are less complex, more accurate and more stable than Marker-less systems.

2.2.3 Transforming Camera and World Space Coordinates

These edges and points of edges and points can be transformed into 2D screen space. [1] solves the camera pose problem for point and line constraints. A set of 3D world space points $\mathbf{p}$, can be translated to a corresponding set of 3D camera space points $\mathbf{q}$ using:

$$\mathbf{q}_i = R\mathbf{p}_i + T \tag{2.1}$$

as provided by [1]. Where $R = (r_1^t, r_2^t, r_3^t)^t$ and $T = (t_1^t Z t_2^t, t_3^t)^t$ are rotations and translation matrices. We can then use the collinearity equation to convert points in $\mathbf{p}_i$ to image points in $\mathbf{g}_i = (u_i, v_i, 1)^t$, the normalised image plane.

$$\mathbf{g}_i = \frac{1}{\mathbf{r}_3^t\mathbf{p}_i + t_z}(R\mathbf{p}_i + T) \tag{2.2}$$

as provided by [1].

When trying to minimise the error between feature coordinates and image space coordinates (in feature based tracking), we define the image space error $\mathbf{E}_i^p$

$$E_i^p = \sqrt{\left(\hat{u}_i - \frac{\mathbf{r}_1^t\mathbf{p}_i + t_x}{\mathbf{r}_3^t\mathbf{p}_i + t_z}\right)^2 + \left(\hat{v}_i - \frac{\mathbf{r}_2^t\mathbf{p}_i + t_y}{\mathbf{r}_3^t\mathbf{p}_i + t_z}\right)^2} \tag{2.3}$$

as provided by [1]. The projection with the smallest error is the most accurate world-camera space transform.

2.2.4 Relevant Augmented Reality works

This section looks at existing and past AR billiards systems.

Billiard Radar

Billiard Radar [10], is a discontinued app that offered augmented reality shot guidance. There is little information on this app, other than a web page on the developer's website, describing it as "the very first augmented reality mobile app that assists you playing billiards". The app itself describes the setup process often as "a bit tricky" [10, Screenshot 4] and often must be repeated if lighting changes. No dates of the app's release or discontinuation are mentioned, but on Android it was only supported up to Android OS 2.2 [10], which was released in 2010 [12].

Mobile AR Toolkits

So the release BilliardRadar was a very early smartphone AR application, released before Apple and android released their own AR development platforms: ARKit in 2017 and ARCore in 2018 [25]. ARKit provides realtime position data without the need for calibration. In an experiment, [11] concluded ARKit was acceptable for use in an educational science experiment, after dropping an iPhone 1.7 meters and using ARKit's position data to calculate the acceleration due to gravity. ARKit's data was used to calculate an acceleration of $9.74ms^{-2}$, 0.6% off the theoretical value of $9.8ms^{-2}$ [11].

In [8] Cervenak and Masek describe the challenges of locating a mobile indoors, due to an absence of Global Navigation Satellite Systems. They outline how to map points in ARKit, which they report to be very accurate, but inconsistent in poor lighting. It is possible BilliardRadar may have been more successful if it was made with the ease and accuracy of ARKit or ARCore.

More Recent Billiard AR Apps

There are existing billiard helper apps that use the device's camera, such as Drillroom AI [24], however this is used to detect balls getting potted for accuracy statistics, rather than guiding shots.

There are existing technologies to project potential billiard shots, however they are not monitor-based AR systems. Cassapa [26], uses a mounted camera and a physical projector to project lines onto a billiards table in real life. Commercial projector-based billiard AR systems also exist in the form of IQ Billiards [13], which offers a trajectory projection training mode and MagixPool [30], a projector which adds AR games and effects to pool tables. These systems require a dedicated, overhead camera and projector system, which mean they are only accessible for people who can travel to a venue with the system installed.

2.3 Modelling Billiard Physics

The game model must include a physics model for simulation of potential shots, and a method of modelling or teaching rules.

2.3.1 Pooltool - An Open Source Billiards Simulator

The quickest way to implement the physics of billiards may be to use an existing API. One such simulator API is include within PoolTool [20], an open source billiards simulator published in the Journal of Open Source Software. At a high level, PoolTool can be played as a 3D simulator for billiard games, however it also provides an API with functions to simulate shots and update a model of the game as a result. A possible solution for implementing billiard physics into an AR app, would be to use object detection from the camera input and translate these into PoolTool objects. The documentation suggests this should be possible, as the API provides functionality to set the dimensions and positions of objects such as balls and tables [17, API Reference]. In addition, PoolTool comes with the ability to model entire games on a shot-by-shot basis, with rule sets for both pool and snooker [17, API Reference]. This could allow an AR APP to provide tips based on the current match's history and provide analysis of shots over time. The main problem with PoolTool, may be trying to simulate many potential shot trajectories in real-time to suggest them to users. This is because PoolTool's physics is very in depth, considering everything from ball spin to ball-air interactions [18]. PoolTool uses a time-evolution algorithm, which requires the calculation of 5000-25000 candidate time until event calculations per shot [18].

While physics of this depth may increase the accuracy of shot predictions, generating potential shots using a brute-force method or similar will require more processing power, which may cause delays in outputting shot suggestions. This is not very useful in a real-time environment. In addition, the accuracy of this physics is dependent on how accurately the position of the real-world balls can be found. It is expected that the position of the balls will have some error, so physics of this depth would be both impractical and unnecessary. Pooltool does offer support for custom physics models [17, API Reference], however this would defeat the purpose of using it to simulate the physics. It may still be possible to use PoolTool to predict shots, however a a way of selecting a small number of candidate shots to simulate would be needed. PoolTool could still be useful to track the state of a game if its physics are not used.

2.3.2 Using Custom Physics

This section explores how a more simplistic custom physics model could be used.

Augmented Reality Billiards Assistant Research Project [21], a student design project researching the possibility of AR billiard shot overlay system, decided the physics of ball spin should not be included if such a project was going to be developed over the course of one semester. This is roughly the amount of time available to develop SmartSnooker, so a project of this scope should ignore ball spin.

One projector system, An Augmented Reality System to Assist Inexperience pool players [28], only calculates the trajectory of the cue ball until it collides with another ball, meaning only collisions with the cushion are calculated using:

$$o = 2n(l \cdot n) - l. \tag{2.4}$$

As stated in [28]. Where n is a 2D vector of the normal to the cushion and l is the vector in the reverse direction of the cue ball movement. This makes calculating trajectories much easier, however its functionality is limited.

If we simplify the model to only calculate basic trajectories without a time evolution algorithm, we only need to model shots involving ball-cushion collisions and one ball-ball collision. This could be used to model the trajectory of a cue ball hitting a stationary (target) ball and the target ball's trajectory after the collision. This assumes the stationary ball does not collide with any other balls after the collision, which is beyond the scope of easy shots for novices. In The Physics of Pool/Billiards [19], Keifl states that for a ball interacting with a stationary ball, the new trajectory of the stationary ball is parallel to the line connecting the two balls [19].

Using a 2D coordinate system, let's label the position of the moving ball instantaneously before a collision (the cue ball) $\vec{A}$ and the initially stationary ball (target ball) $\vec{B}$. This means we could simply calculate the trajectory of the initially stationary ball as $\vec{AB} = \vec{B} - \vec{A}$ normalising $\vec{AB}$ to a unit vector for the direction we get:

$$\hat{AB} = \frac{\vec{B} - \vec{A}}{|\vec{B} - \vec{A}|} \tag{2.5}$$

We are ignoring the displacement, velocity and angular velocity for the sake of a simpler, trajectory-based model.

This approach would be much easier to implement and would be much less resource intensive than a time-evolution algorithm, every single candidate event is not being considered at each time step. We only have to consider the cue ball, target ball and cushions. Resources and complexity are also saved by not modelling spin, friction or velocity. This means shots are not modelled over time, so shot predictions will not consider the target or cue ball colliding with any other balls after the initial collision with the cue ball. So when selecting possible shots, the system will also need a way of determining if a ball is along the trajectory of the cue ball or target ball. Another drawback of this model is a novice will not receive information on how hard to hit the ball.

2.4 Computer Vision

[14] describes computer vision as "the host of techniques to acquire, process, analyse, and understand complex higher-dimensional data from our environment for scientific and technical exploration". A full review of the background theory of computer vision is beyond the scope of this project, hence I will focus on computer vision toolkits that could be used in the project. The role of computer vision in Smartsnooker is to detect the locations of the table, pockets and balls from a camera input. Once detected, the system should be able to transform the position of the balls in the camera space to a 2D model of the table (or PoolTool model).

There are existing open pool table object detection APIs. One such dataset is listed on the Roboflow computer vision platform by a user named Joy Wolves [31]. This API can output the position and size of balls and pockets in terms of pixels from a picture input. The model can be used on IOS, using a hosted API [31, API Docs]. This means a device with internet connection could sent a photo of the static balls and wait for a response with the positions of balls. Alternatively the API can be hosted on device, but this is not compatible with IOS [31, API Docs]. This means for a mobile app to use this API, it may have to be statically mounted, for instance on a tripod in order to work. Hough transforms can also be used to find the position of spheres, Shcmidt[27] implemented this on a mobile phone using OpenCV (an open source computer vision library), but concluded the radius measurement may be imprecise.

The biggest challenge of SmartSnooker, may be Finding an accurate transform used to convert the screen-space coordinates to 3D world and 2D table space coordinates. The model of the table can be generated by connecting the pockets identified by the computer vision API. Or alternatively, by using setting ARKit's anchor as the 2D plane of the table and use the PlaneGeometry.geometry method [5, ARKit in IOS] to get the world coordinates of the table boundaries. An ARKit app could then use the techniques outlined in 2.2 or other point-based techniques found in [1], to find an accurate screen space - world transform. The experimental nature of this project, means part of the development of the app will involve experimenting with these different methods to find what works best.

2.5 Rules of Billiards and Snooker

SmartSnooker is an experimental project, with emphasis on creating an accessible AR tool. Pool has simpler rules around which balls can be hit. Thus focusing on billiards rather than snooker allows more time to be spent on the core problem of AR and computer vision, rather than the rules and tactics of the game. If the application is successful for pool, the app could be improved to work on Snooker tables and other cue sports as well.

2.6 Improving Snooker Game

This section provides a very brief overview on how novice billiards players improve their game.

D. Alciatore, outlines a "pyramid of progress" for billiard players [2, Figure 1.1], similar to Maslow's famous phycological concept of a "pyramid of needs". In this pyramid, each layer must be mastered in order to progress to a higher level [2]. In order of most basic to higher level skills are: fundamentals, executing basic shots, position play/strategy and shot making. So, a sensible progression of skill based off D. Alciatore's book would look as follows:

1. cue grip, bridge and stance.
2. aiming methods, ball paths and pocket selection.
3. adding spin
4. strategy
5. advanced shots

Another core property for improvement is fun, focus and discipline, as players are less likely to dedicate energy to improving if they do not enjoy playing [2, Principle 1.1].

[4] observed the effects of technical and tactical training on highly skilled billiard players and a control group. They concluded that novice players benefitted more from computer technology improving technique and more advanced players benefitted more from tactical training. This suggests it is more helpful for a novice to focus on execution, rather than wider tactics.

Chapter 3

Analysis and Requirements

This section describes the objectives of Smartsnooker, discusses the requirements of the application and outlines the criteria for a successful application. Requirements are informed by the objectives of the application and the literary review in Chapter 2.

3.1 Smartsnooker Objectives

This section looks at the goals/objectives for the application, creating the basis for the requirements.

3.1.1 Basic Description of Objectives

The primary goal of Smartsnooker, is to create an accessible mobile application the help novice players improve their game. Hence the success of the application is mostly correlated with how helpful it is to players and the application's ease of use. The aim is to use augmented reality to provide players with real-time guidance on shot selection and execution. The project also aims to increase usability by including an interactive tutor character, providing an engaging way to offer tips and advice to players.

3.1.2 Implementing the Pyramid of Progress

It is important to consider the role the application will have in improving a novice's skills. In 2.6, a pyramid of progress for billiard players was discussed [2]. The first fundamentals for a novice to master are cue grip, bridge and stance. Using Augmented Reality in teaching these skills would be be an entire project in itself, thus these skills will be explained and illustrated by the tutor character. The second layer of the pyramid (aiming methods, ball paths and pocket selection) is where the augmented reality aspect is most useful. In this context, AR is used to show which shots are easiest to execute and rough trajectories of how to execute them. If a player is good enough to precisely follow the trajectory and miss as a result of small inaccuracies in the AR system, they are likely not a novice.

The next layer of the pyramid adds spin to shots. This is beyond the level of skill a novice will need to learn and would add a much greater level of complexity to the project. As discussed in 2.3.2, [21] found modelling spin to be beyond the scope of a similar project. Thus, spin and any higher levels of the pyramid of progress will not be included in the project. These levels could however, be implemented as a continuation of the project.

3.1.3 Implementing Rules

Teaching the rules of the game is fundamental to ensuring players do not commit fouls and play the game properly. The easiest way to implement rules would be to use the tutor character to explain them in a tutorial, similar to the grip, stance and bridge tutorial. Using the AR model of the game would be a much more useful way to teach the games rules. For example, if a foul was detected or a user was trying to pot the black. In context hints like this would be much more engaging for the user and help them understand when certain rules are relevant. The project will aim to implement in context hints, if enough time is available after the AR trajectory problem has been solved. This is another feature that could be implemented as an extension to the project.

3.1.4 Accessibility, Performance and Usability (identifying non-functional requirements)

A key objective of the project is to be easy to use for novices. The tutor character can be utilised to help achieve this. The AR setup process should be accessible with instructions from the character, in one click from the home screen. Plus, a typical user should be able to perform the setup in under a minuet. Any tutorials not using Augmented Reality should also be available from the home screen intuitively in two clicks.

The application should also work in real-time to ensure it is easily usable. Given the constraints of computer vision outlined in the literary review 2.4 (a hosted computer vision API and use of a tripod), some reasonable performance requirements can be outlined. The system should be able process and overlay trajectories within an average of 2 seconds after balls are no longer in motion.

3.2 Requirements

The following is a list of requirements based on the analysis of objectives in 3.1. The requirements' priorities are decided using the MoSCoW method, where requirements labelled as must are essential, should are important but non-essential and could are desirable. Functional requirements are based on the analysis in 3.1.2 - 3.1.3 and non-functional requirements are based on 3.1.4.

Functional Requirements List			
ID	Priority	Description	Acceptance Criteria
F.1	Must	have an original interactive tutor character to provide tutorials and hints.	The application has an original, friendly to provide dialogue out the application.
F.2	Must	have tutorials for cue grip, bridges and stance, provided by the interactive character.	Users can enter a tutorial where the character explains good technique for grip bridges and stance, with pictures/illustrations.
F.3	Must	have tutorials for billiard rules, provided by the interactive character.	Users can enter a multi-step tutorial, where the character explains all the fundamental rules of the game.
F.4	Must	provide AR overlays of shot trajectories to different pockets.	The augmented reality view overlays different potential shots to different pockets. A user can select one of these shots to display on its own.
F.5	Must	show user which shots out of a selection are easiest to execute.	The user is given an indication of what shots out of the selection are easier.
F.6	Should	provide possible shots involving a collision with the cushion.	Easy trajectories involving cushion collisions are included in the displayed shot selection.
F.7	Should	have accurate enough AR to illustrate roughly the region a target ball should be hit to pot the ball.	The angle of collision between the cue and target in AR should be within 10 degrees of the needed angle in real life to pot the ball.
F.8	Should	have an tutor character provide user with advice on how to successfully execute a shot.	Interactive tutor appears in AR view, giving advice on how to aim for different types of shot.
F.9	Could	have rules and tips explained by tutor character, based on the state of the AR table.	Interactive tutor appears in AR view to give tips on avoiding fouls and potting the black ball, based on the current table state.
F.10	Could	allow option to switch to snooker rule set.	All features for rules and strategy have equivalent components to work for a snooker game.

Non-Functional Requirements List			
ID	Priority	Description	Acceptance Criteria
NF.1	Must	A typical user to have the AR component setup within a minuet.	AFG
NF.2	Must	AR setup process / instructions must be one click from the landing screen	A button on the landing screen must enter the AR view, with the tutor character giving instructions on how to setup the table in AR.
NF.3	Must	Tutorials not using AR should also be available from the home screen intuitively in two clicks	Rules, cue grip, bridge and stance tutorials are accessible within two clicks of the landing screen.
NF.4	Must	The App must work in real-time with users not waiting for the shot selection to update	The AR shot trajectories cannot take longer than 2 seconds after the previous shot ends to update, with a stable internet connection.

3.3 Assessing Success

Usability testing will be utilized to assess the utility of the application and gather feedback during and after development. Players of varying skill levels and age groups will be selected to ensure feedback considers a range of abilities and digital literacy. Users will be questioned on both ease of use of the application and how helpful they found it. The main goal is for novice players to agree that the app helped increase their understanding of the rules of the game and how to select and execute shots. Feedback from other abilities is also important to guide further development of the application to help intermediate and advanced players. A feedback form related to the goals above will be given to the test users at different stages of development.

Chapter 4

Progress and Plan

This section describes the the current progress of the project and the plan for next semester.

4.1 Current Progress

At the time of writing, implementation has not started on the application, however this will be the first step of progress after this report. Through this research, I have found different toolkits and methods for augmented reality, computer vision and the physics/rules of cue sports. The next section (4.2), describes when different tools will be experimented with.

4.2 Implementation plan

4.2.1 Experimental Objectives

This section defines 5 different problems that need to be solved during development and possible methods or toolkits to solve them. These correlate to different stages of development. The progression through these stages assumes each step can be solved to an acceptable standard in reasonable time. It is likely one or more of these stages prove to be problematic, so the feasibility of the app's requirements must be under constant review. If requirements change, it is important useful features for an app, or building blocks for future research are still deliverable.

Experimental Objective 1 - ARKit

The first stage will creating an ARKit project in Xcode (Apple's development environment) and getting the app to detect a billiards table as a plane.

Experimental Objective 2 - Ball and Pocket Identification

The next step will involve trying to detect objects from the camera output, such as balls and pockets. Existing open source libraries, such as Joy Wolve's pool table library [31] will be trailed, or failing this, OpenCV's Hough Transforms APIs [27].

Experimental Objective 3 - Ball and Table Position Estimation

Once balls and pockets are identified, they need to be translated to either a 3D-world coordinate system or a 2D-table coordinate system, either using ARKit's features or techniques outlined in [1].

Experimental Objective 4 - Creating a Game and Trajectory Model

Once a model of the table and ball positions has been created, I will attempt to use Pooltool [20] to simulate potential shots and assess how this affects the app's performance. If this is to resource intensive, a trajectory-based system as described in section 2.3.2 will be implemented to find possible ball paths. Trajectories to pot balls will be calculated, working backwards from pockets, to potential target balls, to the cue ball. Easy shots can be defined as having the smallest angle between the cue ball path and the target ball path.

Experimental Objective 5 - Improved Shot Suggestions

The final experimental stage will involve further research into what makes a shot easy with the intension of improving the shot selection process to suggest the easiest shots to user.

4.3 Designing The Tutor Character

The tutor character will need to be designed before implementation. Text-based diagonal would be an easy way to implement the character. This would involve animated frames of the character appearing on screen, with dialogue displayed as text. These different frames of the character must be designed, so the character can be interactive and change emotion during dialogue.

Chapter 5

Conclusion

This report has analysed the potential technologies and methodologies that could be used to create an AR billiard assistant. There are many potential challenges in accurately modelling a billiards table from a camera input, but this report has identified many starting points for implementation and experimentation. Requirements may change during development depending on the ease of implementation and accuracy that can be achieved. The fundamental aim of the project is create an accessible app to help novices, with the help of AR. This will not change and so will inform any requirement changes.

Bibliography

[1] Ababsa, F., and Mallem, M. Robust camera pose estimation combining 2d/3d points and lines tracking. In *2008 IEEE International Symposium on Industrial Electronics* (2008), pp. 774–779.

[2] Alciatore, D. *The Illustrated Principles of Pool and Billiards*. Union Square & Company, 2017.

[3] Amin, D., and Govilkar, S. Comparative study of augmented reality sdk's. *International Journal on Computational Science Applications 5* (02 2015), 11–26.

[4] Andruchshishin, I., Karaneev, A., Denisenko, Y., and Geraskin, A. Experimental study of efficiency of technical and tactical actions in billiard sport. *International Journal of Human Movement and Sports Sciences 11(4)* (2023), 779–788.

[5] Apple Developer. Arkit — apple developer documentation. https://developer.apple.com/documentation/arkit . visited on 2024-11-22.

[6] Broadfoot, W. *Billiards*, vol. 5. Longmans, Green and Company, 1896.

[7] Carmigniani, J. Furgt, B. A. M. Augmented reality technologies, systems and applications. *Multimedia Tools and Applications 51* (12 2010), 341–377.

[8] Cervenak, R., and Masek, P. Arkit as indoor positioning system. In *2019 11th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)* (2019), pp. 1–5.

[9] Cheng, J., Chen, K., and Chen, W. Comparison of marker-based ar and markerless ar: A case study on indoor decoration system. In *Lean and Computing in Construction Congress (LC3): Proceedings of the Joint Conference on Computing in Construction (JC3)* (2017), pp. 483–490.

[10] Crudebyte. Billiardradar. https://www.crudebyte.com/billiardradar/. visited on 2024-11-22.

[11] Dilek, U., and Erol, M. Detecting position using arkit. *Physics Education 53*, 2 (jan 2018), 025011.

[12] endoflife.date. endoflife.date. https://github.com/endoflife-date/endoflife.date?tab=readme-ov-fileread accessed on 2024-12-04.

[13] IQ Reality. Iq billiards. https://iqreality.com/. accessed on 2024-12-04.

[14] Jähne, B., Haussecker, H., and Geissler, P. *Handbook of computer vision and applications*, vol. 2. Citeseer, 1999.

[15] Jebara, T., Eyster, C., Weaver, J., Starner, T., and Pentland, A. Stochasticks: augmenting the billiards experience with probabilistic vision and wearable computers. In *Digest of Papers. First International Symposium on Wearable Computers* (1997), pp. 138–145.

[16] Kajastila, R., Holsti, L., and Hämäläinen, P. The augmented climbing wall: High-exertion proximity interaction on a wall-sized interactive surface. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2016), CHI '16, Association for Computing Machinery, p. 758–769.

[17] Kiefl, E. Pooltool documentation. https://pooltool.readthedocs.io/en/latest/index.html. visited on 2024-12-04.

[18] Kiefl, E. The algorithmic theory behind pool/billiards simulation. https://ekiefl.github.io/2020/12/20/pooltool-alg/ , Dec 2020. visited on 2024-11-22.

[19] Kiefl, E. The physics of pool/billiards. https://ekiefl.github.io/2020/04/24/pooltool-theory/ , April 2020. visited on 2024-11-22.

[20] Kiefl, E. Pooltool: A Python package for realistic billiards simulation. *Journal of Open Source Software 9*, 101 (sep 2024), 7301.

[21] Medved, S. Augmented reality billiards assistant. *William's College Honors Research Projects, The University of Akron* (2020).

[22] Milgram, P., Takemura, H., Utsumi, A., and Kishino, F. Augmented reality: A class of displays on the reality-virtuality continuum. *Telemanipulator and Telepresence Technologies 2351* (01 1994).

[23] Olwal, A. *Unobtrusive augmentation of physical environments: interaction techniques, spatial displays and ubiquitous sensing.* PhD thesis, KTH, 2009.

[24] Orange Loops. Pocket billiards drills stats — drillroom. https://apps.apple.com/us/app/drillroom-billiard-training/id1539827505. accessed on 2024-12-04.

[25] Oufqir, Z., El Abderrahmani, A., and Satori, K. Arkit and arcore in serve to augmented reality. In *2020 International Conference on Intelligent Systems and Computer Vision (ISCV)* (2020), pp. 1–7.

[26] Porto, A. Aporto/cassapa: An augmented-reality system for pool and billiard players. https://github.com/aporto/cassapa, Dec 2019. visited on 2024-11-22.

[27] Schmidt, E. Measuring the speed of a floorball shot using trajectory detection and distance estimation with a smartphone, 2016.

[28] Sousa, L., Alves, R., and Rodrigues, J. Augmented reality system to assist inexperienced pool players. *Computational Visual Media 2* (2016), 183–193.

[29] Sport Scotland. Sports participation: Snooker, billiards and pool fact sheet. urlhttps://sportscotland.org.uk/media/a01dodt2/snookerbilliardsandpool.doc, 2023. accessed on 2024/12/08.

[30] TouchMagix. Magixpool. www.touchmagix.com/products/magixpool. accessed on 2024-12-04.

[31] Wolves, J. Roboflow ball dataset. https://universe.roboflow.com/joywolves/ball-qgqhv , aug 2023. visited on 2024-12-05.

[32] Zhang, Y. Women as 'space invaders' in a gendered billiards hall: The male space and moments of change. *European Journal of Cultural Studies* (2024).

[33] Zhou, F., Duh, H. B.-L., and Billinghurst, M. Trends in augmented reality tracking, interaction and display: A review of ten years of ismar. In *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality* (2008), pp. 193–202.