

College of Business, Technology and Engineering

**Department of Computing
Project (Technical Computing)
[55-604708]
2022/23**

Author:	Joe Kirkup
Student ID:	29026253
Year Submitted:	2023
Supervisor:	Michael Meredith
Second Marker:	Bob Steele
Degree Course:	Computer Science BSc
Title of Project:	Tracking snooker balls to assist in their replacement after a foul shot

Confidentiality Required?

NO

I give permission to make my project report, video and deliverable accessible to staff and students on the Project (Technical Computing) module at Sheffield Hallam University.

YES

Acknowledgements

I would firstly like to thank my project supervisor Dr Mike Meredith. Without his unwaveringly positive support and useful advice I would not have been able to complete this project.

I also wish to say thanks to Steve Barton of BartonSnooker for permission to use his video throughout the project, and his kind words of encouragement.

Finally, many thanks to the owner and patrons of Lincoln Snooker Club for their permission to film in the premises.

Abstract

In the game of snooker, there is no existing solution for accurately or quickly replacing the balls after a foul shot despite modern technology being applied to many similar problems in other sports.

This project was undertaken to create a technical proof-of-concept for a software solution which would be connected to a video feed of a match of snooker, track and store the positions of the balls after each shot and allow recollection of this data to help replace the balls after a foul.

In completing the majority of the project aims, the concept was proven, demonstrating the viability of such a system were someone to undergo the task of creating a market-ready system using similar algorithms to those used in this project and described in this report.

Table of Contents

1	Introduction.....	6
1.1	Project Background and Motivation	6
1.2	Scope.....	6
1.3	Project Aims.....	6
2	Research.....	7
2.1	Similar Projects	7
2.1.1	The Pool Ball Tracker (Marr, 2012)	7
2.1.2	Snooker balls tracking on video (Trusov, 2022)	7
2.1.3	TrackingSnookerBalls (Nirenshteyn, 2021).....	8
2.1.4	snooker-ball-tracker (Black, 2023)	8
2.2	Relevant Technologies.....	9
2.2.1	Code Libraries.....	9
2.2.2	Languages	10
2.2.3	Engineering Practices.....	11
3	Design and Architecture	12
3.1	Selected Technologies.....	12
3.2	Planned Architecture and Design Considerations.....	12
4	Development	13
4.1	Terminology.....	13
4.2	Project Structure.....	13
4.3	Perspective Warping	13
4.4	Tracking Balls.....	14
4.4.1	Shapes	14
4.4.2	Colours.....	15
4.5	Basic Output – rendering the balls.....	15
4.6	Video.....	16
4.6.1	Supporting video throughput	16
4.6.2	Detecting when a new shot has been played.....	16
4.6.3	Storing past states.....	17
4.6.4	‘Rewinding’ to a previous shot on-demand	17
4.7	Performance Considerations	18
4.8	Abstract Views.....	18
5	Critical Evaluation and Reflection.....	20
5.1	Ethical Concerns	20
5.1.1	Research.....	20
5.1.2	Development and testing.....	20
5.1.3	Real world impact	20

5.2	Success of Engineering Approaches	20
5.3	Evaluating Project Success	21
5.4	Evaluating Personal Development	22
	Bibliography	23
	Table of Figures	25
	Appendices.....	26
	Appendix A – Project Specification.....	26
	Appendix B – UREC2 Ethics Form	29
	Appendix C – Copyright Permission for YouTube video.....	34

1 Introduction

1.1 Project Background and Motivation

When a foul is committed in the game of snooker, the player who the foul was committed against can request for the balls to be replaced to their positions before the foul shot and have their opponent play the shot again.

If a referee is present, it is their job to replace the balls, and in televised matches the TV cameras are rewound to inspect their previous positions if necessary. Due to the inaccuracy of this method, players sometimes disagree with how the referee has placed the balls, and there have been instances of players disagreeing with each other on this matter.

In non-televised matches, be it non-TV tables at professional tournaments, amateur tournaments, or casual settings, there is obviously no TV camera to rewind. While some snooker clubs may have cameras recording the tables, these are usually not configured in a way where their footage could be accessed while a game is ongoing. This presents the issue that, for all the listed scenarios, there is no solution at all for accurately replacing the balls – only the player’s judgement of the positions can be used as a resource, and particularly if more than one balls needs replacing, this is likely to be unreliable and inaccurate.

Given the understanding we now have of this unsolved issue in the game, the aim of this project is to create a software solution which greatly eases this process, and, importantly, removes subjectivity from the matter.

1.2 Scope

While all planned functionality is listed below as within scope, obviously a comprehensive list of everything out of scope cannot be produced. The functionality specified as outside of scope is a clarification on ideas that were mentioned or potentially implied in the Appendix A – Project Specification, Abstract, or Introduction.

Within scope:

- Tracking balls and their colours
- Supporting video throughput
- Recalling a previous position

Outside of scope:

- Advanced output methods such as augmented reality
- Automatic scoring
- Automatically detecting foul shots, or any other referee-called decision

1.3 Project Aims

1. Detect the positions of the snooker balls in an image of a snooker table.
 - a. Also detect the colour of each ball.
2. Display the tracked information in some way that can be used to restore the balls to this state.
3. Support video throughput as a proof-of-concept for a camera feed.
 - a. Store each new shot as it is played.
 - b. Allow displaying a previous table state on-demand.
4. Explore advanced output methods such as augmented reality.

2 Research

2.1 Similar Projects

As part of the research stage of the project, projects that aim to tackle similar problems or are technically relevant were identified and their technologies and approaches were reviewed. The information gathered was used to inform decisions on technologies and design approach.

None of the projects found were markedly complete at the time of research, or writing, and their working status is unknown. For this reason, they cannot be analysed as products but simply as projects. However, this does not detract from the fact that they have all implemented a significant amount of core functionality and much can be learned from them.

2.1.1 The Pool Ball Tracker (Marr, 2012)

The team at GoCardless worked on this system over a 48-hour ‘Hackathon’ and their experience is summarised at high level in a follow-up blog post.

They developed their system in C, using OpenCV. They convert each input frame from an overhead webcam to HSV and mask the image multiple times using each of the colours of the balls.

After initially using the Hough transform (via OpenCV’s *houghCircles* function) to detect balls in the filtered image, they found greater reliability using *findContours*, an OpenCV function which detects generic shapes in an image. This was due to motion blur and slow shutter speed distorting the balls when they were in motion, leading to *houghCircles* failing to detect them.

Marr notes towards the end of the post that the system works at a low 10 frames per second, due to expensive computation. This can be taken as a fair warning to be careful about the algorithms and approaches used in design, as they could quickly become very costly.

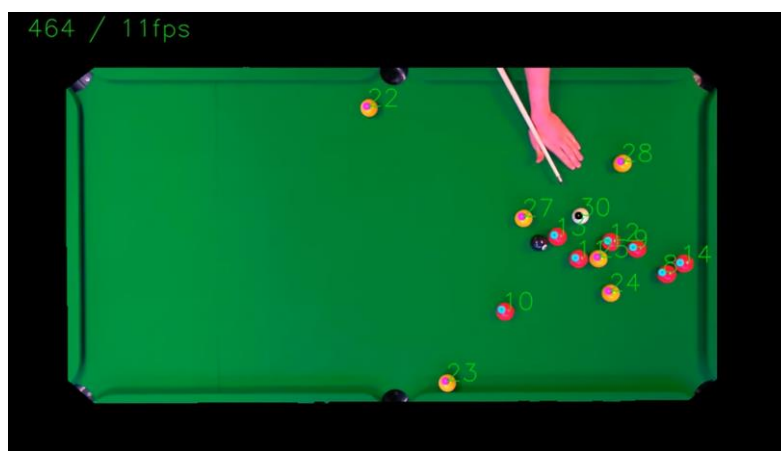


Figure 1: The Pool Ball Tracker's output as of the end of the ‘Hackathon’.

As shown in Figure 1, the ball tracking itself seems to be able to detect all the balls and does not produce any false positives, giving merit to their approach.

2.1.2 Snooker balls tracking on video (Trusov, 2022)

This notebook goes into detail about the technical approach and outlines the power of OpenCV when it comes to image processing.

The system is developed in Python, using OpenCV, and relies on template matching for the core functionality of ball detection. This means that the solution may need new templates manually creating if a different camera or camera setup was used unless convincing perspective correction and removal of optical distortion were to be implemented.

Trusov performs pattern matching in a greyscale colour space for the sake of speed, and this has the interesting effect of grouping the patterns based on lightness value, making determining colours more straightforward as they are automatically partially sorted past the pattern matching phase.

The code runs slower than real-time with the 2:42 duration clip linked at the bottom of the notebook taking 9:47 to be processed – another early warning about performance of these types of computer vision applications.

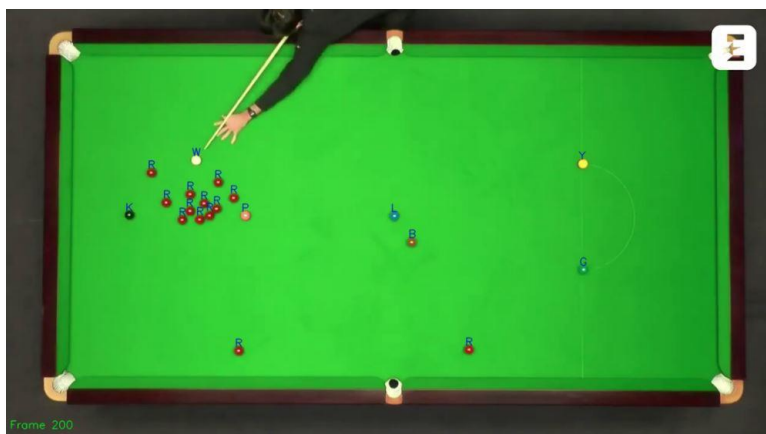


Figure 2: Trusov's code labelling the colours of each ball.

2.1.3 TrackingSnookerBalls (Nirenshteyn, 2021)

Written in Python with OpenCV, Nirenshteyn's code streams a video file, using hard-coded values to warp the perspective of each input frame to a top-down view. The green of the table is then discarded using a mask, leaving only the objects on or above the table. After applying the *findContours* function, the results are filtered based on their geometry to detect balls in the frame.

Having earlier converted the image to HSV, the colours are found using the average colour inside each ball's contour and used to project the ball at its detected position onto a generated 2D top-down view. A frame of the result is shown in Figure 3.



Figure 3: TrackingSnookerBalls generating a 2D view of the table.

2.1.4 snooker-ball-tracker (Black, 2023)

A more complex project than any of the previously reviewed, Black set out to build a precursor to an automatic scoring system by first developing a ball tracking solution in Python, using OpenCV. The system features a GUI which displays a visual output of what the system is seeing, configurable options and debug and user info.

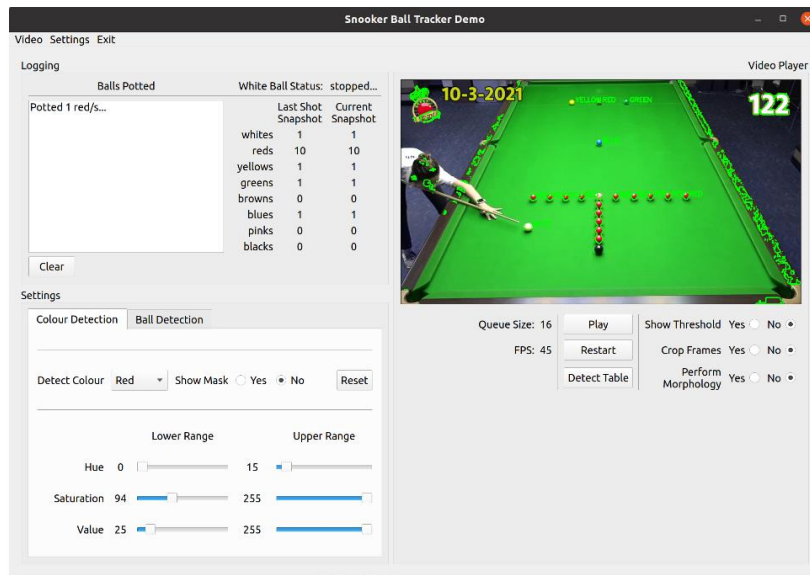


Figure 4: The snooker-ball-tracker GUI, showing options, output and what it sees.

Similarly to other projects, the table is masked, *findContours* is used, the contours are filtered, and the colours are detected in the HSV colour space.

This project goes further than others: there is some automatic potting detection and scoring functionality implemented but it remains a work in progress at the time of research and writing.

2.2 Relevant Technologies

To determine the most applicable and effective technologies, research was conducted to determine available tools and their suitability for the project. Below is the resulting overview of tools found and conclusions drawn.

2.2.1 Code Libraries

The aim being to develop a heavily computer vision-focused piece of software meant a library for this technology was chosen early in the project, with research being done into popular options. Multiple online lists (SuperAnnotate, 2021) (Boesch, The 12 Most Popular Computer Vision Tools in 2023, 2023) were reviewed and condensed into the shortlist below.

OpenCV

OpenCV is the “de facto standard tool for image processing” (Boesch, What is OpenCV? The Complete Guide, 2023), and its long-standing status as industry standard means there is a massive amount of information online about it, and well-maintained wrappers exist for languages beyond the officially supported C++, Python, Java and MATLAB interfaces (OpenCV, 2023), including languages such as JavaScript (OpenCV, 2023), C# (Emgu, 2022), and others.

Along with the wide variety of language support and online resources, OpenCV supports Windows, Linux, Android, and MacOS (OpenCV, 2023), making it a good choice for a proof-of-concept whose algorithms are likely to be ported to an unknown platform.

SimpleCV

As implied by its name, SimpleCV aims to provide a simpler approach to computer vision. It is not itself a library, but a framework wrapping a more simplistic interface around OpenCV and other libraries (SimpleCV, 2023). SimpleCV only supports Python but does so with extensive documentation and tutorials.

Its simplicity makes it a tempting choice for developing a proof-of-concept, as it should allow for easier prototyping of algorithms.

BoofCV

(BoofCV, 2023) An open-source computer vision library available for Java.

Developed in Java and available for use in Java, Kotlin, Python, and Processing, BoofCV has extensive computer vision features that targets mainly higher-level processing (SuperAnnotate, 2021).

Given the time constraints of the project, the lesser popularity of BoofCV is worth considering as there are likely less learning resources and answered questions online than for other libraries.

TensorFlow

Though it is targeted at machine learning, this topic and computer vision are closely related, and TensorFlow provides computer vision tools for this reason. Providing official APIs for Python, C++, Java, and JavaScript (TensorFlow, 2021), and many other community-maintained language bindings being available, TensorFlow can be used on almost any development platform and has a huge amount of online support and resources available.

Due to the extensive learning resources and documentation available, TensorFlow is a tempting choice for a proof-of-concept app, but it is worth keeping in mind said resources will be mostly targeting machine learning as opposed to specifically computer vision.

OpenVINO

OpenVINO is a large framework providing algorithms from many different libraries such as TensorFlow, PyTorch, Caffe, and others. It aims to provide tools to leverage machine learning in vision, audio and language models, and targets C++ and Python (OpenVINO, 2023). Community-maintained APIs exist for languages such as Java (Kurtaev, 2023)

As OpenVINO is an attempt to bring together the most optimal algorithms from many different libraries, it is likely unnecessary for a technical proof-of-concept where optimisation is not of the utmost importance – a singular library may be preferable to keep the prototyping process simple.

2.2.2 Languages

C++

While C++ may be famous for its efficiency due to its low-level and barebones nature, it is infamous for its difficulty to learn and unhelpfulness when developing software for the same reason. This would be a strong contender for a market-ready application, especially targeting efficiency to ensure viability on lower end hardware, as computer vision can be quite resource intensive.

However, given the aim of the project is to create a proof-of-concept, it may be wise to choose a language which allows for faster prototyping and ease of development.

Python

Python is known for being good for rapid prototyping (Zeller, 2020), with many developers using it as a platform for creating proof-of-concepts or rough implementations to better understand the required architecture or data structures to save rewriting more complex codebases later.

Even though there are personal gaps in the developer's knowledge and experience with Python development coming into the project, there is a huge wealth of relevant resources online and most Python concepts and libraries are designed ease of use in mind so these would be unlikely to present any major issues.

C#

With Visual Studio as an IDE for writing C#, tools are provided to greatly ease the difficulty and speed of GUI creation in the form of its WPF designer. C# / .NET has extensive documentation and online support resources and given its straightforward but modular approach to program architecture, would make for a worthy platform for prototyping.

Given the lack of official support from computer vision libraries however, it may not be the best choice as this is likely to result in less documentation and resources specific to the scope of this

project, something that is not ideal when undertaking a large amount of learning before and during the development process.

HTML / JavaScript

HTML, backed up by JavaScript code, is the most popular development platform in the world (Stack Overflow, 2022), and potentially the most used, due to its deployment on the web and access via any user platform using any modern web browser.

HTML itself facilitates what is arguably one of the simplest GUI development processes, with positions of GUI elements often not even needing to be specified unless required, which may help avoid time wasted designing a GUI layout or getting tied up in UX/UI concerns developing what is a proof-of-concept where these concerns are mostly irrelevant.

JavaScript running in the backend of a HTML page enables not only dynamic control of the page (GUI) elements, but any other processing required, and the two work together seamlessly.

2.2.3 Engineering Practices

Agile software development framework

Throughout the development process, an agile approach will be taken. As the project aims may be subject to change in reaction to development roadblocks in an attempt to keep the project moving, an agile approach is necessary in order to react to such a thing.

Furthermore, as the project is formatted as a deliverable and a report, and extensive documentation is not required, this puts it closer in line with the standard agile approach of focusing more on software than documentation (Brush & Silverthorne, 2022).

Feature driven development is a hallmark of the agile framework and is a primary aim of this project as development takes place – because the list of features is not extensive due to the proof-of-concept nature of the planned deliverable, this simplifies the development process down from worrying about integrating features and user experience to focusing on algorithms and functionality.

Version control

While developing the deliverable, version control will be used in the form of a GitHub repository. This ensures integrity of the codebase, helps debug hard to solve issues in code by reviewing earlier versions, and allows for greater flexibility in design by removing the worry of keeping old and likely unnecessary code around.

3 Design and Architecture

3.1 Selected Technologies

HTML, JavaScript and OpenCV.

Following the research conducted into languages and computer vision libraries, the decision was made to use HTML and JavaScript as this will allow for very straightforward rapid prototyping of basic but intuitive UI via HTML and necessary algorithms via JavaScript. Paired with the choice of industry standard computer vision library OpenCV, which has an official JS interface and a good amount of online documentation and learning resources, the tools chosen should be more than sufficient for developing the proof-of-concept project.

Any other minor code libraries required may be used, if so, this will be specified where applicable in the Development section.

3.2 Planned Architecture and Design Considerations

Though the application will be a website, it will be designed for local access – the use of HTML is only to provide a user interface and support JavaScript code running in the background.

Web-specific JavaScript features could be utilised but should not be at the core of the functionality as it should be possible to port the designed algorithms to another platform with ease.

Since the aim of the project is only to create a proof-of-concept app, there will not be much consideration given to fluent user experience design or stylisation of the user interface – so long as there is a reasonably intuitive method of operating the solution, this will be acceptable.

4 Development

Throughout the development process, a lot was learned, and the direction of the project was tweaked many times. This section will detail thoughts behind the technical implementation as well as any issues encountered, what was done to solve them, and their effect on the direction of the project.

4.1 Terminology

There are some terms used in this section whose meanings may not be immediately clear, as they have been defined within the scope of the project.

State: The array of ball positions returned by *trackBalls()*, the function outlined in 4.4.1.

Candidate state: The **state** extracted from the current frame of source video.

Current state: The latest **candidate state** that is meaningfully different from the latest current state (to eliminate tracking inaccuracies)

Last state: The latest ‘end-of-shot’ **state** stored and displayed in the mini-canvases.

4.2 Project Structure

The project is structured as a locally hosted website. Making use of a typical basic web folder structure – *html*, *js*, *media*, and *css* folders – the aim of the high-level architecture was to keep it simple to avoid wasting development time on something that is, in a proof-of-concept, mainly irrelevant.

Due to technical restrictions, which will be discussed in the Evaluation, the project was hosted on a local server during development and testing. This was done using npm package *http-server* (Thornton, 2022).

The *index.html* landing page allows the user to choose an image or video file. Upon choice, the page redirects to */html/selectCorners.html* which asks the user to select the four corners of the table, as discussed in the next section. Finally, depending on whether an image or video was selected, the page redirects to */html/analyseVideo.html* or */html/analyseFrame.html*, and the deliverable processes the media.

4.3 Perspective Warping

Before any detection takes place, there should be some ‘normalisation’ of the image to a corrected top-down view of the table; it will make detection far simpler if we can design detection and filtering algorithms with as many uncertainties as possible removed.

OpenCV provides a function to do this: *cv.warpPerspective*. It requires a transform calculated by *cv.getPerspectiveTransform*, into which we pass an array of user-selected corner values, retrieved on the *selectCorners* page using a click event listener recording the coordinates of four user clicks on a frame of the chosen video or image. Along with the corners of the output canvas, the transform is calculated and applied to warp the input frame appropriately:



Figure 5: A frame of example source video (Barton, 2021).



Figure 6: The input frame warped to a top-down perspective.

This code is wrapped up in the *normaliseView* function and called on every new input frame. There are optional parameters to display the warped frame on a canvas using *cv.imshow*, and this functionality is used in the deliverable to show what the program is ‘looking at’ as it tracks.

4.4 Tracking Balls

4.4.1 Shapes

Many shape detection algorithms have been devised throughout the history of computer vision, and OpenCV implements some of the most widely used ones. These were investigated through online resources and the most appropriate seeming algorithms were trialled to ascertain their viability for the use case.

HoughCircles (Hough Transform)

Initially, the *cv.HoughCircles* function was used for ball detection, as it seemed an obvious choice out of the gate when reading about OpenCV’s capabilities. However, at the time it was difficult to produce acceptable results using the function so further research was done into other object detection algorithms which were then employed and developed further.

FindContours

Implementing a now famous algorithm published in 1985 (Suzuki & Abe, 1985), *cv.findContours* takes an input image and some basic parameters and returns the found contours. As we expect this algorithm to find objects that are not balls, we must filter the returned data. The method created for achieving this is as follows:

Get the contour's *minAreaRect* – the bounding box around the contour. This bounding box will be rotated to fit the smallest size possible, but since we are looking for circular objects, we can forget about this angle as anything that fits our filter should do so with complete rotational symmetry.

Next, calculate the area and aspect ratio of the bounding box. The aspect ratio can be used to filter out any objects that are not circular or close to circular, as a perfect circle's bounding box will have a 1:1 ratio – we simply discard objects that are not within a certain range around the perfect 1:1 value. Area is then used to further filter objects that may pass the aspect ratio check but are too big or small to be a ball.

The bounding boxes that successfully pass through this filter are added to an array and returned as the current state of the table.

4.4.2 Colours

Now that we know the positions of the balls, we can use this information to get their colours from the perspective-corrected input image. The process is a simple one: a blank mask is created, and a small circle is drawn at the position of the ball. The mask is then applied to the input image and *cv.mean* is used to get the mean RGB value inside the masked area.

While all the example projects operated in the HSV colour space, this complication was avoided here as matching colours to the exact colours that the balls should be, was considered to be out of scope for the project. Instead, the mean RGB value that we just found is directly used as the fill colour for the relevant ball.

If colour matching were to be attempted, it would be advisable to convert the input image to either the HSV or L*A*B colour space, as these facilitate more accurate and reliable colour comparison.

4.5 Basic Output – rendering the balls

As some form of output is necessary for debugging during development anyway, a function was created very early on to visually represent the positions of the balls. It simply draws a white circle at each ball's position:

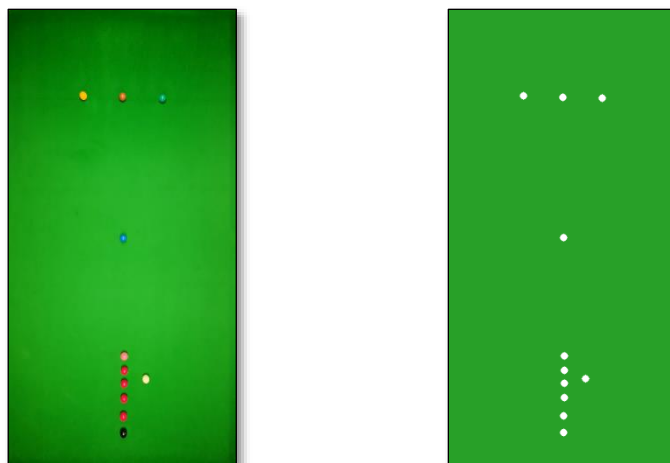


Figure 7: The warped input frame and rendered tracked balls.

Note that the size of the drawn balls is not dependent on the size of the detected ones, as this only opens the door to inconsistencies – we know exactly what size the balls should be and some balls, especially the green, may not have their entire area detected.

The output can be improved by utilising the ball colour detection function and adding the table's markings (white lines and dots) (Blue Moon Leisure, n.d.):

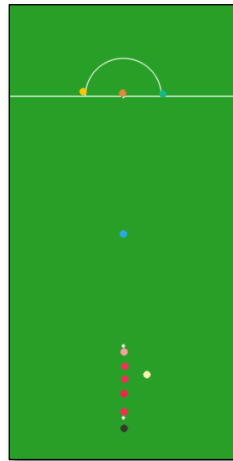


Figure 8: The tracked balls rendered in full colour with the table markings

The markings for the table are drawn statically; not detected from the tracked media. In Figure 8, some of the balls are visibly misaligned with the markings. This is mainly because of optical distortion in the source video, something that was not addressed in this project but will be discussed in the Evaluation.

4.6 Video

4.6.1 Supporting video throughput

On the *analyseVideo* page, frame data is retrieved from a HTML video element via the OpenCV *VideoCapture* object's *.read()* function. The source video is hidden via css and the perspective warped frames are displayed for debugging purposes.

4.6.2 Detecting when a new shot has been played

An algorithm for evaluating state difference

For each frame of video that is tracked, we want to determine if the balls have moved since the previous frame. This is an important piece of the functionality as we need this in order to detect when the balls have stopped moving, indicating they have come to a stop after a shot has been played, so that we can save their current state. To achieve this, a function called *compareStates* was created which returns a number representing the difference between two arrays of balls.

The function first attempts to discern which ball in the new frame correlates to which ball in the previous frame, which we do not explicitly know but can be determined with a high degree of certainty as we can expect the balls to not move much between two frames, especially when they are slowing to a stop, and this is the point in time that we are most interested in. This is done by iterating through each of the balls in one of the arrays and finding the ball closest to it in the other array; whenever a new closest ball is found, its index and distance from the ball are stored.

Finally, if there are any extra balls in the longer of the two arrays, the closest ball to each is double checked. This helps to reduce erroneously high values that may emerge from tracking errors ('ghost balls') or fast-moving balls.

If the *compareStates* function returns a value greater than a certain threshold – 5 was arbitrarily chosen after observing its output values for some time – then the candidate state is meaningfully different from the last known one, so we consider the balls to have moved, and we set the value of the current state to that of the candidate state.

Comparing numbers of balls

In addition to calculating the difference in ball positions between states, the number of balls is compared. This is because we need to know if a ball was re-spotted, as if it was then we need to replace the previously stored state with the new one. For example, if the shot taken was a pot on the

black ball, then the black ball will be placed back on its spot by the referee. However, it is possible that the system detected the end of the shot before the black ball was re-spotted, and we must update the stored state if this was the case.

Utilising confidence values

Sometimes balls fail to be tracked for a frame or two, or, inversely, a video artefact will cause a temporary ‘ghost ball’. Additionally, the tracking algorithm does not have any temporal understanding, meaning each frame is tracked independent of information learned from the previous one, causing balls’ tracked positions to sometimes vary by a few pixels from frame to frame due to video artefacts.

Because the ball tracking functions do not provide perfect data, we must design the algorithms that rely on this data with this imperfection in mind. To account for this then, a series of values are used to express the code’s confidence that a change happened:

For example, if there is one less ball in the candidate state than the last recorded state, the so-called ‘confidence value’ for a ball being potted is incremented by 1, starting at 0. After each incrementation, its value is checked and if it reaches a pre-set threshold then the code assumes that a ball has indeed been potted.

These confidence values are used for the balls being still, a ball being potted, and a ball being re-spotted.

4.6.3 Storing past states

When the above logic is applied, we occasionally find a new still state, whether a ball has been potted, replaced, or neither, we wish to store at least the most recent previous state for the purpose of replacing the balls if needed – this is the main aim of the program.

Some deliberation was given as to how to store past states, and which states need storing at all. To demonstrate functionality more clearly, and because the performance impact is minimal, the decision was made to retain a list of all previous states while the video is being analysed. The states themselves are already represented neatly as an array of objects and we simply push each new state to an array, giving us a record of all past states in the current video.

For the purpose of demonstration and debugging, each new state is rendered to a miniature canvas, and a message is displayed showing the program’s reasoning for adding or updating a state.

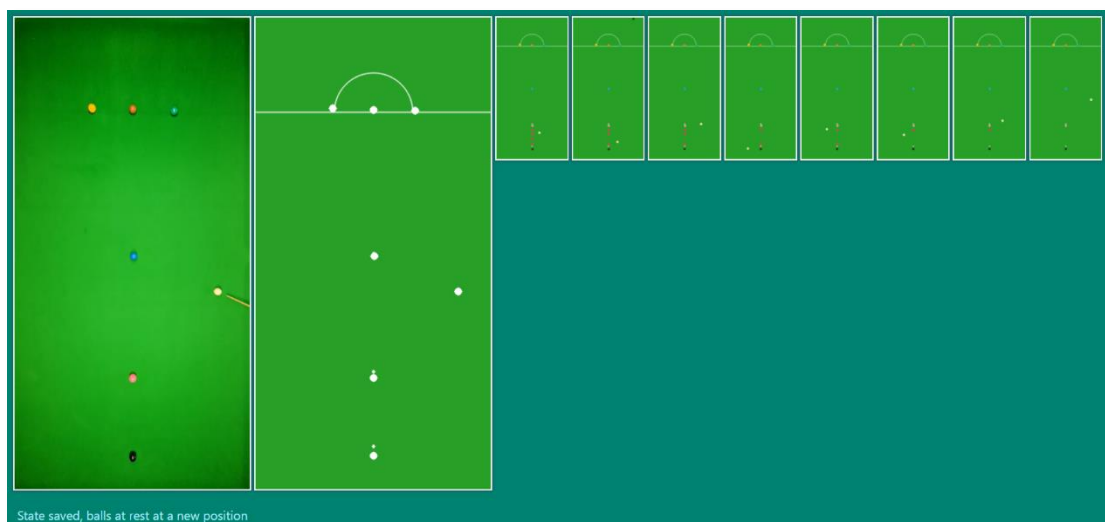


Figure 9: Miniature canvases displaying each state.

4.6.4 ‘Rewinding’ to a previous shot on-demand

Thanks to the grid of all previous shots that the program shows, we can select any of the previous shots, for example using a click event listener on the mini canvases. A hotkey or button could also

easily be added to recall the most recent saved table state; this would likely be the route taken in a market-ready application.

Accessing a past table state is trivial in the code thanks to the array of states, and a function already exists to render one of these states to a canvas complete with table markings, so creating a large display of a previous state is easily achieved:



Figure 10: A large render of one of the previous states.

4.7 Performance Considerations

Computer vision applications often require a lot of computational resources[source], as was in the Research section when reviewing similar projects which all struggled to run in real-time. Thankfully, due to the nature of this project, there are several liberties which can be taken which benefit performance:

Limiting colour detection

Detecting the colours of balls is expensive – the input frame is processed and scanned in several steps for each ball detected. Luckily, we do not need to detect the colour of the balls all of the time.

Given that the main aim of the project is to help replace the balls to a previous still position, we only need colour information when a still position is detected. This takes us from running colour detection once per frame to once every several seconds as the balls come to a stop at the end of a shot.

Real-time processing

Since we only care about accurate recording of position when the balls have come to a complete stop, the program does not actually need to keep up with all the input frames; when the balls are not moving the code will have enough time to realise this even if it is not running in real-time.

If necessary, performance, in the way of framerate of video analysis, can even be purposefully limited in order to consume less client resources, reducing power usage and potential device lag, especially in a market-ready application which may target mobile platforms.

4.8 Abstract Views

As discussed earlier, perspective warping is performed by the code to normalise every input frame to a top-down cropped view of just the bed of the table, as it helps to standardise the input data as much as possible. However, this method has its limitations:

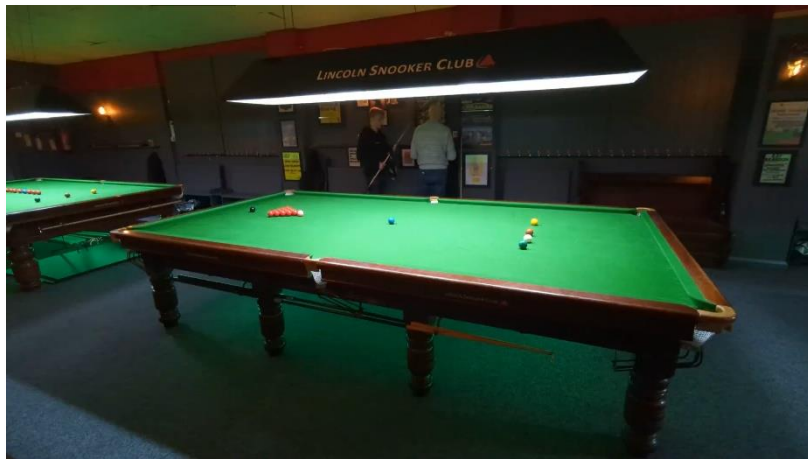


Figure 11: A frame of video taken from a mobile phone placed as high as often possible in a casual setting.

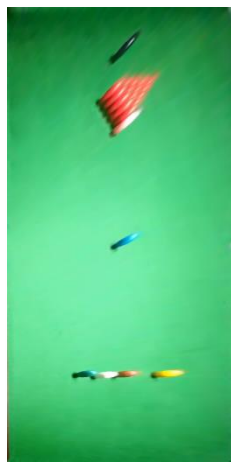


Figure 12: Figure 11 warped to a top-down view using the `normaliseView()` function.

If the angle of an image or video is too far from our desired top-down perspective, then a simple perspective warp results in an unusable output frame – too much visual information is lost and the warped image cannot be used to determine discern one ball from another, let alone any accurate positional data. To parse the visual data from this angle would require a different approach and heighten the need for correcting optical distortion, which is a problem not tackled in this project.

5 Critical Evaluation and Reflection

5.1 Ethical Concerns

Due to the nature of the project and the approach taken in research and development, there was little in the way of ethical concerns, but an outline of considerations taken is as below.

5.1.1 Research

No market research or user questionnaires were necessary in the research phase of the project as the deliverable was a technical proof-of-concept, so no formal ethical considerations were required.

Standard ethical procedures were followed, such as citing all used sources, integrity, non-maleficence, integrity, and informed consent.

5.1.2 Development and testing

The main sample video used throughout testing and for demonstration in the report was taken from Steve Barton's YouTube channel BartonSnooker. Explicit permission was obtained, as demonstrated in Appendix C – Copyright Permission for YouTube video. Thanks are extended to Steve for his cooperation and kind words.

As part of gathering further sample video for use during both the development and testing of the deliverable, an approximately 30-minute video was recorded of the developer and a family member playing a frame of snooker in the Lincoln Snooker Club. Explicit permission was granted by the owner of the club and any patrons who might have appeared in frame. Any frames or sections of video used in the report or any demonstrations only include the developer and their family member.

5.1.3 Real world impact

The argument could be made that, given enough evolution of ball tracking technology, and a system which understood the rules of the game of snooker, that the jobs of human referees could be put in danger.

In accordance with ethical non-maleficence, and in general good-will, this project does not propose any such system, and in fact specifically targets a system which would explicitly and only assist snooker referees. Many other sports have adopted technologically assisted refereeing, and though such technology has proved controversial in some sports, ethical reports argue for it (Tamir & Bar-eli, 2021) (Spitz, Wagemans, Memmert, Williams, & Helsen, 2021), and it goes unnoticed in many as it silently works to improve fairness.

Furthermore, no decision-making features are suggested in the software proposed in this report: it aims only to remove subjectivity from something that is accepted as objective within the game but difficult to measure as such without the appropriate tools.

5.2 Success of Engineering Approaches

The agile framework was successfully followed throughout the development of the deliverable, with multiple short sprints to focus on development of particular features in-between working on the project report and other university assignments.

Though project management tools such as Trello were not used, there was arguably little need for such tools as the high-level aims and architecture of the project was straightforward, given well-defined aims – most of the development time was spent designing and fine-tuning individual algorithms, not switching between many different points of functionality. That being said, the use of such tools would have still likely eased the development process and potentially improved the quality of the deliverable and the number of project aims met.

Version control was employed somewhat successfully via GitHub, though only later on in the project than it should have been – use of it was delayed as earlier on in the development cycle, early code was

determined as too far from an early version and too simple to begin making use of version control. As time passed, the use was delayed until far later than it should have been. This is a regret and making better use of these tools would have likely inspired greater confidence during development of the deliverable and made it easier to report on early iterations of the project and early versions of the algorithms designed.

5.3 Evaluating Project Success

Project success can be quantified by reviewing the Project Aims. A reiteration of the aims and an overview of their status is below:

Project Aim	Status
1 - Detect the positions of the snooker balls in an image of a snooker table.	Complete, however tracking method has flaws
1a - Also detect the colour of each ball.	Complete
2 - Display the tracked information in some way that can be used to restore the balls to this state.	Complete
3 - Support video throughput as a proof-of-concept for a camera feed.	Complete
3a - Store each new shot as it is played.	Complete
3b - Allow displaying a previous table state on-demand.	Complete
5 - Explore advanced output methods such as augmented reality.	Not implemented, but considered

Figure 13: Table evaluating the Project Aims.

Throughout the planning phase of the project, there were significant issues establishing both project direction and project scope. This affected the latter stages of the project, and clearer aims laid out earlier on in the project would have likely eased the research and development processes. The poorly thought-out project aims resulted in a lack of metrics which could be used to realistically quantify the progress of the application at checkpoints throughout the development process, hurting development timelines and making the priority for development focus unclear.

To achieve clearer project aims, more time should have been spent deliberating the scope and realistically achievable goals given time and knowledge constraints. Additionally, further research into computer vision techniques and algorithms before development began would have eased the learning curve during development.

Retrospectively, Node.js should have been reviewed as a potential development platform, as it would likely have made development easier than a ‘vanilla’ HTML site, given its approach to modularity and powerful package management. The use of Node.js or some other server-based environment would have also alleviated the need for the site to be hosted using an external piece of software; while this is not a major concern due to the proof-of-concept nature of the project, it would have simplified and streamlined development, debugging and demonstration of the project.

With the above considered, the project can be considered an overall success. While some of the initial aims such as a server/client system and AR were not implemented, these aims were poorly thought out and should have been placed outside of scope from the very start. The aims presented at the start of this report represent more realistically achievable goals given time and knowledge constraints and they have been mainly fulfilled.

Furthermore, the viability and application of the platform and code library(s) chosen for use in developing the proof-of-concept application are soundly proved by meeting the majority of the project aims.

5.4 Evaluating Personal Development

Throughout a large portion of the project, I personally struggled a lot with work ethic, motivation, and focus. This has been a long-running issue that I have just about dealt with throughout my education but, I believe due to the sheer amount of thought, learning and work required, affected this project more than any other.

In attempting to tackle my issues with motivation and work ethic, I found it immensely helpful to come to the Adsetts Centre library daily. Getting out of the house, an always available change of scenery by moving somewhere else in the library helped keep up a positive state of mind, and being surrounded by others focusing on their work created a motivating space to be in. I met up with peers from my course while at the library, and it was very useful to be able to bounce high-level ideas such as which features to focus development on or how to organise report sections off others who understood the module, course, and subject in general.

I believe I could have done a much better job with establishing a clear and manageable direction for the project and defining project scope, and this would have made the project much more feasible, both from a technical and personal standpoint – at many points throughout the project I found myself getting lost in terms of what to focus on and drifting wildly out of scope when considering potential features and capabilities of the software. This should have been avoided from an early stage by spending more time planning the project, and taking the planning phase more seriously – I naively thought I had a good grip on what could and should be achieved within the time constraints, but this was proven wrong on multiple counts, as evidenced by the adjusted Project Aims in this report versus those in the original Appendix A – Project Specification. Going forwards, I will do my best to not make this mistake again be it in educational, hobbyist, or professional projects, treating the planning phase with greater diligence.

As a result of all that has happened during this project, and my recent reflection on it, I feel I have learned much about myself and how to tackle similarly large or even larger projects in the future. Perhaps most importantly, I have gained an appreciation for what it takes to create and maintain a positive and productive state of mind and stay focused and motivated.

Bibliography

- Barton, S. (2021, March 12). *Snooker BIRD'S-EYE View / Do you like it?* Retrieved from YouTube: <https://www.youtube.com/watch?v=bsJxTFPoCG0>
- Black, D. (2023). *snooker-ball-tracker*. Retrieved from GitHub: <https://github.com/dcrblack/snooker-ball-tracker>
- Blue Moon Leisure. (n.d.). *What are the markings for a pool table and a snooker table*. Retrieved from Blue Moon Leisure: <https://www.bluemoonleisure.com/what-are-the-markings-for-a-pool-table-and-a-snooker-table/>
- Boesch, G. (2023). *The 12 Most Popular Computer Vision Tools in 2023*. Retrieved from visio.ai: <https://viso.ai/computer-vision/the-most-popular-computer-vision-tools/>
- Boesch, G. (2023). *What is OpenCV? The Complete Guide*. Retrieved from visio.ai: <https://viso.ai/computer-vision/opencv/>
- BoofCV. (2023). *Main Page*. Retrieved from BoofCV: http://boofcv.org/index.php?title=Main_Page
- Brush, K., & Silverthorne, V. (2022, November). *Agile software development*. Retrieved from TechTarget: <https://www.techtarget.com/searchsoftwarequality/definition/agile-software-development>
- Emgu. (2022). *Main Page*. Retrieved from Emgu: https://www.emgu.com/wiki/index.php/Main_Page
- Kurtaev, D. (2023). *openvino_java*. Retrieved from GitHub: https://github.com/dkurt/openvino_java
- Marr, H. (2012, October 30). *Hacking on Side Projects: The Pool Ball Tracker*. Retrieved from Gocardless: <https://gocardless.com/blog/hacking-on-side-projects-the-pool-ball-tracker/>
- Nirenshteyn, D. (2021, March 17). *TrackingSnookerBalls*. Retrieved from GitHub: <https://github.com/Dimnir/TrackingSnookerBalls>
- OpenCV. (2023). *About*. Retrieved from OpenCV: <https://opencv.org/about/>
- OpenCV. (2023). *Introduction to OpenCV.js and Tutorials*. Retrieved from OpenCV.org: https://docs.opencv.org/3.4/df/d0a/tutorial_js_intro.html
- OpenVINO. (2023). *Overview - OpenVINO™ documentation*. Retrieved from OpenVINO: <https://docs.openvino.ai/latest/home.html>
- SimpleCV. (2023). *SimpleCV - What is it?* Retrieved from SimpleCV: <http://simplecv.org/>
- Spitz, J., Wagemans, J., Memmert, D., Williams, A., & Helsen, W. (2021). Video assistant referees (VAR): The impact of technology on decision making in association football referees. *Journal of Sports Sciences*, 147-153.
- Stack Overflow. (2022, June). *2022 Developer Survey*. Retrieved from Stack Overflow: <https://survey.stackoverflow.co/2022/#technology-most-popular-technologies>
- SuperAnnotate. (2021, October 7). *Top 15 computer vision libraries*. Retrieved from SuperAnnotate: <https://www.superannotate.com/blog/computer-vision-libraries>
- Suzuki, S., & Abe, K. (1985). Topological structural analysis of digitized binary images by border following. In S. Suzuki, & K. Abe, *Computer Vision, Graphics, and Image Processing* (pp. 32-46). Elsevier.
- Tamir, I., & Bar-eli, M. (2021). *The Moral Gatekeeper: Soccer and Technology, the Case of Video Assistant Referee (VAR)*. Front Psychol.
- TensorFlow. (2021, 04 22). *API Documentation*. Retrieved from TensorFlow: https://www.tensorflow.org/api_docs
- Thornton, J. M. (2022, May 31). *http-server*. Retrieved from NPM: <https://www.npmjs.com/package/http-server>

Trusov, P. (2022, April 3). *Snooker balls tracking on video*. Retrieved from Kaggle:
<https://www.kaggle.com/code/paveltrusov/snooker-balls-tracking-on-video>

Zeller, A. (2020). Coding Effective Testing Tools Within Minutes. *TAIC PART 2020*. Porto: TAIC
PART. Retrieved from <https://www.fuzzingbook.org/beta/html/PrototypingWithPython.html>

Table of Figures

Figure 1: The Pool Ball Tracker's output as of the end of the 'Hackathon'	7
Figure 2: Trusov's code labelling the colours of each ball.....	8
Figure 3: TrackingSnookerBalls generating a 2D view of the table.	8
Figure 4: The snooker-ball-tracker GUI, showing options, output and what it sees.....	9
Figure 5: A frame of example source video (Barton, 2021).	14
Figure 6: The input frame warped to a top-down perspective.	14
Figure 7: The warped input frame and rendered tracked balls.....	15
Figure 8: The tracked balls rendered in full colour with the table markings	16
Figure 9: Miniature canvases displaying each state.	17
Figure 10: A large render of one of the previous states.	18
Figure 11: A frame of video taken from a mobile phone placed as high as often possible in a casual setting.....	19
Figure 12: Figure 9 warped to a top-down view using the normaliseView() function.	19
Figure 13: Table evaluating the Project Aims.	21

Appendices

Appendix A – Project Specification

PROJECT SPECIFICATION - Project (Technical Computing) 2022/23

Student:	Joe Kirkup
Date:	18/03/2023
Supervisor:	Mike Meredith
Degree Course:	Computer Science BSc
Title of Project:	Developing an app to assist in replacing the balls after a foul in a snooker match

Elaboration

In the game of snooker, when a foul and a miss is called, the player who the foul was committed against can request the referee to replace the balls to their position before the foul shot.

Currently, on the professional snooker tour, there is no technology in place to assist with this process. The verbal back-and-forth between referee and an offscreen assistant rewinding the TV cameras has, along the years, created many a drawn-out moment and instances of players disagreeing with the positions of the balls.

The need to replace the balls in a casual / club scenario presents even more of an issue, as there are no TV cameras to rewind. Players in these situations often resort to simply never putting the balls back as the positioning would effectively be a guess, thus missing out on part of the game.

The aim of this project is to create an application which greatly eases this process, and, importantly, removes subjectivity from the matter.

In an ideal world, given more knowledge about the relevant technologies and enough time, I would produce a software solution such that:

A smartphone with my app installed can be mounted, using a tripod or similar mechanism, to look at a snooker table with its camera. The system could also be configured to use an external camera feed such as an overhead camera, with professional use cases in mind.

While the game is played, the app tracks each shot individually, keeping score within each frame and detecting the end and winner of each frame.

The app also detects foul shots, and notifies the players of such. Then, either the mounted device or another device which is 'joined' to the game via the internet can be used to display an AR overlay of the correct positions of the balls on the table, giving a signal when all balls have been replaced correctly.

After either manual termination or a pre-set number of frames have been completed, the main mounted device and all 'joined' devices will display the score, frame scores and statistics such as pot success rate and average shot time.

Obviously I would not be able to develop this 'dream system' within the timeframe of the project. For that reason, I will be developing only part of the solution: detecting the positions of the balls after a shot, and assisting in replacing them.

To clarify the above development goals: Depending on time constraints, I will attempt to implement the AR overlay for replacing the balls, but this is a stretch goal if there is sufficient time after the core functionality is implemented. If a compromise seems necessary, I will instead implement a much simpler overhead view of the table in the app, with a confirmation if the balls are in the right place. I intend to implement the server-and-client functionality.

Project Aims

Aims

- Develop a system that, using computer vision, indexes the position of every ball on a snooker table.
- Create some form of initial output to digitally display the balls in their positions.
- Extend the system to be able to store some number of shots until reset (by manual or automatic end-of-frame detection, or for some other reason)
- Host a server on the main 'watching' device which can share the game's data to other devices running the app.
- Create advanced AR output to help replace the balls (*stretch goal*)

Objectives

- Research computer vision (detecting objects in frame, and straightening an abstract viewing angle given fixed points)
- Build digital replica of a freeze-frame of an overhead view of a snooker table.
- Detect and store each shot in a video of an overhead view of a snooker table.
- Straighten an abstract viewing angle and apply existing code to create flexible solution.
- Create a client-and-server connection between two or more devices.
- Use the server to share game data.
- Implement AR view mode for replacing the balls.

Project deliverable(s)

I am going to produce a mobile application which, on one device, is configured to 'watch' the table and run a server or through some other means communicate with a client device, which can be used to help with the process of replacing the balls.

I plan to create a mobile application for Android; if compiling it as a cross-platform compatible app then I will also support iOS etc. but this is not a priority given my time and resource constraints.

In terms of engineering approaches, I will be targeting a scrum-style approach, using a Trello board and a GitHub repository throughout the project for tasks and tracking development of the deliverable, and holding short weekly meetings with my project supervisor to resolve any issues surrounding development. I will be making use of unit testing in my code where possible as I expect the codebase to be fairly complex.

Action plan

Find a supervisor.	14 th October 2022
Greenlight project idea.	16 th January 2023
Project Specification and Ethics Form	26 th January 2023
Initial research	2 nd February 2023
Information Review	9 th February 2023
Research computer vision and decide on technologies	16 th February 2023
Build digital replica of a freeze-frame of an overhead view of a snooker table.	23 rd February 2023
Create contents page.	23 rd February 2023
Detect and store each shot in a video of an overhead view of a snooker table.	28 th February 2023

Straighten an abstract viewing angle and apply existing code to create flexible solution.	6 th March 2023
Draft report and critical evaluation.	9 th March 2023
Create a client-and-server connection between two or more devices.	13 th March 2023
Use the server to share game data.	16 th March 2023
Implement AR view mode for replacing the balls.	10 th April 2023
Finalise report, evaluation and deliverable.	20 th April 2023
Project demo	20 th April 2023 – 4 th May 2023

BCS Code of Conduct

I confirm that I have successfully completed the BCS code of conduct on-line test with a mark of 70% or above. This is a condition of completing the Project (Technical Computing) module.

Signature: Joe Kirkup

Publication of Work

I confirm that I understand the "Guidance on Publication Procedures" as described on the Bb site for the module.

Signature: Joe Kirkup

GDPR

I confirm that I will use the "Participant Information Sheet" as a basis for any survey, questionnaire, or participant testing materials. The participant information sheet form is available on the Bb site for the module and as an appendix in the handbook.

Signature: Joe Kirkup

Appendix B – UREC2 Ethics Form

Please note that as of the time of submission the ethics form is not yet signed but the project supervisor has agreed to sign it upon witnessing evidence of the copyright permission, which is displayed in Appendix C – Copyright Permission for YouTube video



UREC 1 RESEARCH ETHICS REVIEW FOR STUDENT RESEARCH WITH NO HUMAN PARTICIPANTS OR DIRECT COLLECTION OF HUMAN TISSUES, OR BODILY FLUIDS.

All University research is required to undergo ethical scrutiny to comply with UK law. The University Research Ethics Policy (<https://www.shu.ac.uk/research/excellence/ethics-and-integrity/policies>) should be consulted before completing the form. The initial questions are there to check that completion of the UREC1 is appropriate for this study. The supervisor will approve the study, but it may also be reviewed by the College Teaching Program Research Ethics Committee (CTPREC) as part of the quality assurance process (additional guidance can be obtained from your College Research Ethics Chair¹)

The final responsibility for ensuring that ethical research practices are followed rests with the supervisor for student research.

Note that students and staff are responsible for making suitable arrangements to ensure compliance with the General Data Protection Regulations (GDPR), for keeping data secure and if relevant, for keeping the identity of participants anonymous. They are also responsible for following SHU guidelines about data encryption and research data management. Guidance can be found on the SHU Ethics Website <https://www.shu.ac.uk/research/excellence/ethics-and-integrity>

Please note that it is mandatory for all students to only store data on their allotted networked drive space and not on individual hard drives or memory sticks etc.

The present form also enables the University and College to keep a record confirming that research conducted has been subjected to ethical scrutiny. Students should retain a copy for inclusion in their research projects, and a copy should be uploaded to the relevant module Blackboard site.

The form must be completed by the student and approved by supervisor and/or module leader (as applicable). In all cases, it should be counter-signed by the supervisor and/or module leader and kept as a record showing that ethical scrutiny has occurred. Students should retain a copy for inclusion in the appendices of their research projects, and a copy should be uploaded to the module Blackboard site for checking.

Please note that it may be necessary to conduct a health and safety risk assessment for the proposed research. Further information can be obtained from the [University's Health and Safety Website](#)

¹ College of Social Sciences and Arts – Dr. Antonia Ypsilanti (a.ypsilanti@shu.ac.uk)
College of Business, Technology and Engineering – Dr. Tony Lynn (t.lynn@shu.ac.uk)
College of Health, Wellbeing and Life Sciences – Dr. Nikki Jordan-Mahy (n.jordan-mahy@shu.ac.uk)

1. General Details

Details	
Name of student	Joe Kirkup
SHU email address	b9026253@hallam.shu.ac.uk
Department/College	Department of Computer Science
Name of supervisor	Michael Meredith
Supervisor's email address	acesmm7@exchange.shu.ac.uk
Title of proposed research	Tracking snooker balls to assist in their replacement after a foul shot
Proposed start date	30/01/2023
Proposed end date	27/04/2023
Brief outline of research to include, rationale (reasons) for undertaking the research & aims, and methods (250-500 words).	<p>The areas I will be researching are as follows: the existing problem (replacing the balls without any visual reference or verbally guided from a visual reference), existing solutions if any and similar products, potential solutions and their viability, and finally technologies suitable for developing a solution.</p> <p>The existing problem must be explored and its effect on the game highlighted to justify development of a solution. This will involve reviewing occurrences in professional matches where play was disrupted due to lack of an adequate solution, and exploring scenarios in casual play where play was disrupted or affected by a lack of a solution.</p> <p>Existing solutions and similar products will be evaluated, in order to reflect on the needs of users, review approaches and their success, and consider applicable technologies and techniques.</p> <p>Technologies must be researched because there are a wide variety of frameworks and tools available to assist in mobile development and computer vision. This pool of options must be distilled down to appropriate and applicable choices followed by a final set of framework(s) and tool(s) chosen for use in developing the deliverable.</p>

I confirm that this study does not involve collecting/using data or samples from human participants

Please tick ☒

Below questions apply to the whole project not just research portion – recording footage in snooker hall or using TV Footage applies to these questions.

2. Research in external organizations

Question	Yes/No
1. Will the research involve working with/within an organization (e.g., school, business, charity, museum, government department, international agency, etc.)?	No
2. If you answered YES to question 1, do you have granted access to conduct the research? <i>If YES, students please show evidence to your supervisor. PI should retain safely.</i>	Yes
3. If you answered NO to question 2, is it because: A. you have not yet asked B. you have asked and not yet received an answer C. you have asked and been refused access. <i>Note: You will only be able to start the research when you have been granted access.</i>	-

Research with Products and Artefacts

Question	Yes/No
1. Will the research involve working with copyrighted documents, films, broadcasts, photographs, artworks, designs, products, programs, databases, networks, processes, existing datasets, or secure data?	Yes
2. If you answered YES to question 1, are the materials you intend to use in the public domain? <i>Notes: 'In the public domain' does not mean the same thing as 'publicly accessible'.</i> <ul style="list-style-type: none"> • Information which is 'in the public domain' is no longer protected by copyright (i.e., copyright has either expired or been waived) and can be used without permission. • Information which is 'publicly accessible' (e.g., TV broadcasts, websites, artworks, newspapers) is available for anyone to consult/view. It is still protected by copyright even if there is no copyright notice. In UK law, copyright protection is automatic and does not require a copyright statement, although it is always good practice to provide one. It is necessary to check the terms and conditions of use to find out exactly how the material may be reused etc. <i>If you answered YES to question 1, be aware that you may need to consider other ethics codes. For example, when conducting Internet research, consult the code of the Association of Internet Researchers; for educational research, consult the Code of Ethics of the British Educational Research Association.</i>	No

Question	Yes/No
3. If you answered NO to question 2, do you have explicit permission to use these materials as data? <i>If YES, please show evidence to your supervisor.</i>	Yes
4. If you answered NO to question 3, is it because: A. you have not yet asked permission B. you have asked and not yet received an answer C. you have asked and been refused access. <i>Note You will only be able to start the research when you have been granted permission to use the specified material</i>	-

3. **Does this research project require a health and safety risk assessment for the procedures to be used?** Discuss this with your supervisor and consult the [Risk Assessment Toolkit](#) for teaching research.

☐ Yes

☒ No

(If **YES** the completed Health and Safety Risk Assessment form should be attached). You can find a [Blank/Sample Risk Assessment Form](#) at the Checklist, Generic and TORS Risk Assessments on the [Risk Assessment Toolkit](#)

Adherence to SHU policy and procedures

Ethics sign-off	
Personal statement	
I can confirm that: <ul style="list-style-type: none"> I have read the Sheffield Hallam University Research Ethics Policy and Procedures I agree to abide by its principles. 	
Student	
Name: Joe Kirkup	Date: 10/03/2023
Signature: Joe Kirkup	
Supervisor or another person giving ethical sign-off	
I can confirm that completion of this form has confirmed that this research does not involve human participants. The research will not commence until any approvals required under Sections 2 & 3 have been received and any health and safety measures are in place.	
Name:	Date:

Ethics sign-off	
Signature:	
Additional Signature if required:	
Name:	Date:
Signature:	

Please ensure that you have attached all relevant documents. Your supervisor must approve them before you start data collection:

Relevant Documents	Yes	No	N/A
Research proposal if prepared previously	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Any associated materials (e.g., posters, letters, etc.)	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Health and Safety Risk Assessment Form	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Appendix C – Copyright Permission for YouTube video

Video Usage Request

3 messages

Joe [REDACTED] 15 April 2023 at 12:27

Hi Steve,

First of all, thank you for all the free educational content you provide on your channel!

I am a big fan of snooker and for my final year project at university I'm developing a proof-of-concept app to help replace the balls after a foul.

I am emailing you to request permission to use your "Snooker Bird's-Eye View" video (<https://youtu.be/bsJxTFPoCG0>) in my project demonstration (a private presentation to my two markers) and to use individual frames of it in my project report to demonstrate the functionality.

Thanks for your time,
Joe

Steve Barton [REDACTED] 15 April 2023 at 16:56

Hi Joe,

Thanks for the message!

Yes, that's no problem. Please feel free to use that video.

Good luck with your project 🍀

Steve
[Quoted text hidden]

Joe [REDACTED] 16 April 2023 at 14:20

Thanks Steve - for the quick response and the permission!

Cheers,
Joe
[Quoted text hidden]