International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 13, No.1 (Mar-23)

http://dx.doi.org/10.12785/ijcds/130144

Planet Adventures: An Augmented Reality Game using Facial Emotion Recognition

Aaliq Moin[1], Lohitha Kolli[2], Venkatramaphanikumar Sistla[3], Venkata Krishna Kishore Kolli[4]

[1,2.3.4] *Department of Computer Science and Engineering, Vignan's Foundation for Science, Technology & Research, Guntur, India*

E-mail address: alienaaliq@gmail.com, lohithakolli1999@gmail.com, svrphanikumar@yahoo.com, kishorekvk_1@yahoo.com

Received 11 Jun. 2021, Revised 13 Dec. 2022, Accepted 5 Mar. 2023, Published 16 Mar. 2023

Abstract: Augmented reality is now on the forefront of technological advancements around the world with major consumer electronic companies like Apple, Microsoft, Ninatic, Lucyd, Augmented Reality Labs etc., investing in it to develop an AR interface for its users. In this work, authors proposed a game using various technologies in sync with augmented reality, visual scripting, and facial gestures without use of any complex devices. Facebook's SparkAR studio with integrated AI models are used in this proposed framework to recognise the face gestures based on DeepFace. The game works well along with all the individual systems like the collider system, point system and other parts in sync and a little bit of JavaScript code as well to display the points on the user's screen. The proposed AR game overcame the limitations of other AR games by using light weight and efficient technologies like visual scripting and 2D assets and eliminated the use of high-level hardware. A survey on user experience have been done using beta version and found good performance when compared to other similar AR games.

Keywords: Augmented Reality, Visual Scripting, SparkAR, Emotion Recognition, JavaScript Bridging, DeepFace.

1. INTRODUCTION

Augmented reality [1] is a way of implementing an immersive experience by adding interactive components to the real world. It also helps the users better understand concepts and is generally more immersive than normal visual perception. Augmented Reality has been an evolving technology for the past 20 years and also the recent advancements within the field have made it more accessible and more entertaining to the end-users. It has brought innovation to the field of research and entertainment. In the area of gaming, augmented reality provides unique opportunities. Augmented reality is an upgraded version of the real physical world that is applied through the utilization of digital visual factors. At present, it is a growing trend among business applications and mobile computing. This mobile augmented reality is a tailor-made augmented reality. It is triggered by just pointing a device at a print image, logo, or object. It will be operated on iOS or Android applications. Unlike traditional video games, these mobile AR games don't seem to be imprisoned within the screen space but also provide interaction with the encompassing world.

In mobile augmented reality games [2][3], the realm for taking part in becomes borderless and literally that they may be played anywhere and anytime. The ongoing advancement of recent mobile phone technology is finally giving exposure to more people than ever as the hardware necessary to participate and skill such games. In general, there are three dominant platforms to create augmented reality games on Unity, SparkAR Studio and, Snap lens Studio.

Unity: Unity is one of the world's largest frameworks for immersive game content creation which serves and offers tools for producing amazing games and publishing on a broad spectrum of devices. The main platform of Unity provides a more effective and efficient design of games, but it needs a separate platform to publish a game and it does not support many languages to code the game including JavaScript.

SparkAR studio: SparkAR is a project funded by the social giant Facebook and it has a vast library and rich features to use in game development. The platform was originally intended to create Instagram or Facebook filters with which many people are familiar. These days it has gained a lot of traction with its ever-expanding user base and features. Many popular brands are using it to create AR experiences for their brand as a part of their internet advertising. It has many useful AI features as well that can help in facial recognition and gesture recognition. It also has a visual scripting tool called patch editor which can help us code the game in real-time instead of compiling code every single time and checking for errors. Another highlight of this software supports JavaScript natively.

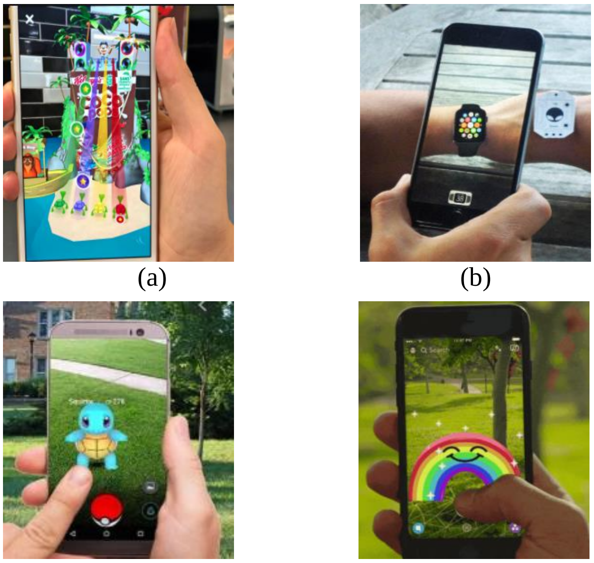
Snap lens studio: Snap lens has modern features like hand gesture recognition and segmentation that can be used to create a lot more immersive game content. It also supports JavaScript and also a web-based API available to publish the game without using any other platform. Based on the above feature set, the authors intended to demonstrate their work on SparkAR Studio rather than the other two platforms.

1.2. Methods of Making AR Games

There are various methods to choose from when developing AR games like marker-based, world-based and face-based games.

Marker-Based Games: Marker-based AR games have a scanning algorithm in place that needs a particular pattern or code to be detected in the viewfinder to start the game or AR effect. Many examples of marker based AR games are available nowadays like the QR codes on cereal boxes that need to be scanned and give specific rewards when scanned. There have also been instances of toys, which give a game item with a specific tag when scanned.

World-Based Games: World-based AR games [5] [6] have a similar algorithm to the marker-based ones, but instead of searching for a target, they search for a plane on which they can start the effect like floors, walls, etc. They work by utilizing the depth information given by the camera to sense how far away a plane is and how leveled it is to check if the effect or object in the AR experience can actually fit in that plane. Some AR developers add the option of scaling for both planes and objects so that the objects can be scaled up or scaled down to be placed in multiple world planes of various sizes. This kind of AR effect is also very popular in branding and advertising of various products like furniture as presented in Figure 1.



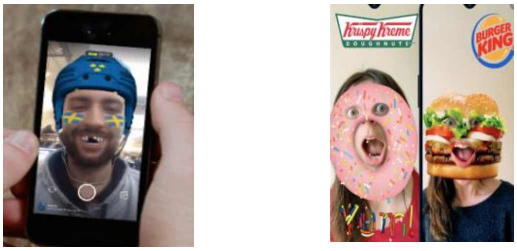
(a) (b)

(c) (d)

Figure 1(a): *A marker based AR game on a cereal box* (b) *A marker based AR game augmenting a watch (c) A world based AR game (Pokemon GO) and* (d) *A rainbow augmented in world scene*

1.3 Face Tracking

Face tracking-based AR experiences [8][9] work on the concept of using pattern recognition and computer vision techniques to recognize and track the user's face(s) and overlay multiple effects and objects into the scene. Sample games with face tracking are depicted in Fig 2.



(a) (b)

Figure 2: (a) *A face filter through AR augmenting a helmet* (b) *A segmentation-based face filter*

This approach is stable and is not resource intensive when compared to other AR techniques as it is mainly used to create AR filters that can improve a user's looks on camera. In this game, facial expressions will be used to trigger the actions to control the in-game objects.

1.4 Resource Gathering

For any functional game [4][5], we need to have various Visual, functional, and sound assets added to the game as follows:

Visual Assets: All the visual assets for the game like the background, the player, obstacles/enemies and, effects are designed in Photoshop by using 8-bit pixel art by taking inspiration from the classic runner games and have been exported as PNG and compressed to minimize their size.

Sound Assets: The sound assets have been taken from no-copyright sources which suit well with the overall theme of a runner game in addition to sound effects for attacks, background music etc,.

Functional Assets: Functional assets like the code and logic have been developed by the team and have certain non-editable features which cannot be changed without a proper understanding of the implementation and logic. The game is designed on the base of JavaScript.

2. Related Work

In this section, a thorough study is presented on AR games, case studies on other AR projects and, AR frameworks with complex requirements.

2.1 Case studies on various AR games

A case study on "Pokémon Go" [6] shows how appealing a simple game can be in interactive with the real world.

AR just takes the step of imagination out of any outdoor game and makes it look real. This has played a major part in the revival of the Pokémon trend which was slowly fading away after it saw its peak in the early 90's, this just shows how an AR game can be exponentially more interesting than a normal game if implemented correctly. In this game, 3D animations were used to serve as targets, and as it was a World based AR game. There was no need to have a constant trigger to play this game and it need a stable internet connection and a phone.

Mario Kart a simple AR game developed on the concept of mixing robotics with AR was developed in 2019-2020 and has a huge worldwide audience. Authors [16] demonstrated the concept of an augmented coliseum with the help of robots and AR. In this work, the authors used imaging algorithms to produce a virtual environment for the user and then combined the robot with Bluetooth sensors to receive directions. The Pac-Man game [18] was developed to understand consumer satisfaction with AR games. This Pac-Man game extends immersive aspects and a new dimension of interactivity. The game was recreated in AR using marker-based tracking of specific objects such as power-ups and enemies which were all seen as projected in the virtual HUD of the head contraption.

Capture the flag [19] is another AR game, where two teams were identified by their tags and they had to capture the opposite team's flag which was also identified by a unique id tag. If a team is succeeded to get the opponent's flag to their base, then they would win the game. This game was very popular in developing social skills along with competitive team spirit among the children.

The networked AR multiplayer game [20] played online and considered as a Real tournament distinguished itself from the competition by providing multilayer 2D and 3D on-screen projections of opponents and targets. It had its specific designed handsets, algorithms, infrastructure, and security features for its players. A present-day game with a similar implementation is "Ingress". Authors [2] demonstrated augmented reality from scratch with the implementation of google maps, to provide a 3D view of the surrounding area with a feature called "Street View" and the fact that the whole world could be mapped inside of a device with rich details and directions.

These researchers [4] suggested the cultural heritage after it is enhanced by augmented reality. With technology development, cultural heritage is not expected to pass down, but augmented reality can also do wonders by enhancing one's interest in a seemingly dull topic such as history.

The impact of AR on learning [7] is elaborated by using RFID tags for triggers and trackers in the place of Bluetooth to save energy consumption.

2.2 General Research projects on AR

In on the recent next-gen mobile AR games [1], the authors reviewed existing AR approaches towards small scale and user modifiable games to large scale event-based games. This creates an immersive experience for the user that provides a gaming experience in the comfort of their own homes with just a capable mobile phone.

A group of game developers suggested various methods to use augmented reality and virtual reality together as a concept of mixed reality. They [3] suggested that there are gaps that can be bridged using reality altering immersive systems in day to day lives. Authors of [8] presented the difficulties and responsibilities involved in developing an AR game and how it actually looks by sharing the snapshots behind the development of old AR games. Augmented reality is a powerful tool to focus more on the game immersiveness and gives better experience than traditional games [9]. Authors of [10] implemented a touch less interactive game with a camera to get the input and led the screen to show the output. They used various AI algorithms to do so the segmentation and motion tracking of the user's hands and/or face.

Augmented reality is a technology through which we can add something to the existing environment and it doesn't require much complicated setup. On the other hand, VR takes us into another space altogether by replacing the surrounding environment [11]. Virtual reality and augmented reality can both be pretty similar and yet be the quite opposites of each other, to decide which one would be more suitable for our project has been explored. VR does sound more immersive at first glance but the complicated requirements and limited capabilities of VR make it quite inferior to AR at least in terms of gaming and accessibility.

Study has been carried out on a group of kids who were interested in gaming. They suggested very drastic improvements both in the experience and game play of those in AR version of the same PC game. A total of 44 students were recruited for this test and they were told to try out a shooting game on PC and then the same game but in the AR version. The results showed that not only the students performed really well in the AR counterpart, but they rated their experience higher on the AR version compared to the PC version [12]. A paper tried to consolidate the requirements and capabilities of AR by providing statistically sound facts about AR, it compares various genres of games like racing, shooting, defensive and strategic games and they all while being different games and having very different algorithms and implementations had very identical hardware requirements. They mostly required a camera and a scene to track or place objects [13]. Marker based or target based augmented reality is a sub part of AR and in a journal it was explained how it actually works. The computer vision algorithm would constantly be scanning or specific pattern in the scene by using the feed taken by the camera and then when the marker specifically programmed in the system is detected it acts as a trigger to start showing the augmented objects around that specific target, which was a clever implementation of pattern recognition with computer vision and it was a strong competition to the geo-location based AR games like Pokémon Go [17].

2.3 Research on AR frameworks and complex systems

A team made a case study towards augmented reality systems by studying its un-obstructive integration into daily activities from playing games to doing non-gaming activities and then reviewing the key moments of their activity in AR. Their case study mostly involves the traditional AR frameworks like AR GO and AR Drum kit. [5]. Unity a popular game engine also has very capable AR features that can be used to develop AR games and experiences. In a paper they showed how unity framework is used to design AR games and experiences by using “Vuforia” as the AR kit. A comparison was made between a first gen independent AR prototype and a similar counterpart developed using Unity engine. The comparison showed that unity was a major improvement as it was using a lot if image processing algorithms at once as a part of its “Vuforia” tool which led to about 55% improvement in total experience of the user [14]. EMMIE is a multi-user augmented reality system developed by a group of computer vision engineers. It was used to develop a near-game like project called MARS which was an indoor-outdoor interactive UI application which was an experiment to check and demonstrate the capabilities of their system which used head-worn instruments which had a camera to map the environment or User interfaces which were enhanced tablets connected to satellite feeds. This system was also considered to a part of US naval research program [15].

2.4 Facial recognition algorithms

Facebook published a paper on their AI which could compete with state-of-the-art technology in facial recognition and named it DeepFace. It used Deep neural networks with 120 million parameters to convert a 3D face which may not be facing towards the camera into an accurate representation of the face and a flat 2D representation which could be used to easily extract facial features [21]. In 2016 Facebook researchers published another article on how they are implementing a new technology called DeepMask and SharpMask which can be used to segment objects in a scene and extract necessary details. SharpMask as it was name performed slightly better and provided a sharp mask overlay on top of the targets as compared to the DeepMask both of which are being used in their popular AR platform SparkAR [22]. From the above we can understand the limitations of previous Augmented reality applications especially in gaming. They either required complex equipment like a HUD and other controllers or required very high computation power which cannot be delivered by a handheld device. We plan to solve these problems by proposing an idea of a light weight AR game that doesn’t require high computation power or complex hardware devices like headsets [23-25].

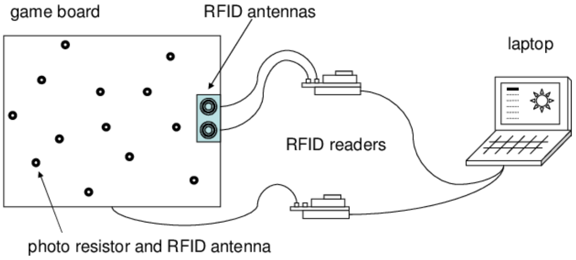


Figure 3 (a) – Older AR game versions with sensing devices.

Figure 3 (b) – Recent AR games in mobile phones

In old approaches the players would have to wear a complex set of devices that would complicate the setup process to play a game which included wireless devices that would act as tags or targets, sensors, scanners and some other devices which would send signals that would be compiled in a central processing hub with the game is shown in Figure 3 [26]. To understand better about what we have done differently and how our method is different we must look at a brief comparison of various modern AR tools. To start with the major ones in the AR game or any gaming genre would be Unity and Unreal Engine which are the go-to for any game developer. But, they lack various features when it comes to deployment of a game. For instance, if someone wanted to develop an android game, they must specifically choose the framework that only caters to android, and IOS would be left out and vice-versa. They would have to build a separate version of the same game for different platforms which consumes a lot of time and resources. By using spark AR as out framework we eliminate any such need as it is cloud-based and we can easily import the same game on various devices as long as they support social media apps like Instagram and Facebook. And for a light-weight concept implementation like our game, Spark would be the perfect choice. To conclude this section, Unity and all the other available frameworks used by other developers are great and work very well with conventional game packages but they are not made to design on-the-go games which would be played instantaneously on social media without any download required [27-28].

3.PROPOSED PLANET ADVENTURES GAME

Based on feedback from the previous implementations and research done by many people on the subject. The augmented reality is an immersive way to develop games which are more enjoyable in general compared to normal games. A lot of choices available to choose from to make the AR game like 3D, 2D, marker based, world based or face based. Some research has been carried on the available methods and algorithms to choose the one that would be well suited for our proposed application is presented in Figure 4. After reading and understanding the systems designed by others and the idea of system is proposed with the objectives of user's face and expressions based controls such as simple run, attack, dodge game. A collider system also included that can detect any collisions with obstacles and a reward point system that maintains player score and changes the levels accordingly. It will also have a visuals and sound module similar to any game. The following algorithm describes in terms of how it runs the game and some important modules like collision and point system [29-30].

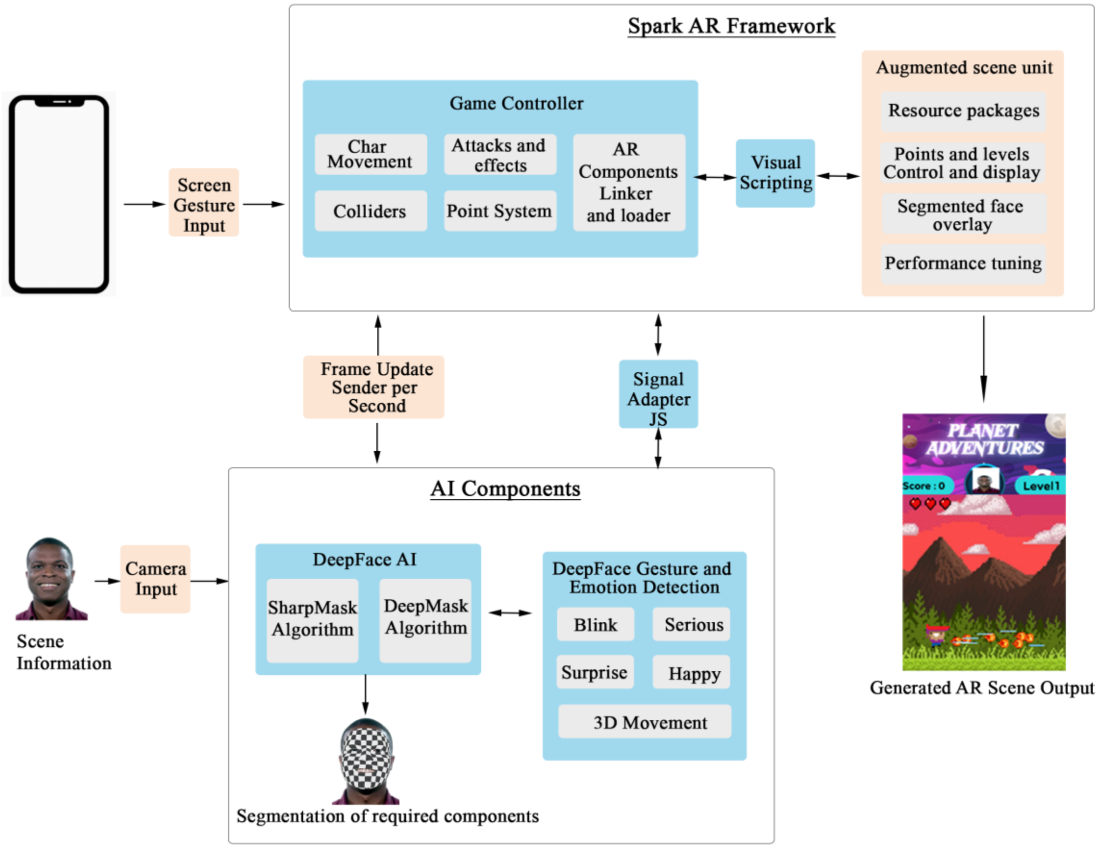


Figure 4: ***Architecture of the Proposed Work***

3.1 The AI behind Face and Emotion Recognition

The AI behind the all the facial recognition features in this AR project have been provided by Facebook's powerful integrated face tracker that comes with the SparkAR framework. It's called DeepFace, which closed much of the distance in AI based face recognition and is on the verge of becoming the most popular criterion. It has near human standard precision in understanding faces and it has been tested on a large dataset of faces when it was first introduced it could exceed the systems available at the time which had a very limited performance.

The algorithm delivers a highly significant and Compact face picture, which is impressive when compared to other available AI face recognition systems

The common pipeline of a facial recognition system is *Detect – Align – Represent – Classify* and this algorithm revisits the steps of alignment and representation to improve their performance.

The algorithm starts by detecting the 6 fiducial points from the viewport of the camera feed, which are approximated at the center of the eyes, tip of the nose and mouth location. They are used to scale, rotate and translate the image into six anchor points or locations

After this an algorithm uses its training based on the 3D scans from the USF Human-ID database which helps in adding 67 fiducial points to the output from the previous step with the help of post-processing, this lets us translate the data from 2d and 3d planes observed from the same camera perspective. This enhances the both the segmentation and recognition of faces so that they can be recognized more accurately. Then a formalization technique is applied to ensure that proper normalized output is achieved. i.e., even when there is a picture of a person slightly turned or tilted towards one side. The algorithm uses the 67 fiducial points to render the portions that are invisible in the scene. Without this step the produced output would be a warped image of the persons face. DeepFace is an open sourced and lightweight face recognition approach with leading AI models. DeepFace provides an approach for facial segmentation and recognition in the background. It is a language independent package and having functionalities such as covering new facial attribute models with score prediction, wrapping new facial recognition models works on a Cloud API [32]. Stepwise working process of DeepFace algorithm is presented in Figure 5.

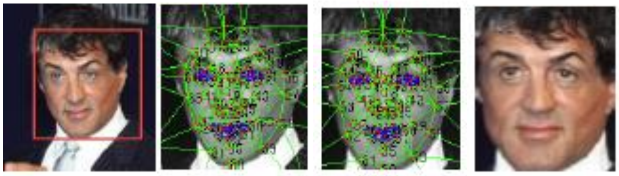


Figure 5 ***Stepwise working of DeepFace algorithm***

In addition to this algorithm, *DeepMask* and *SharpMask* are added to segment in next step and mask the object or the target in a scene that are based on feed forward and CNN architectures. It can be used to project other materials on to the scene or to extract a certain part of a scene. The DeepMask algorithm is a based on a Feed Forward + Skip architecture is presented in Figure 6. It segments the target by compressing down the pixels in the original picture into half of its size on each step and retaining only relevant information of the target in the scene. This is further enhanced by adding refinment steps in between each compression. This sharpens the output from previous step before giving it to the next one. More of the relevant information is retained with effective mask.

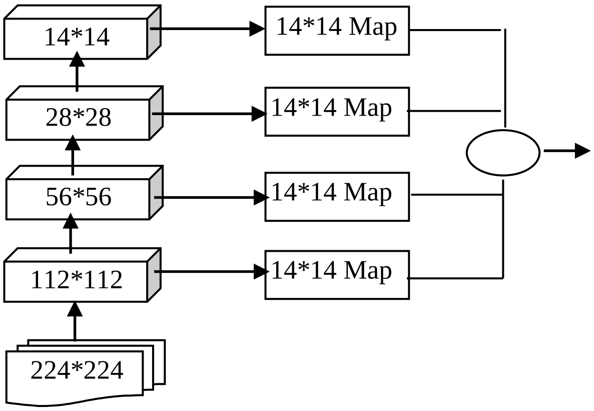


Figure 6(a) ***A basic flowchart of deepmask***

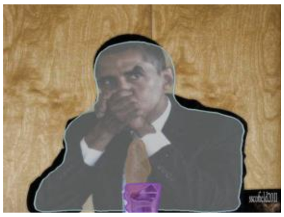


Figure 6(b) ***Result of deepmask***

The Model further works as an emotion classifier and it has been trained on over 260K samples by the facebook team.

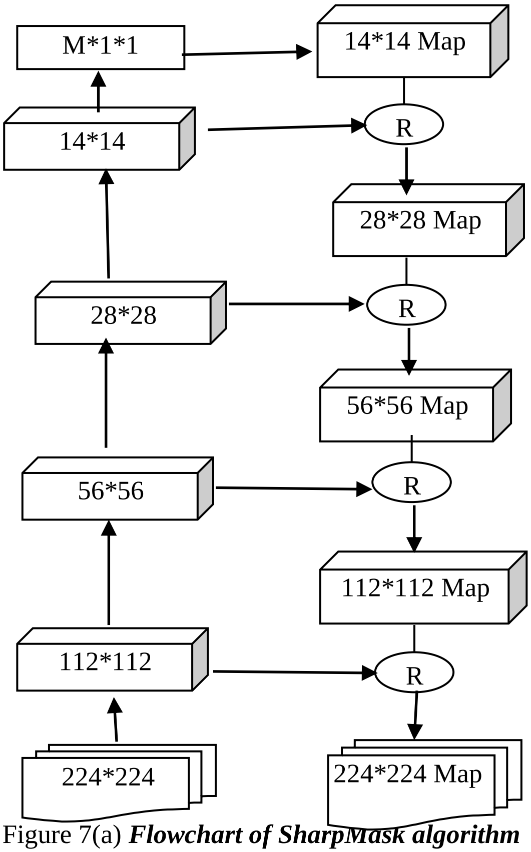


Figure 7(a) ***Flowchart of SharpMask algorithm***

Figure 7(b) ***Result of sharpmask***

It calculates the distances between each of the prominent fiducial points on a selected face and then checks for other inputs like mouth open, eyes open and other face gesture states. The model used these features to classify emotions and architecture of SharpMask is presented in Figure 7.

3.2 Implementation of Collider and point system

The Collider and Point system in the game is partly implemented using visual scripting and partly using the traditional coding in JavaScript as the software doesn't yet support the updating of frames and values simultaneously. The collider system presents at Figure 8(a) sends a random obstacle at the user at a random interval but at the same uniform speed on each level.

The random obstacle has a Vector3 value as it is implemented on a 3D plane which has X, Y and Z values The Character/Player also has a similar implementation on a 3D plane and has a similar set of Vector3 values.

At each frame the compare method in the collider system is called to check if the values form the X and Y-coordinates of an obstacle and the character are matching.

If match occurs, then a collision is triggered and the signal is sent to the collider output. The character can however choose to avoid the obstacle by two methods.

1- By skipping the obstacle and jumping over it (In which case the Y value will not match and collision will not be triggered) and no points will be obtained.

2- By attacking the obstacle with fire or laser (In which case there will be a relaxation of 4 seconds for the user and points will be given for each destroyed obstacle).

However, there are certain non-skippable obstacles like walls that cannot be skipped and need to be destroyed in order to not lose a life.

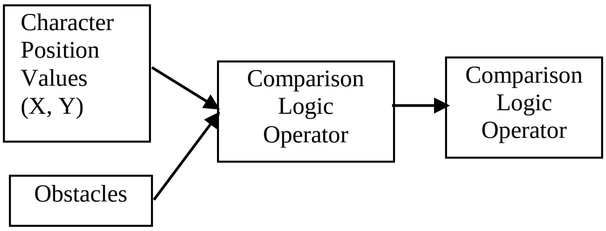

Character Position Values (X, Y)
Obstacles
Comparison Logic Operator
Comparison Logic Operator

Figure 8(a) ***Collider system logic***

The Coordinates of object, character and wall obstacles are sent through the collision checker to see if any collision occurs at those particular coordinates. If there is an overlap of location coordinates between the character and any obstacle, then the points system counts a point each time. It receives the pulse output from the attack active state and the collision system along with a slight accommodation for time delay if required to eliminate false points. Then it will be sent to a counter to store and display the current score on the screen. The points system in the game is responsible for displaying the score, changing the level and increasing the difficulty of the game with each level. The point system is partly dependent on JavaScript to display the score to the user presented in Figure 8(b). As the snapshot in Figure 9 shows, how a JavaScript linked to the visual scripting tool for displaying the score and level values on the screen for each updated frame.

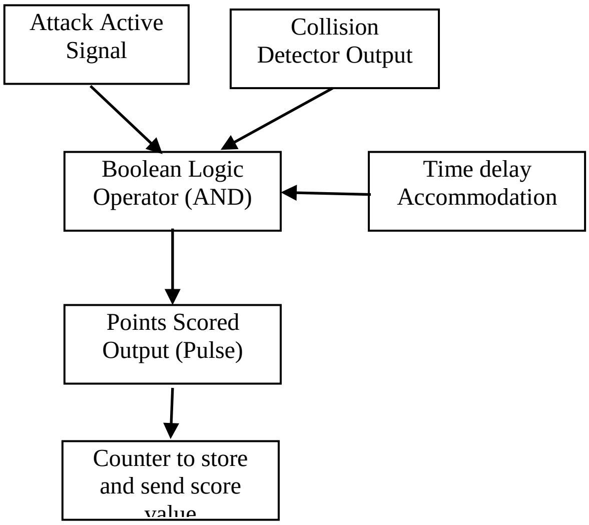
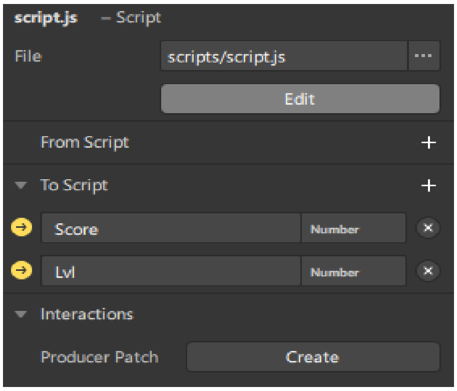

Attack Active Signal
Collision Detector Output
Boolean Logic Operator (AND)
Time delay Accommodation
Points Scored Output (Pulse)
Counter to store and send score value

Figure 8(b) ***Point system logic***



script.js – Script

File scripts/scriptjs ···

Edit

From Script +

To Script +

Score Number

Lvl Number

Interactions

Producer Patch Create

Figure 9: ***linking scripts to patch editor***

The difficulty of the game increases as the game progresses and it is based on the simple *speed = distance*

/ time formula. As the player reaches a new level the speed increases by 15% each time.

The distance in the formula is the distance that needs to be covered by the randomly spawned obstacle from right to left on the screen and the time is a universally fixed value based on the spawn time of the obstacle.

4. EXPERIMENTAL RESULTS & DISCUSSION

Our approach was to implement a casual runner game with facial expression controls by using augmented reality and visual scripting. The game works well along with all the individual systems like the collider system, point system and other parts in sync and a little bit of JavaScript code as well to display the points on the user's screen. The face detection and masking works well. It has a transparent overlay that won't be visible to the user in the game. It is only used for tracking the face and detecting face gestures like blinks, head nods and other actions. It could send a signal when an emotion is shown like happiness, anger and surprise on the user's face. These triggers are then used to activate attacks for the character in the game using face mapping presented in Figure 10. Each of the three expressions activates a different attack such as Fireball attack with Happy face, Laser attack with Surprise face and Shielding with Angry face using various emotions presented in Figure 11.

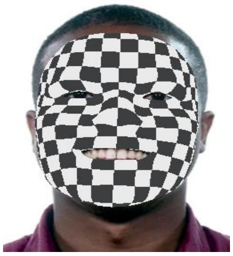


Figure 10 ***Face mapping***

A blink gesture detection is included to trigger a jump in the game, so that the character can dodge some obstacles. No points will be awarded for dodging an obstacle. The character in the game also moves left or right based on the user's head movement. The below image shows the activation of various in-game actions triggered by user's emotions.

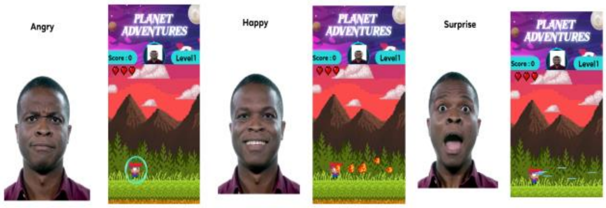


Figure 11 ***Showing various attacks triggered by emotion state of the user's face***

Note: The Person in the examples is the default actor video provided with the software

The next module was the movement module and it is built using the patch editor was presented in Figure 12. The logic is to normalize the head rotation value of the user along X-axis to better match with the sensitivity and speed of the character's movement. Limiters have been placed on both sides of the screen to bound the character's movement on the visible portion of the screen without letting it out of the screen.

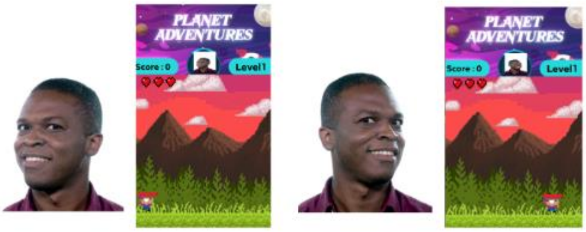


Figure 12 ***Movement of character based on user's head rotation***

The Point system has also been implemented via visual scripting is presented in Figure 13. It takes the attack active inputs, which are high when fireball or laser or a shield is on. A collider input checks if the user has crossed a destroyed obstacle and awards the points.

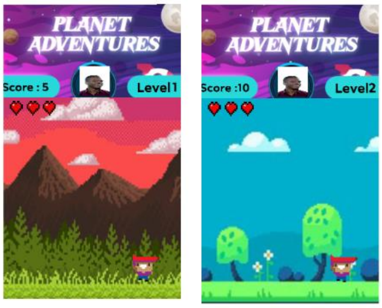


Figure 13 ***Point accumulation and level change***

For each ten points accumulated, the level in the game changes and the character reaches a different planet which has a 15% higher difficulty level. This is implemented by signaling the counter each time by using the formula Points % 10, which gives zero as output each time the score is in the multiples of 10. The collider system is the heart of the game which makes a signal whenever the character hits an obstacle and sends the output to decrement the available lives for our character. There are three lives in the game and each time the character touches an obstacle, they lose a life and after losing all three lives, a game over screen is displayed to the user. Then, the touch event lister is active again and the game can be restarted by double tapping the screen. Collider system and game over screen is depicted in Figure 14.

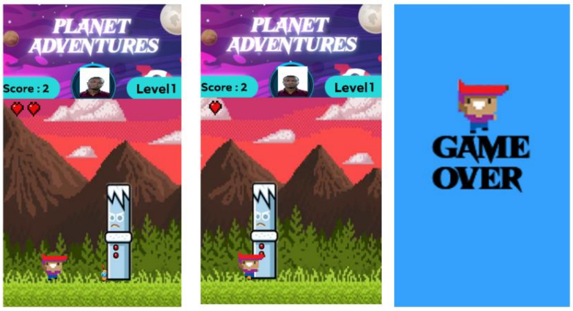


Figure 14 ***Collider system working and game over screen***

This game also contains minor bugs and glitches, which will be resolved in later versions. One such glitch in the game is, when the user is transitioning from one expression to another, occasionally triggers both attacks at once. However, it doesn't allow the user to get more points and is just a visual glitch.

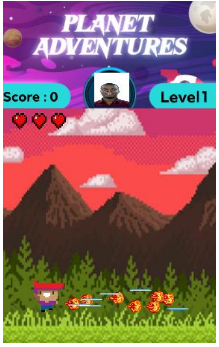


Figure 15 ***showing both attacks being triggered at once***

When the game doesn't detect a face in the scene a message "Find a face" is displayed to the user as per Figure 15.

Find a face!

Figure 15: Alert as a Find a face message

Simulation and testing results

The game is tested in various environments and on different levels of hardware and software systems. The performance of the game is measured using an FPS test which tells us how many frames a processor is able to generate per second and 25 FPS is generally considered playable. Performance evaluation of proposed game on various mobile processors is presented in Table 1 and Figure 17.

Table 1: Testing on Various Mobile Processors

Processor - level	Clock speed	GPU	Framerate obtained
SD 710 – midrange	2.2Ghz	Adreno 616	35 FPS – Stable
SD 660 – low end	1.84Ghz	Adreno 512	25 FPS – Stable
MTK P35 – very low	2.3Ghz	PowerVR GE8320	20FPS – Unstable
SD 865 – High end	2.84Ghz	Adreno 650	60 FPS – Stable
SD 660 - Low	1.84Ghz	Adreno 612	25FPS - Stable
MTK P35 – Very Low	2.3Ghz	GE8320	20FPS – Unstable
SD 710 – Midrange	2.2Ghz	Adreno 616	35FPS - Stable
SD 675 – Midrange	2.0Ghz	Adreno 612	30FPS - Stable
SD 712 – Midrange	2.3Ghz	Adreno 616	35FPS - Stable
SD 460 – Very Low	1.8Ghz	Adreno 510	20FPS – Unstable
MTK D1200 – High End	3Ghz	Mali G77	60FPS – Stable
SD 865 – High End	2.8Ghz	Adreno 650	60FPS – Stable
SD 888 – High End	2.8Ghz	Adreno 660	60FPS - Stable

Here SD – Qualcomm Snapdragon Processors and MTK – Mediatek Processors. For gathering the performance data, a feature called USB debugging included on all android devices that enables the monitoring of frame rate for any given application. To run an AR application, the GPU of a processor shall be capable to play a framerate of at least 24 FPS. Most Midrange and high-end processors are capable of handling such tasks and few low-end devices of years old also give us a playable FPS with manageable stability. The only devices that cannot handle the load are low end devices, which did not have the GPU or CPU capabilities to provide a stable AR experience.

In-house testing has been conducted and results of the devices are available to us at the time and may vary on various devices based on their configuration and other features.

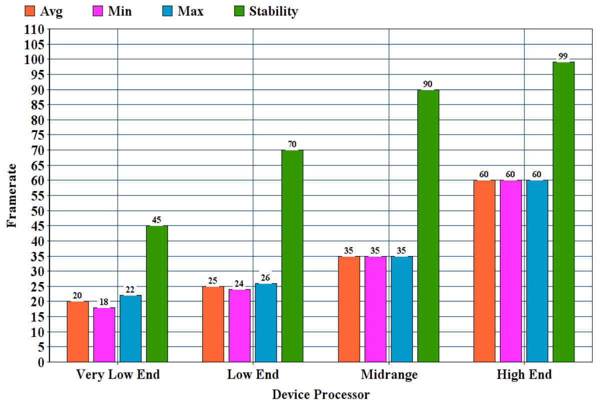


Figure 16 –Test results of AR game – planet adventures on various device configurations

However, our game logic system works mostly by recognizing the user's facial gestures which have been implemented using Spark AR which makes our game stand out from other such AR games available. The issue with non-web-based AR games or other non-web-based games in general is compatibility and optimization with various devices available in the market, some games require developers to specifically optimize them for certain devices so that they perform well enough on them. One such example is that android and IOS devices deliver different results for the same game, which is not because of the different hardware but the optimization done by game developers. This adds extra steps in the development cycle of games and is also creating a monopoly with the big brands as newer start-ups cannot afford to get developers to optimize apps and games for their specific devices.

On the contrary, a web based game has the same level of performance on all the devices and doesn't require any major optimizations. Keeping these in mind we designed an application which is just 2D example of an AR game as 3D games with more objects and assets would not perform well on some devices due to network connectivity and internet speed issues. This approach is a step towards web based gaming which could possibly run full 3D games in the future with 5G connectivity coming soon. There are some 3D AR and web based games already available but they tend to have some hardware and connectivity requirements as well.

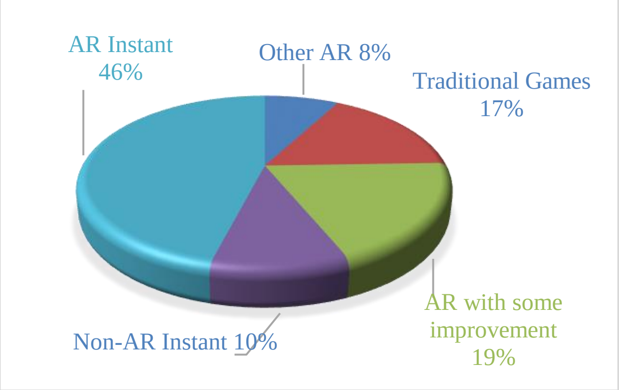


Figure 17 – Poll on preferred game genre on Social media

From the results depicted in Figure 17, it can be deduced that players are preferring instant AR games instead of traditional games which require some form of a download. The next survey further clarifies the user satisfaction of the game Planet Adventures. The Pie-chart shown in Figure 18 the satisfaction and feedback of people after trying out the Planet Adventures and over 50% of people liked the game and a quarter of the population wanted improvement in the graphics department.

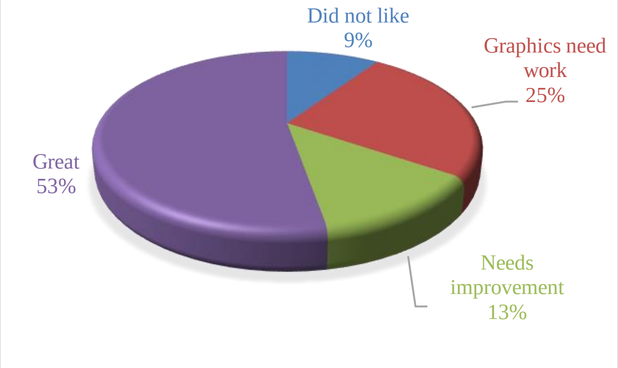


Figure 18: Satisfaction and Feedback analysis of people on proposed Planet Adventure game

The most modern AR games are 3D, require a fast internet connection for smooth play. The 2D games available to perform similar approach as they are built around the same Spark AR framework. To play some popular 3D AR games, many low-end devices are not even supported and tabulated in Table 2.

Table 2: Performance Evaluation of proposed game with various devices

Game	Genre	Low	Mid	High-end
Pokemon Go	3D	NA	33	60
Jurassic World	3D	20	31	60
Ingress	3D	NA	35	60
Knightfall	3D	18	27	55
AmongUs IG	2D	17	30	60
Minecraft IG	2D	20	26	60
Minecraft Earth	3D	NA	30	60
Angry Birds AR	3D	15	33	60
Planet Adventures	2D	25	32	60

In all of the above tests the framerate limit has been removed and average of consecutive runs has been taken for each game on each tier of device.

This attempt at making a fully functional AR game using AI and AR with features like face gesture and emotion recognition seems to have achieved its goals as an attempt to experiment with various features and create an enjoyable game. The features like graphics and gameplay can be improved and optimized with future updates as the platform and technology matures and more developers start contributing to the AR community.

5. Conclusion & Future work

The goal of creating a simplified version of an AR game has been achieved. It runs mostly on any hardware as long as it is a mobile device with internet and camera capabilities. The game has some minor bugs and glitches which will be fixed in upcoming versions. The AR game is controlled via the user's facial gestures and expressions complete with attacks, visuals, sound system and a score system which can be played on a smartphone having an internet connection. This implementation of an AR game has overcome some of the limitations in other AR games made in the past by using light weight and efficient technologies like visual scripting and 2D assets and eliminated the requirement of high-level hardware. The game was tested by some volunteers as a part of beta-testing, and received mostly positive reactions. The only critical points being lack of any 3D objects that could improve the visual aspects of the game while also increasing the requirement for computing capabilities. The total size of this game comes around 2 MB to 2.5 MB depending on the platform. From the survey, it is clearly evident that 50% people liked game and over 25% of people shared that there is a need of graphics department.

6. References

[1]. Broll, W., Lindt, I., Herbst, I., Ohlenburg, J., Braun, A. K., & Wetzel, R. (2008). Toward next-gen mobile AR games. *IEEE Computer Graphics and Applications*, *28*(4), 40–48. https://doi.org/10.1109/MCG.2008.85.

[2]. Wetzel, R., Blum, L., Broll, W., & Oppermann, L. (2011). Handbook of Augmented Reality. *Handbook of Augmented Reality*. https://doi.org/10.1007/978-1-4614-0064-6.

[3]. Itzstein, G. S. Von, Billinghurst, M., Smith, R. T., & Thomas, B. H. (2020). Encyclopedia of Computer Graphics and Games. *Encyclopedia of Computer Graphics and Games*, *January*. https://doi.org/10.1007/978-3-319-08234-9.

[4]. Ekonomou, T., & Vosinakis, S. (2018). Mobile augmented reality games as an engaging tool for cultural heritage dissemination: A case study. *Sci. Cult*, *4*(97-107), 5.

[5]. Yamabe, T., & Nakajima, T. (2013). Playful training with augmented reality games: case studies towards reality-oriented system design. *Multimedia Tools and Applications*, *62*(1), 259-286.

[6]. Rauschnabel, P. A., Rossmann, A., & tom Dieck, M. C. (2017). An adoption framework for mobile augmented reality games: The case of Pokémon Go. *Computers in Human Behavior*, *76*, 276-286.

[7]. Schmitz, B., Specht, M., & Klemke, R. (2012, October). An Analysis of the Educational Potential of Augmented Reality Games for Learning. In *mLearn* (pp. 140-147).

[8]. Klopfer, E., & Sheldon, J. (2010). Augmenting your own reality: Student authoring of science-based augmented reality games. *New directions for youth development*, *2010*(128), 85-94.

[9]. Nilsen, T., Linton, S., & Looser, J. (2004). Motivations for augmented reality gaming. *Proceedings of FUSE*, *4*, 86-93.

[10]. Lv, Z., Halawani, A., Feng, S., Ur Réhman, S., & Li, H. (2015). Touch-less interactive augmented reality game on vision-based wearable device. *Personal and Ubiquitous Computing*, *19*(3), 551-567.

[11]. Liarokapis, F. (2006). An exploration from virtual to augmented reality gaming. *Simulation & Gaming*, *37*(4), 507-533.

[12]. Avery, B., Piekarski, W., Warren, J., & Thomas, B. H. (2006, January). Evaluation of user satisfaction and learnability for outdoor augmented reality gaming. In *AUIC* (Vol. 6, pp. 17-24).

[13]. Tan, C. T., & Soh, D. (2010). Augmented reality games: A review. *Proceedings of Gameon-Arabia, Eurosis*.

[14]. Kim, S. L., Suk, H. J., Kang, J. H., Jung, J. M., Laine, T. H., & Westlin, J. (2014, March). Using Unity 3D to facilitate mobile augmented reality game development. In *2014 IEEE World Forum on Internet of Things (WF-IoT)* (pp. 21-26). IEEE.

[15]. Höllerer, T., Feiner, S., Terauchi, T., Rashid, G., & Hallaway, D. (1999). Exploring MARS: developing indoor and outdoor user interfaces to a mobile augmented reality system. *Computers & Graphics*, *23*(6), 779-785.

[16]. Kojima, M., Sugimoto, M., Nakamura, A., Tomita, M., Nii, H., & Inami, M. (2006, January). Augmented coliseum: An augmented game environment with small vehicles. In *First IEEE International Workshop on Horizontal Interactive Human-Computer Systems (TABLETOP'06)* (pp. 6-pp). IEEE.

[17]. Cheng, K. H., & Tsai, C. C. (2013). Affordances of augmented reality in science learning: Suggestions for future research. *Journal of science education and technology*, *22*(4), 449-462.

[18]. Cheok, A. D., Goh, K. H., Liu, W., Farbiz, F., Fong, S. W., Teo, S. L., ... & Yang, X. (2004). Human Pacman: a mobile, wide-area entertainment system based on physical, social, and ubiquitous computing. *Personal and ubiquitous computing*, *8*(2), 71-81.

[19]. Cheok, A. D., Sreekumar, A., Lei, C., & Thang, L. N. (2006). Capture the flag: mixed-reality social gaming with smart phones. *IEEE Pervasive Computing*, *5*(2), 62-69.

[20]. Wu, M., Mitchell, K., McCaffery, D., Finney, J., & Friday, A. (2004). Real Tournament–mobile context-aware gaming for the next generation. *The Electronic Library*.

[21]. Taigman, Y., Yang, M., Ranzato, M. A., & Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1701-1708).

[22]. Pinheiro, P. O., Lin, T. Y., Collobert, R., & Dollár, P. (2016, October). Learning to refine object segments. In *European conference on computer vision* (pp. 75-91). Springer, Cham.

[23]. Rauschnabel, P. A., Rossmann, A., & tom Dieck, M. C. (2017). An adoption framework for mobile augmented reality games: The case of Pokémon Go. Computers in Human Behavior, 76, 276-286.

[24]. Laine, T. H. (2018). Mobile educational augmented reality games: a systematic literature review and two case studies. Computers, 7(1), 19.

[25]. Shin, D. (2019). How does immersion work in augmented reality games? A user-centric view of immersion and engagement. Information, Communication & Society, 22(9), 1212-1229.

[26]. Bueno, S., Gallego, M. D., & Noyes, J. (2020). Uses and gratifications on augmented reality games: An examination of pokémon go. Applied Sciences, 10(5), 1644.

[27]. Liberati, N. (2018). Phenomenology, pokémon go, and other augmented reality games. Human studies, 41(2), 211-232.

[28]. Parekh, P., Patel, S., Patel, N., & Shah, M. (2020). Systematic review and meta-analysis of augmented reality in medicine, retail, and games. Visual computing for industry, biomedicine, and art, 3(1), 1-20.

[29]. Das, P., Zhu, M. O., McLaughlin, L., Bilgrami, Z., & Milanaik, R. L. (2017). Augmented reality video games: new possibilities and implications for children and adolescents. Multimodal Technologies and Interaction, 1(2), 8.

[30]. Pombo, L., & Marques, M. M. (2020). The potential educational value of mobile augmented reality games: The case of EduPARK app. Education Sciences, 10(10), 287.

[31]. López-Faican, L., & Jaen, J. (2020). EmoFindAR: Evaluation of a mobile multiplayer augmented reality game for primary school children. Computers & Education, 149, 103814.

[32]. Magdy, R., Rashad, S., Hany, S., Tarek, M., Hassan, M. A., & Mohammed, A. (2021, May). Deep Reinforcement Learning Approach for Augmented Reality Games. In 2021 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC) (pp. 330-336). IEEE.

Aaliq Moin

A Computer Science Student from Vignan’s University, having keen interest in developing projects and solutions based on latest technologies with experience in Scripting, Game development and Machine learning.

Lohitha Kolli

A Computer Science Student from Vignan’s University, interested in machine learning and research projects with experience in creating innovative ideas with the help of latest technology.

Venkatramaphanikumar Sistla

Dr. Venkatramaphanikumar S received his Doctoral degree in Computer Science and Engineering from JNTU, Hyderabad, Telangana. He is currently working as Associate Professor in the Department of Computer Science & Engineering. He got more than 11 years of teaching and published more than 40 research articles in reputed National and International Conferences & Journals. His current research interests include Digital image processing, Text Analytics, Pattern Recognition and Medical Imaging.

Venkata Krishna Kishore Kolli

Prof K.V. Krishna Kishore received his Doctoral degree in Computer Science and Engineering from Acharya Nagarjuna University, Guntur, Andhra Pradesh. He is currently working as Professor and Head of Information Technology. He got more than 23 years of teaching and Research experience. His current research interests include Digital image processing, Text Analytics, Pattern Recognition and Medical Imaging.