# Being a Realistic Master: Creating Props & Environments Design for AAA Games

**Harshith Desamsetti[1*], Karu Lal[2]**

[1]Software Engineer, Tata Consultancy Services (TCS), **USA**
[2]Integration Engineer, Ohio National Financial Services, **USA**

**ISSN: 2311-8636 (Print)**
**ISSN: 2312-2021 (Online)**



**Source of Support:** Nil

**No Conflict of Interest:** Declared

***Email for correspondence:**
harshithdesamsetti9@gmail.com

## ABSTRACT

To maximize the potential of computer games beyond entertainment, it is crucial to lower design and development costs and delegate production to domain specialists. While specialists may be interested in incorporating games, they often need more technical expertise to design or customize games for their requirements. To simplify game creation and improve user experience, we suggest defining games as a mixture of simpler games developed through combinatorial creativity. This method avoids the cognitive strain of learning a new design language. Along with speeding up game descriptions, designers might use a set of archetypical games to achieve their goals rather than starting from scratch. The design technique is enhanced by a game platform that provides authoring tools for exporting game designs to XML files and a game engine that automatically generates a 3D virtual world for play. This study explored how this approach allows designers to build games and evaluate their validity through play iteratively. Here, the combinatorial technique was assessed in two focus groups, demonstrating its viability and appeal among technical and non-technical users.

**Keywords:** Game Design, Game Development, Creative Thinking, Problem-Solving, Digital Literacy, Serious Game Simulation

## INTRODUCTION

One of the most challenging components of game production is the creation of models in three dimensions (3D). When we consider that even a game with a medium scope will require thousands of one-of-a-kind assets and that matches with a more extensive size (like Skyrim) will have tens of thousands, many would argue that this is the single most challenging aspect. Even though creating every 3D model will take a different amount of time and effort, this is still an enormous work! We can assist if they are just about to begin working with 3D design for video games and are assessing the amount of labor involved. This article will provide us with advice and direction (from professionals in the industry) on how to create the 3D content of our game, as well as ways to speed up the process, including using bespoke 3D modeling services. The present gaming scene is quite diversified, with dozens of different genres and ways to play.

The terms "compelling characters" and "action-packed scenes" are often what spring to mind when we think of "AAA games." However, the setting plays an equally significant role in the process of developing a really immersive gaming experience. The game comes to life because of a well-designed environment, which provides additional layers of depth, context, and emotional effect to the narrative (Thaduri et al., 2016). The narrative and atmosphere of the game are both impacted to some degree by the environment.

Let us investigate the process of making realistic objects and environments for AAA video games by examining the techniques, strategies, and obstacles involved in creating visually stunning and authentic environments (Desamsetti, 2016a). The team at Kevuru has perfected the art of merging artistry and technology due to their many years of experience in producing props and environments for AAA video games. Their work demonstrates how careful attention to the environment may improve the gameplay as a whole by producing a consistent and believable world that maintains the players' interest (Gutlapalli, 2017a). Working with various 2D and 3D environment design approaches has helped the studio hone its competence.

## CATEGORIES OF 3D MODELS

The broad categories of gaming 3D models have stayed stable throughout the previous decade. Most assets fall within these categories:

- **Characters 3D Models:** Most gaming formats and genres prioritize playable characters. Most modern video games contain playable and non-playable characters (NPCs) who provide essential information and advance the story. Players focus on them. Therefore, their models are the most detailed in the game. Character 3D modeling takes longer than other types of modeling because of this.
- **Environment 3D models:** 3D environments can range from a single house or room to a game-level map, play area, or building complex. It might also be anything in between. Creating miles-wide 3D gaming environments may appear impossible at first. Still, topography is sometimes easily or automatically constructed, and natural object assets (trees, rocks, etc.) are often reused (Alkawaz et al., 2014).
- **Props 3D Models:** A "prop," or non-living game component, can be anything. They can be something a player character interacts with or a setting backdrop. AAA games may prioritize visualizing minor details when modeling 3D props, while most other project types develop props faster than characters or environments. Remember that performances often require several braces.
- **Cars and Vehicles 3D models:** Most cars require complex equipment with hundreds of interconnected parts. The good news is that we do not need to repeat all these parts for a game's 3D car model. Most games depict vehicle exteriors or player seating. The unique geometry of cars and the many factors (some visible from the outside) make vehicle design time-consuming for modelers.
- **Weapons 3D models:** Guns could be props if the game did not emphasize them more than most other items and inventory. Unless we build a new weapon type, players will expect the 3D design to match the game's mechanics and gameplay. They have high weapon aesthetic criteria. Thus, we will likely use 3D model guns for high-polygon games.

## ROLE OF AN ENVIRONMENT ARTIST

We may be interested in learning more about the role of an environment artist, or we are already actively pursuing that role. However, what exactly does an environment artist do when working on producing a video game? Naturally, this varies from studio to studio and from production to production. The project's scale significantly impacts our day-to-day responsibilities and the people we will collaborate with. In this essay, we aim to provide a high-level overview of the duties of an environment artist working on video game development (Gillern & Zina, 2016).

- **Making the World:** The creation of the game world is the primary responsibility of an environment artist. This encompasses everything from planning the room, causing blackouts, and creating the final structural elements and props. This can also include anything else. Put the portion of the screen that uses the most pixels. This is only sometimes the case, however, and it depends on the scale of the project and the studio we work at (Lal, 2015). The expectation is that an environment artist working on a smaller production, such as an independent video game, will own a more significant portion of the pipeline. In a more prominent display, such as a AAA game, the environment artist focuses more on a more specific aspect of the production pipeline and is highly specialized in that aspect, such as set-dressing. We will frequently be required to develop a prop or structural part's design, blockout, and final form while working on independent game development. On an AAA production, we usually already have a plan from concept art, and level designers mostly do the block out. All we need to do is construct the final prop, or we can turn the block out into a simple art block and make a brief for outsourcing so that the outsourcing studio can make the last prop. Set dressing is a significant portion of the work involved in both scenarios (Lal, 2016). This consists of incorporating a wide variety of items into the setting to give the impression that the world is real and to fulfill the requirements of the art director (Suzy et al., 2016, Desamsetti, 2016b)**.**
- **Teamwork:** As environment artists, we are responsible for deciding what will be released into the natural world. Because of this, we will inevitably find ourselves at the core of many different departments. As an illustration, level designers have created a blackout, and the next step is to transform it into a "shippable" level. As an environment artist, it is up to us to ensure this takes place. To make this a reality, we must have conversations with our level designers, art directors, concept artists, prop artists, and animation directors if any cinematic or sequences will be set in that location (Gutlapalli, 2017b). In addition, we need to have a conversation with tech-art to find out the constraints for this environment and how they and we will collaborate to make it run on the hardware with the most minor performance. This is primarily written from the point of view of AAA. When working on an independent production, we will most likely not receive any concept art, and we will be expected to act as our prop artist and level designer. As the environment artist, we are responsible for a much more significant portion of the pipeline.

These are some of the departments that environment artists frequently collaborate with:

- **Level Design:** They are responsible for creating the level blockouts in Level Design.
- **Prop Art**: They not only create the props themselves but also additional content, such as textures, and they are responsible for coordinating the outsourcing of the props.

- **Animation**: In this particular scenario, they are in charge of the spirits or sequences in the environment.
- **Game Design:** The design of the game is something that they are responsible for, including deciding what kinds of weaponry are available, how far the player may jump, and so on.
- **Concept Art:** They come up with the ideas and designs of the environments, referred to as concept art.
- **Art Direction:** To create a visually consistent game, these individuals collaborate closely with concept artists and members of other departments.
- **Tech Art:** is responsible for ensuring that the game runs smoothly on all of the hardware that is intended to be used. In addition, they produce tools and various other exciting items for use by environment artists and other departments.
- **Marketing**: They are the ones that are in charge of promoting the game.

## Environment Design from Scratch

When constructing something from scratch, we start by brainstorming and selecting several photographs from Google or Pinterest to use as visual references. We typically search for images, films, historical references, concept art for shapes, and references for mood. We also seek things that are only sometimes related to the primary topic because doing so helps us generate a unique idea (Mandapuram, 2016).

Following this phase, we sift everything, select the screenshots and references I want to use, compile them into a mood board, and then do some fast sketches based on the board. We will produce a composition sketch or a thorough props sketch, depending on what we must do. When trying to find a solution for an element that already exists at the level but needs to be artistically plausible enough or has some technical limits, we usually sketch something merely to find that answer (Worawan et al., 2017).

At my place of employment, the vast majority of the time, we will bypass all of these steps and immediately begin sketching the environment design right in the engine, with either no references or only half of the concerns being images that the art direction has already chosen (Gutlapalli, 2017c). Sometimes, we draw only to find a solution for an aspect that already exists in the level but needs to be artistically believable (for the universe of For Honor, for example) or has certain technical limits due to the level design. Other times, we draw to practice our drawing skills. Once we have received permission, we will clean the assets with the appropriate metrics and widgets and make any necessary adjustments to the scene based on the comments provided by the art director. After that, we began to define the modeling of the various elements better and make high-resolution versions of them, if necessary, in either ZBrush or 3ds Max (Gutlapalli, 2016a).

We do unwrapping in the meantime (or after all the modeling is finished). The first pass of the materials, texturing, and lighting will be the next thing to be completed. Our only responsibilities at Ubisoft are modeling and level art; other team members are responsible for creating the textures, materials, and lighting. However, if it is a project we are working on for ourselves, our workflow at this stage starts to become more organic, allowing us to go back and make changes to the modeling or UVs based on what we come up with. After we are satisfied with the outcome, we move on to polishing the lighting, and then we polish the materials and textures.

## Importance of Grid

The grid is the primary instrument with which we do all of our work and the first item we decide before beginning a new project. Before beginning tasks such as building environment modules, texture design, and level design, it is vital to determine the units of measurement used in the project (Desamsetti & Mandapuram, 2017). In addition, we need to consider the environment's dimensions that will be believable from the player's perspective and the gameplay-specific measures, such as the smallest distance that will still permit two players to fight freely against one another. After these standards have been determined (often by the gameplay team), all of the game's artists, animators, and level designers must adhere to them (Deming et al., 2018). Therefore, the grid is the most significant tool for the environment artists to use when constructing their modular assets, and it is present in both the 3D application and the engine. The first step in configuring the grid is ensuring its dimensions are compatible with the one within the game engine. For instance, the units used in Unreal 4 are centimeters; hence, it is advised that the grid and the teams in the 3D program be set up to be the same to prevent scaling issues and ensure that the assets may be exported without any interruptions. We need to use the snap tool when working with the grid to construct in 3ds Max with the correct dimensions and the pivot (gizmo) in the appropriate locations. The position of the pivot point on the modules is critical because once installed in the engine; the snap will enable us to place them quickly and on the grid in the same manner as Lego blocks. When exporting, the grid is also critical since the asset's pivot needs to be aligned with the grid's origin (0, 0, 0) to prevent the support from having an offset pivot in the engine. To return to the engine, the snap tool will make it easy to construct and iterate environments in a relatively short time using the grid (Gareth et al., 2015).

## Keeping the Balance

When we build the first pass of the essential modular components, we test the gameplay to ensure it is smooth and get some feedback from the level designers on improving it. For instance, the doorway leading into a room or region in a third-person fighting game must be at least two meters wide so players feel they are not being hemmed in. In addition, the open space where the battle will take place must be at least three or four meters in size. When working in the enclosed areas, we must be cautious about where we put the props and the minute details (Mandapuram, 2017a). The rationale for this is that we want to make sure the player is clear if they are required to respond rapidly. It is also applicable to textures; the people who work on our surfaces try to avoid including an excessive number of colors or noisy features, and as a modeler, we have to be careful when we choose textures and when we do the unwrapping. Here is where we put the game through its paces to ensure it is playable and does not cause any confusion!

## Asset Approach: Rock Sculpting

Realistic asset creation is complex. In several AAA games, the team uses photogrammetry or procedural technologies with displacement and tessellation for rocks. In For Honor, procedural rocks were not the most excellent solution because much memory was already used for gameplay elements like AI (Navarrete, 2015).

We used ZBrush sculpting and scanning because we were a small yet aggressive team used to generate new content regularly. The cliffs in Jordan and Monument Valley, US, inspired us; therefore, we searched for pictures:

After the rocks were approved, we chose a bank-like 3D scan model. Low-resolution scans are easy to edit. In 3ds Max, we cleaned the scan and roughly reshaped it. A texture is sometimes mapped on the reference picture. The goal is to generate a topology with forms and basic fracture information, not the most excellent scan ever.

The high-resolution scan sculpts the most significant task. We copy the scan, close its holes, and delete its exploded triangles in ZBrush. We then transformed it into a high-rez dynasty to preserve the definition.

After a quick UV pass, we project the scan's textures and details. Since it will be used as a reference for sculpting, minute details are unnecessary. We sculpted the rock's primary fissures and details before adding the details. We can improvise with the remaining crack details without more references. Again, the goal is to make rocks seem reasonable from every angle and lighting.

After sculpting, we export the mesh to 3ds Max, conduct a low-resolution PolyDraw retopo, UV with peel mode, and bake normal maps. We recommend Marmoset for baking, but we can use any software.

We can use the TrimSmoothBorderBrush with alpha from the online ZBrush library, a simple version, and a Clay Build-Up in spray mode as our main brush.

## Utilizing Modularity

While working on For Honor, we had the opportunity to create modular assets such as pillar walls and basement arches, in addition to several prefabricated kits of interchangeable fortress wall components. Before sending any of them out to be polished, the only thing I had to do for some of them was make sure that they were working (Zarraonandia et al., 2017). In the workflow for the modular pieces, the first step would be adapting everything to the defined grid, as we described previously in this paragraph. This would be done to ensure that everything fits appropriately. The measurements of the modular piece must be accurate, considering the pixel ratio and the texturing (Thodupunori & Gutlapalli, 2018). In most cases, where the texture dimensions are multiples of 2, we will construct modular assets with measurements that are multiples of 2. For instance, the pixel ratio in a third-person game might be 1 meter equal to a texture of 512, 2 meters equal to 1024, etc., and the length of the modular components might be 2 meters, 4 meters, or 8 meters. Remembering that there is no hard and fast rule about the pixel ratio a game should have is essential. It is different every time we play the game.

## Create 3D Models For Games.

If we follow the usual approach, an experienced modeler should be able to enter a project quickly and start generating 3D material with a minimal amount of delay. The opposite should be true if we do not follow the standard procedure (Mandapuram, 2017b). This is because the following steps are what constitute the typical procedure:

**Collecting references:** The vast majority of the time, 3D models are based on previous assets or things found in the real world; as a result, it is beneficial to acquire the relevant references and keep them close at hand. For instance, we can construct 3D material by starting with a reference image, photo, sketch, video, gaming component, or even actual people and real places. Even in circumstances in which the modeler's imagination ought to serve as the only source of inspiration for the design of the object, it is helpful to jot down or sketch the general appearance of the thing to maintain coherence as the process develops. This is because the modeler's imagination should serve as the solitary source of inspiration for the design of the object. (Chen, 2013).

**Choosing a modeling method and planning:** As soon as we have built the fundamental concept for the model, the following phase is to make decisions regarding some of the model's characteristics, such as the amount of detail it will have (Dekkati & Thaduri, 2017). Wee that adding more information leads to an increase in the poly count and the amount of necessary modeling time, not to mention the rise in the pressure placed on the players' hardware. We can always add more characteristics to the model by texturing it, but some 3D objects work better when rendered in a low-poly format. In addition, there is a vast range of different ways to 3D modeling for games, and we can select one of these approaches based on the model's intended application. We can do this because there is a wide variety of different approaches to 3D modeling for games.

**Modeling:** When we begin the process of modeling in its most exact form in a 3D design application (such as 3DS Max, Maya, or ZBrush), we are constructing our polygon mesh. This is the first step in the modeling process. It is occasionally required to begin with a basic three-dimensional box shape and then modify it by slicing, pulling, and adding polygons. This can be done when necessary. Scanners that capture data in three dimensions (3D), photogrammetry, computer-aided design (CAD), and algorithms are all methods that, depending on the specifics of the situation, can be utilized to move the process along more quickly. These inside-and-outside-the-box thinking approaches are prevalent when it comes to the design of structures and environments.

**Render:** This step is entirely discretionary and is done to exhibit the completed model. After photographing the model in its natural environment, a still image is created from the model (Lal & Ballamudi, 2017). This image can then be utilized for marketing purposes or to demonstrate the game's visual quality. Even after a render has been produced, the 3D model can still be employed for various additional applications and incorporated into software.

## INTEGRATE A 3D MODEL INTO A GAME

It is crucial to understand that 3D models are not finished goods; they lack components such as color, texture, lighting, and other features required to make them playable in a video game. Consequently, texturing is the next stage after modeling, although it is sometimes done simultaneously. We may give the object color and other visual aspects by creating a 2D map (texture), which is then pulled over the 3D model to make its features look as though they are complete. This process allows us to give the object color and other visual elements. This takes place once the form of the thing has been completed in its entirety (Lal et al., 2018). Shading and lighting typically achieve a more profound sense of realism. If the 3D object is meant to do something other than sit at a particular spot in the game, the professionals will also create animation (Desamsetti, 2018). How characters and other living beings navigate space is determined by the rigging applied to them. Because the developers have integrated the assets into the software and written code for these behaviors, certain more intricate things have also been scripted to behave in a particular way depending on the player's actions (Buyuksalih et al., 2017).

## Modeling for Different Platforms

This is a common concern voiced by video game developers whenever they begin collaborating on a project using a different platform.

**Mobile:** Even though most modern mobile devices have a good amount of processing power, games still need many optimizations to avoid causing the hardware to become overheated or run more slowly. As a direct result of this, the poly count in mobile games tends to be lower, the quantity of assets and content is typically lower, and all of this contributes to a design stage that is typically more quickly completed (Mandapuram et al., 2018). The requirements for 3D modeling in games for Android and iOS are less stringent than those for some of the other possibilities.

**Console/PC:** Games for consoles and personal computers have the advantage of running on powerful hardware, meaning developers have greater leeway and opportunities to produce high-detail models without significantly impacting the games' playability. However, because many games, such as open-world games, make far too much content immediately accessible to players, developers can program games to render 3D models in high resolution only when the player is relatively close to them (Mat et al., 2014).

**Web:** Web games typically do not have the luxury of being playable offline, so optimization is paramount. Even before gameplay begins, it may take a significant amount of time for a user to load a game map and its associated content (Mandapuram & Hosen, 2018). Because of this, it is more frequent to see low-poly models in online games to make gameplay smoother or to display a loading screen while the user's client caches most of the data.

**VR/AR:** In terms of performance, virtual reality games are comparable to those found on personal computers and game consoles, but augmented reality games are most appropriately categorized as mobile titles (Ballamudi & Desamsetti, 2017). It is essential to remember that players of virtual reality games have an expansive field of view and generally observe the game environment from a first-person perspective; as a result, even the tiniest of details are incredibly noticeable. Because of this, the most successful virtual reality games are packed with high-poly graphics that the designers have put much effort into crafting (Gutlapalli, 2016b).

## Conclusion

A new method for creating realistic human facial expressions combines sweat and tears with skin tone. The problem is excessive appearances like fear and sweating when happy, crying, and blushing. This method creates realistic animation expressions using sweat and tears. The pulse oximeter and 3D skin analyzer measure how oxygenation affects skin color. Animation and realism of fluid sweat and tears using the particle system and SPH approach are highlighted. Physical and physiological factors affect facial skin appearance. Extreme sweating, tears, and blushing are generated for 3D games and face movement simulations. We use the same techniques to produce detailed and realistic facial animation expressions. The pragmatic facial emotion animation on virtual humans accounts for facial skin coloring fluctuations. A better facial animation model with color, sweat, and tears classification is shown. A link exists between realistic facial color creation and hemoglobin's oxygenation and deoxygenation effects. To our knowledge, no animation model had combined sweat, tears, and color to create accurate 3D avatar facial expressions before this model. Use FACS on individual faces to express and choose any model or animation. Divining face activities into action units has made facial expressions more pleasing. This crumbles and changes shape over time. The suggested framework uses facial energy, particle system, and SPH approach to simulate tears, sweat, and blushing. Our unique combination of face animation framework settings may allow animators to regulate sweating and tears using FACS-compliant parameters.

## References


Alkawaz, M. H., Basori, A. H., Mohamad, D., Mohamed, F. (2014). Realistic Facial Expression of Virtual Human Based on Color, Sweat, and Tears Effects. *The Scientific World Journal*, *2014*, 1-9. https://doi.org/10.1155/2014/367013

Ballamudi, V. K. R., & Desamsetti, H. (2017). Security and Privacy in Cloud Computing: Challenges and Opportunities. *American Journal of Trade and Policy*, *4*(3), 129–136. https://doi.org/10.18034/ajtp.v4i3.667

Buyuksalih, I., Bayburt, S., Buyuksalih, G., Baskaraca, A. P., Karim, H., and Rahman, A. A. (2017). 3d Modelling and Visualization Based on the Unity Game Engine – Advantages and Challenges. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *IV-4/W4*, 161-166. https://doi.org/10.5194/isprs-annals-IV-4-W4-161-2017

Chen, Y. (2013). Studies on Web Game Interface Design. *Applied Mechanics and Materials*, *401-403*, 1756-1759. https://doi.org/10.4028/www.scientific.net/AMM.401-403.1756

Dekkati, S., & Thaduri, U. R. (2017). Innovative Method for the Prediction of Software Defects Based on Class Imbalance Datasets. *Technology & Management Review*, *2*, 1–5. https://upright.pub/index.php/tmr/article/view/78

Deming, C., Dekkati, S., & Desamsetti, H. (2018). Exploratory Data Analysis and Visualization for Business Analytics. *Asian Journal of Applied Science and Engineering*, *7*(1), 93–100. https://doi.org/10.18034/ajase.v7i1.53

Desamsetti, H. (2016a). A Fused Homomorphic Encryption Technique to Increase Secure Data Storage in Cloud Based Systems. *The International Journal of Science & Technoledge*, *4*(10), 151-155.

Desamsetti, H. (2016b). Issues with the Cloud Computing Technology. *International Research Journal of Engineering and Technology (IRJET)*, *3*(5), 321-323.

Desamsetti, H. (2018). Internet of Things (IoT) Technology for Use as Part of the Development of Smart Home Systems. *International Journal of Reciprocal Symmetry and Theoretical Physics*, *5*(1), 14-21.

Desamsetti, H., & Mandapuram, M. (2017). A Review of Meta-Model Designed for the Model-Based Testing Technique. *Engineering International*, *5*(2), 107–110. https://doi.org/10.18034/ei.v5i2.661

Gareth, E., Li, H., & Bin, W. (2015). BIM-based collaborative and interactive design process using computer game engine for general end-users. *Visualization in Engineering*, *3*(1), 117. https://doi.org/10.1186/s40327-015-0018-2

Gillern, S. V., Zina, A. (2016). Games and Game-based Learning in Instructional Design. *The International Journal of Technologies in Learning*, *23*(4), 1-7. https://doi.org/10.18848/2327-0144/CGP/v23i04/1-7

Gutlapalli, S. S. (2016a). An Examination of Nanotechnology's Role as an Integral Part of Electronics. *ABC Research Alert*, *4*(3), 21–27. https://doi.org/10.18034/ra.v4i3.651

Gutlapalli, S. S. (2016b). Commercial Applications of Blockchain and Distributed Ledger Technology. *Engineering International*, *4*(2), 89–94. https://doi.org/10.18034/ei.v4i2.653

Gutlapalli, S. S. (2017a). Analysis of Multimodal Data Using Deep Learning and Machine Learning. *Asian Journal of Humanity, Art and Literature*, *4*(2), 171–176. https://doi.org/10.18034/ajhal.v4i2.658

Gutlapalli, S. S. (2017b). The Role of Deep Learning in the Fourth Industrial Revolution: A Digital Transformation Approach. *Asian Accounting and Auditing Advancement*, *8*(1), 52–56. Retrieved from https://4ajournal.com/article/view/77

Gutlapalli, S. S. (2017c). An Early Cautionary Scan of the Security Risks of the Internet of Things. *Asian Journal of Applied Science and Engineering*, *6*, 163–168. Retrieved from https://ajase.net/article/view/14

Lal, K. (2015). How Does Cloud Infrastructure Work?. *Asia Pacific Journal of Energy and Environment*, *2*(2), 61-64. https://doi.org/10.18034/apjee.v2i2.697

Lal, K. (2016). Impact of Multi-Cloud Infrastructure on Business Organizations to Use Cloud Platforms to Fulfill Their Cloud Needs. *American Journal of Trade and Policy*, *3*(3), 121–126. https://doi.org/10.18034/ajtp.v3i3.663

Lal, K., & Ballamudi, V. K. R. (2017). Unlock Data's Full Potential with Segment: A Cloud Data Integration Approach. *Technology & Management Review*, *2*(1), 6–12. https://upright.pub/index.php/tmr/article/view/80

Lal, K., Ballamudi, V. K. R., & Thaduri, U. R. (2018). Exploiting the Potential of Artificial Intelligence in Decision Support Systems. *ABC Journal of Advanced Research*, *7*(2), 131-138. https://doi.org/10.18034/abcjar.v7i2.695

Mandapuram, M. (2016). Applications of Blockchain and Distributed Ledger Technology (DLT) in Commercial Settings. *Asian Accounting and Auditing Advancement*, *7*(1), 50–57. https://4ajournal.com/article/view/76

Mandapuram, M. (2017a). Application of Artificial Intelligence in Contemporary Business: An Analysis for Content Management System Optimization. *Asian Business Review*, *7*(3), 117–122. https://doi.org/10.18034/abr.v7i3.650

Mandapuram, M. (2017b). Security Risk Analysis of the Internet of Things: An Early Cautionary Scan. *ABC Research Alert*, *5*(3), 49–55. https://doi.org/10.18034/ra.v5i3.650

Mandapuram, M., & Hosen, M. F. (2018). The Object-Oriented Database Management System versus the Relational Database Management System: A Comparison. *Global Disclosure of Economics and Business*, *7*(2), 89–96. https://doi.org/10.18034/gdeb.v7i2.657

Mandapuram, M., Gutlapalli, S. S., Bodepudi, A., & Reddy, M. (2018). Investigating the Prospects of Generative Artificial Intelligence. *Asian Journal of Humanity, Art and Literature*, *5*(2), 167–174. https://doi.org/10.18034/ajhal.v5i2.659

Mat, R. C., Shariff, A. R. M., Zulkifli, A. N., Rahim, M. S. M., Mahayudin, M. H. (2014). Using game engine for 3D terrain visualization of GIS data: A review. *IOP Conference Series, Earth and Environmental Science*, *20*(1), 1-11. https://doi.org/10.1088/1755-1315/20/1/012037

Navarrete, C. C. (2015). Creative Thinking in K-12 Education: Game Design and Development Curricula for Digital Literacies and 21st Century Learning. *International Journal of Computer Research*, *22*(1), 1-24.

Suzy, J., Haytham, S., Diane, S. (2016). It's All in the Game: A 3D Learning Model for Business Ethics. *Journal of Business Ethics: JBE*, *137*(2), 383-403. https://doi.org/10.1007/s10551-015-2557-9

Thaduri, U. R., Ballamudi, V. K. R., Dekkati, S., & Mandapuram, M. (2016). Making the Cloud Adoption Decisions: Gaining Advantages from Taking an Integrated Approach. *International Journal of Reciprocal Symmetry and Theoretical Physics*, *3*, 11–16. https://upright.pub/index.php/ijrstp/article/view/77

Thodupunori, S. R., & Gutlapalli, S. S. (2018). Overview of LeOra Software: A Statistical Tool for Decision Makers. *Technology & Management Review, 3*(1), 7–11.

Worawan, N., Ali, M., Tomohiro, F., Nobuyoshi, Y. (2017). Integrating building information modeling and virtual reality development engines for indoor lighting design. *Visualization in Engineering*, *5*(1), 1-21. https://doi.org/10.1186/s40327-017-0058-x

Zarraonandia, T., Diaz, P., Aedo, I. (2017). Using combinatorial creativity to support the end-user design of digital games. *Multimedia Tools and Applications*, *76*(6), 9073-9098. https://doi.org/10.1007/s11042-016-3457-4

--0--