2020-12

# A SYSTEMS ENGINEERING APPROACH TO COMPARING MIXED REALITY GAMING ENGINES WITHIN THE DOD

Cha, Ted L.; Davis, Blake A.; Shutte, Zachariah R.; Snodgrass, Douglas J.; Wimsatt, Christopher J.; Ybarra, Rene V.

Monterey, CA; Naval Postgraduate School

https://hdl.handle.net/10945/66606

# NAVAL POSTGRADUATE SCHOOL

**MONTEREY, CALIFORNIA**

# SYSTEMS ENGINEERING CAPSTONE REPORT

**A SYSTEMS ENGINEERING APPROACH TO COMPARING MIXED REALITY GAMING ENGINES WITHIN THE DOD**

by

Ted L. Cha, Blake A. Davis, Zachariah R. Shutte,
Douglas J. Snodgrass, Christopher J. Wimsatt, and
Rene V. Ybarra

December 2020

Advisor: Alejandro S. Hernandez
Co-Advisor: Joseph W. Sweeney
Co-Advisor: Douglas L. Van Bossuyt

THIS PAGE INTENTIONALLY LEFT BLANK

| REPORT DOCUMENTATION PAGE | | | *Form Approved OMB No. 0704-0188* |
|---|---|---|---|
| Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC, 20503. | | | |
| **1. AGENCY USE ONLY** ***(Leave blank)*** | **2. REPORT DATE** December 2020 | **3. REPORT TYPE AND DATES COVERED** Systems Engineering Capstone Report | |
| **4. TITLE AND SUBTITLE** A SYSTEMS ENGINEERING APPROACH TO COMPARING MIXED REALITY GAMING ENGINES WITHIN THE DOD | | | **5. FUNDING NUMBERS** |
| **6. AUTHOR(S)** Ted L. Cha, Blake A. Davis, Zachariah R. Shutte, Douglas J. Snodgrass, Christopher J. Wimsatt, and Rene V. Ybarra | | | |
| **7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)** Naval Postgraduate School Monterey, CA 93943-5000 | | | **8. PERFORMING ORGANIZATION REPORT NUMBER** |
| **9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(E**S) Joint Special Operations Command, Fort Bragg, NC 28308 | | | **10. SPONSORING / MONITORING AGENCY REPORT NUMBER** |
| **11. SUPPLEMENTARY NOTES** The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. | | | |
| **12a. DISTRIBUTION / AVAILABILITY STATEMENT** Approved for public release. Distribution is unlimited. | | | **12b. DISTRIBUTION CODE** A |

**13. ABSTRACT (maximum 200 words)**

Joint Special Operations Command (JSOC), the primary stakeholder of this report, identified a need to visualize the operating environment prior to mission execution. Historically, JSOC performed visualization by two-dimensional (2D) means that lacked real-time capabilities such as imagery or sand tables. Technological advances now enable visualization of the operating environment by three-dimensional (3D) means and in real-time environments by using mixed reality (MR). An essential component of an MR system is a gaming engine, which serves as essential software in creating the MR environment. Currently, dozens of proprietary and open-source gaming engines are available for use by system designers. This research applies the systems engineering "V" model, using mixed methods to explore the different comparison criteria of gaming engines. The research team developed a structured approach to assessing different MR gaming engine alternatives. Using Multiobjective Decision Analysis and Additive Value Modeling as a basis, the research team produced a credible, repeatable, traceable selection tool to compare alternatives. The team also discovered that the gaming engine is a critical component of the MR system, but it should not be the sole basis for MR system comparison. Other considerations for selecting an MR system are system developers, the hardware and software security considerations, and interoperability within the DOD system architecture.

| **14. SUBJECT TERMS** selection methodology, mixed reality, analysis of alternatives, gaming engines, augmented reality, virtual reality | | | **15. NUMBER OF PAGES** 113 |
|---|---|---|---|
| | | | **16. PRICE CODE** |
| **17. SECURITY CLASSIFICATION OF REPORT** Unclassified | **18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified | **19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified | **20. LIMITATION OF ABSTRACT** UU |

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

**Approved for public release. Distribution is unlimited.**

# A SYSTEMS ENGINEERING APPROACH TO COMPARING MIXED REALITY GAMING ENGINES WITHIN THE DOD

MAJ Ted L. Cha (USA), CPT Blake A. Davis (USA), CPT Zachariah R. Shutte (USA),
CPT Douglas J. Snodgrass (USA), MAJ Christopher J. Wimsatt (USA),
and MAJ Rene V. Ybarra (USA)

Submitted in partial fulfillment of the
requirements for the degree of

**MASTER OF SCIENCE IN SYSTEMS ENGINEERING MANAGEMENT**

from the

**NAVAL POSTGRADUATE SCHOOL**
**December 2020**

Lead Editor: Rene V. Ybarra

Reviewed by:
Alejandro S. Hernandez
Advisor

Joseph W. Sweeney
Co-Advisor

Douglas L. Van Bossuyt
Co-Advisor

Accepted by:
Ronald E. Giachetti
Chair, Department of Systems Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

# ABSTRACT

Joint Special Operations Command (JSOC), the primary stakeholder of this report, identified a need to visualize the operating environment prior to mission execution. Historically, JSOC performed visualization by two-dimensional (2D) means that lacked real-time capabilities such as imagery or sand tables. Technological advances now enable visualization of the operating environment by three-dimensional (3D) means and in real-time environments by using mixed reality (MR). An essential component of an MR system is a gaming engine, which serves as essential software in creating the MR environment. Currently, dozens of proprietary and open-source gaming engines are available for use by system designers. This research applies the systems engineering “V” model, using mixed methods to explore the different comparison criteria of gaming engines. The research team developed a structured approach to assessing different MR gaming engine alternatives. Using Multiobjective Decision Analysis and Additive Value Modeling as a basis, the research team produced a credible, repeatable, traceable selection tool to compare alternatives. The team also discovered that the gaming engine is a critical component of the MR system, but it should not be the sole basis for MR system comparison. Other considerations for selecting an MR system are system developers, the hardware and software security considerations, and interoperability within the DOD system architecture.

THIS PAGE INTENTIONALLY LEFT BLANK

## TABLE OF CONTENTS

# LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

# LIST OF TABLES

# LIST OF ACRONYMS AND ABBREVIATIONS

| | |
|---|---|
| 2D | two-dimensional |
| 3D | three-dimensional |
| AoA | Analysis of Alternatives |
| AR | Augmented Reality |
| ATAK | Android Team Awareness Kit |
| A/V | Audio and Video |
| AV | Augmented Virtuality |
| BDD | Block Definition Diagram |
| CAMEO | Cameo Enterprise Architecture |
| CIO | Chief Information Officer |
| COA | Course of Action |
| Coms | Communications |
| COP | Common Operating Picture |
| COTS | Commercial-Off-The-Shelf |
| CPOF | Command Post of the Future |
| CPU | Central Processing Unit |
| DOD | Department of Defense |
| DoDI | Department of Defense Instruction |
| DoT | Department of Transportation |
| FMV | Full Motion Video |
| FPS | Frames Per Second |
| FRAGO | Fragmentary Order |
| GPU | Graphics Processing Unit |
| GUI | Graphical User Interface |
| HUMINT | Human Intelligence |
| IDE | Integrated Development Environment |
| INCOSE | International Council on Systems Engineering |
| I/O | Input/Output |

| | |
|---|---|
| IPB | Intelligence Preparation of the Battlefield |
| IPR | In Progress Review |
| ISR | Intelligence, Surveillance, And Reconnaissance |
| JCR | Joint Capabilities Release |
| JEDI | Joint Enterprise Defense Infrastructure |
| JSOC | Joint Special Operations Command |
| K | One-Thousandth (Numerical Value) |
| max | maximum |
| MBSE | Model-Based Systems Engineering |
| MDMP | Military Decision-Making Process |
| MED | Medium |
| min | minimum |
| MOE | Measures Of Effectiveness |
| MOP | Measure Of Performance |
| MR | mixed reality |
| ms | millisecond |
| NDIA | National Defense Industrial Association |
| NPC | Non-Playable Character |
| NPS | Naval Postgraduate School |
| OPORD | Operations Order |
| OPSEC | Operational Security |
| OS | Operating System |
| SEV | Systems Engineering "V" |
| ROI | Return of Investment |
| SIGINT | Signal Intelligence |
| SMU | Special Mission Unit |
| SysML | Systems Modeling Language |
| SoS | Systems of Systems |
| TLP | Troop Leading Procedures |
| UAS | Unmanned Aircraft System |

| | |
|---|---|
| UI | User Interface |
| VR | Virtual Reality |
| Y/N | Yes or No |

THIS PAGE INTENTIONALLY LEFT BLANK

# EXECUTIVE SUMMARY

General George S. Patton said, “a good plan violently executed now is better than a perfect plan executed next week.” The urgency in planning a sophisticated and precise direct-action type of operation for Special Operations Warfighters have increased in demand over the past decade, a key example being the Osama Bin Laden Raid. JSOC requires a mixed reality (MR) mission planning tool to maintain competitive advantage and enhance mission planning. Implementation of new technology is never without challenges. One challenge faced by JSOC is the selection of a gaming engine, a critical software system within the mission planning tool. The premise of this research is to develop a selection methodology to assess available market gaming engines to support JSOC’s selection of a MR mission planning tool.

The research team executed a systems engineering approach to identify the root problem, define gaming engine requirements, and ultimately produce a selection methodology. The primary result of this research confirmed that a systems engineering approach and value modeling can generate a repeatable, traceable selection methodology that supports current and future acquisition of an MR mission planning tool. While the tool is unique to a MR gaming engine, the proven approach has broader application to selecting other technical components of larger DOD and MR systems. The defined process offered by the systems engineering domain, coupled with this research’s application to a system component, offers a road map for other technical applications.

## Conclusions

Through the course of this project, the research team identified several findings of interest. The research team found: that gaming engine is not the limiting factor; a quality tool requires quality analysis; development team, security, and interoperability considerations provide opportunity for further research.

**The gaming engine is not the limiting factor** – The top performing state-of-the-market gaming engines satisfy JSOC’s needs for use as a MR mission planning tool. The measurable difference between top gaming engines is minimal. Since the gaming engine is not the limiting factor for MR capabilities, JSOC should expand their analysis beyond gaming engines to a comparison of MR systems. This analysis should consider the gaming engine as an important component of the MR system, along with the hardware, networking capability, security features, and interoperability with existing systems.

**A quality tool requires quality analysis –** The hard part is allotting the necessary time, people, and resources toward defining the problem and producing quality requirements that satisfy the operational need and drive the value modeling process. It is imperative that organizations allocate time and resources to perform upfront analysis for the value modeling process to provide benefit to the program.

**Development team, security, and interoperability considerations provide opportunity for further research –** While it is an important component of the MR system, the gaming engine should not be the sole basis for MR system comparison. Other considerations for selecting a gaming engine that should be considered include the system developers, the hardware and software security considerations, and interoperability within the DOD system architecture.

## Supporting Evidence

This research project generally followed the System Engineering 'V' process (see Figure 6) dividing the research into three phases: Problem Understanding, Requirements Definition, and Value Analysis.

### *Problem Understanding*

The research team conducted literature reviews, stakeholder analysis, and operational analysis to develop an accurate understanding of the system and the problem JSOC faced. It was important to fully understand the problem so as not to attempt to correct a symptom of the problem. The literature review enabled the research team to gain a collective understanding of existing literature and concepts on gaming engines and MR technologies. Stakeholder analysis identified unique and competing needs between the different entities involved in the system's development. Through operational analysis, the research team captured high level expectations of the system to aid in understanding the problem. Key artifacts of produced during this phase were the stakeholder analysis table and multiple variations of operation concept descriptions. The research and application of the systems engineering process allowed the research team to adequately and thoroughly defined the problem and understand how the system would be employed in its operational environment.

*Requirements Definition*

Based on the operational analysis and prevailing literature on gaming engines, the research team conducted functional decomposition of a gaming engine's core functions. The team identified four top level functional that focused on creating a virtual environment, interoperability, networking, and security. These top-level functions guided the research team toward understanding what a gaming engine should do. A stakeholder needs traceability matrix was used to validate that all functional requirements satisfied stakeholder needs. The functional analysis helped the research team define the basic requirements of the MR gaming engine.

*Value Analysis*

The research team developed a value hierarchy and applied additive value modeling to identify comparison criteria and develop a selection methodology for MR gaming engines. The research team applied its understanding of state-of-the-market MR gaming engines and stakeholder needs to develop a value hierarchy that reflects the MOEs and MOPs of an MR gaming engine for JSOC. Then, the research team put each MOP into two categories: binary and nonbinary. The MOPs categorized as binary are considered screening criteria for gaming engines, meaning the considered gaming engine alternatives must have those features. For the remaining nonbinary MOPs, the research team developed value hierarchies to translate raw performance data into a common value scale of 1–10. The team provided the stakeholder with a Stakeholder Feedback Rubric (Table 17) used to assess the level of importance of each value measure. The research team then applied the Parnell Method (Parnell and Trainor 2009) to elicit swing weights using the stakeholder's levels of importance and variation observed between the different gaming engine alternatives.

The research team used this selection methodology to assess the two leading gaming engines in the industry, Unity and Unreal, against the additive model described above. Based on the data received and using this additive model as a selection methodology, Unity outperforms Unreal by a slim margin. This should not be taken as conclusive evidence that Unity is better than Unreal, however, because much of the data provided is influenced by variables outside of the gaming engine itself. What this selection methodology does provide for the stakeholder is a repeatable, traceable framework that can be used in the acquisition of future MR systems and technologies.

# ACKNOWLEDGMENT

This research was not possible without the humble support of family, friends, scholars, and leaders. First and foremost, the families of the capstone team, who endured a lifetime of deployments, missed holidays, birthdays and anniversaries, numerous military moves, COVID quarantine, hours of distance learning, and many sacrifices that without the love and support of our families, our research would not have been possible.

Secondly, an overwhelming thank you to our primary stakeholder, Laura Cowdrey, whose patience, insight, and invaluable input allowed the team to produce a product that we hope to benefit present and future warfighters of JSOC.

Gratitude and the upmost appreciation to our advisors, Dr. Douglas Van Bossuyt, Dr. Alejandro Hernandez, Professor Joseph Sweeney, and Professor John Dillard (retired). Your systems engineering wisdom, mentorship, and academic guidance allowed for us all to grow stronger as scholars and leaders. Our countless meetings and azimuth checks gave us the needed motivation and challenged our critical thinking, which resulted in the gift of knowledge. A special thank you to the faculty of the Systems Engineering Department at Naval Postgraduate School. This project wouldn't have had the disciplined approach needed to leave the ground. Also, an enormous thank you to Ms. Cheryl Huddleston of the Dudley Knox Graduate Writing Center. The time you invested in helping us edit and giving us an objective perspective enhanced our research project, and most of all made us all better writers.

Albert Einstein said it best, "Every day I remind myself that my inner and outer life are based on the labors of other men, living and dead, and that I must exert myself in order to give in the same measure as I have received and am still receiving." Lastly, a special appreciation to all ours peers, mentors, and leaders past and present who have groomed us to continue on their legacy, and the legacy that we hope to share with others.

THIS PAGE INTENTIONALLY LEFT BLANK

# I. INTRODUCTION

To improve effectiveness, the Armed Forces have increasingly incorporated mixed reality (MR) in training, planning, and operational contexts. Mixed reality describes a spectrum of artificial realities, including augmented reality, augmented virtuality, and virtual reality (De Guzman et al. 2018, 2). The mixed reality spectrum ranges from the real environment to the virtual environment (Deguzman et al. 2018, 2). De Guzman suggests three overlapping distinctions within the spectrum: Augmented Reality (AR), Augmented Virtuality (AV), and Virtual Reality (VR). Figure 1 The Spectrum of Mixed Reality visually expresses the MR spectrum as a sum of its parts. While researchers have provided varying definitions of AR, VR, and MR, for the purposes of this research project, the term MR is used to describe both AR and VR together along the Spectrum of Mixed Reality.

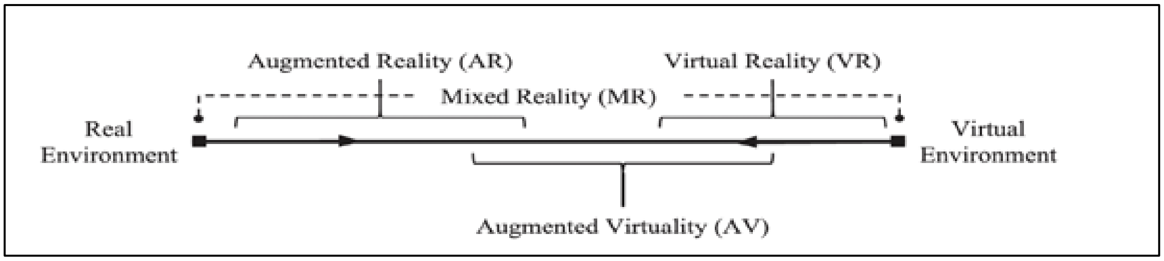



Figure 1. The Spectrum of Mixed Reality. Source: Deguzman et al (2018).

The military views the incorporation of mixed reality into their planning process as a technology that improves the fidelity and accuracy of the process itself, enables decentralized rehearsals among geographically dispersed units, and decreases the time required to execute a mission. The primary stakeholder, Joint Special Operations Command (JSOC), is a unit within the Department of Defense (DOD) exploring MR as a viable medium to increase capabilities to the warfighter. JSOC has identified a need to visualize the operating environment prior to mission execution.

We recommend JSOC consider this project's identification of unique comparison criteria when selecting a MR gaming engine. The comparison criteria may be utilized to create a tool that aids JSOC's performance of an Analysis of Alternatives (AoA). The tool will identify key comparison criteria, accept weighted stakeholder preferences, empirically score each gaming engine under consideration, and return a gaming engine. The tool is applicable to the current JSOC acquisition as well as future DOD acquisitions of systems incorporating gaming engines. Additionally, the comparison criteria this research identifies will inform MR requirements development as the DOD moves forward with further acquisitions of advanced technology.

Historically, JSOC performed visualization by two-dimensional (2D) means that lacked real-time capabilities such as imagery or sand tables. The term "sand table" is an adage used to replicate an operation to scale. Leaders use the scaled replicas to manipulate physical icons on a 2D plane to simulate military operations and gain understanding of the battlefield. There are drawbacks, however, to traditional sand tables. The 2D plane limits the number of participants who can participate. Also, the model is difficult to alter to show real-time changes on the battlefield. Lastly, the replicated terrain is seldom accurate or to scale.

The modern technological advances now enable visualization of the operating environment by three-dimensional (3D) means and in real-time environments. 3D mapping software enables the transformation of full motion video (FMV) from Intelligence, Surveillance, and Reconnaissance (ISR) platforms into data packages that can be displayed using MR gaming engines.

Emphasizing how MR technologies improve the fidelity and accuracy of the planning process the research team referenced a case study that the sponsor related to the project team, describing a small mission unit that was invited to conduct a hasty planning exercise using traditional, two-dimensional, birds-eye-view imagery. The planning team identified three entry points to the target. However, the lack of fidelity, or a three-dimensional view, resulted in dangerous planning assumptions. In reality, the target building was elevated on stilts, reducing accessibility to the target. To solve this situation, the planners were then given access to a MR planning tool to view the battlefield. This change in perspective offered planners a significantly different view of the target and considerably affected their planning considerations. Thus, the planners were able to identify the true entry points into the building.

Currently JSOC is seeking a new, real-time 3D capable system to enhance a common and shared visualization of the operating environment and believes an MR visualization system will best meet their objectives.

The gaming engine represents an essential component of such an MR system, providing the means of transforming raw information into a product beneficial for the warfighter. A gaming engine, or architectural framework for software development, is the backbone of most current MR systems. Gaming engines are programs that host software tools used by developers to construct games. Typical, gaming engines provide programmers with the ability to render 2D and 3D graphics, add sound and animations to a game. “Gaming engine” does not refer to the console used to manipulate or view a game such as, PlayStation, Xbox, PC, or HoloLens. The current market for gaming engines is wide in scope. Dozens of free, open-source gaming engines are available for developers to incorporate into their systems. These free and open-source gaming engines are best applied to basic programing functions and applications, which may not be suitable for JSOC’s complex needs. In contrast, the commercial market offers numerous gaming engine suites suitable for JSOC. These commercial market gaming engines, such as Frostbite, Game Maker, Unity and Unreal are commonly used in high-end video games.

JSOC must consider multiple inputs and outputs when selecting a gaming engine for military incorporation. The selection authority must understand the factors and subfactors prevalent in gaming engine functions and their association with the DOD’s existing architecture. Due to urgent needs and a constrained timeline, JSOC has already selected a market gaming engine to fill the immediate need. However, JSOC, along with other DOD entities, continues to evaluate other market gaming engines and technologies for the next iteration of MR technologies.

JSOC requires a structured methodology to selecting a gaming engine that meets the needs of end users. When conducted correctly, this methodology will enable JSOC to identify requirements, define relevant factors, and assess available gaming engines for incorporation into the planning tool.

## A. PROBLEM STATEMENT

Joint Special Operations Command lacks a formal selection methodology and requirements analysis for a gaming engine which is essential to the development of an MR mission planning tool.

The research team aims to answer one primary research question: What data-driven approach can be applied to select an appropriate gaming engine for a MR system that supports military mission rehearsals? In support of our primary research question, the team identified three supplementary questions:

1. What is a plausible gaming engine for a MR system?
2. What gaming engine characteristics are valued by JSOC?
3. How could the selection criteria be applicable to other JSOC system purchases?

## B. PURPOSE AND OBJECTIVES

The purpose of this research project is to develop a selection methodology to assess available market gaming engines to support JSOC's selection of a MR mission planning tool. This selection methodology must meet JSOC's current need. The team will present JSOC with a credible, repeatable, and traceable methodology to better inform MR gaming engine selection.

After developing a basic understanding of the problem domain and discussing the problem with JSOC, the research team developed the following research objectives. The objectives are essentially key tasks that drive the research team toward the overall purpose and are tied directly to the research questions.

1. Examine available market gaming engines that meet requirements for Department of Defense (DOD) application.
2. Investigate what military requirements drive gaming engine selection.

3. Identify what gaming engine characteristics are of most value to JSOC.

This project approach supports JSOC's iterative development method involving a process that constantly refines the approach to gaming engine comparison and selection. This project will validate the decisions made, inform better system tradeoffs, and support future design iterations.

THIS PAGE INTENTIONALLY LEFT BLANK

# II. LITERATURE REVIEW

The literature review seeks to integrate and generalize information regarding gaming engines and mixed reality (MR) technologies, including augmented reality (AR) and virtual reality (VR) fields. By using a three-pass literature review approach, the research team synthesized foundational knowledge, reviewed technological evaluations, and evaluated varying selection methodologies used in AR/VR gaming engine selection. In addition, the research team amassed information across the MR spectrum and filtered applicable knowledge into categories. Categories of knowledge include foundational knowledge, evaluation knowledge, and selection methodology. Based on this review, researchers have identified significant research gaps for military and industry selection methodologies. Identification of this gap provided justification to continue research in this field.

## A. LITERATURE REVIEW APPROACH AND PURPOSE

The researchers have considered archival literature surrounding mixed reality, augmented reality, and virtual reality systems. The Naval Postgraduate School Dudley Knox Research Library serves as the primary vehicle for research. The library provided access to over 63 research guides covering 15 subjects. Of those available guides, the research team has focused efforts on subjects related to Science and Technology, Military Information, Aerospace and Engineering, and Operations Research and Analysis. The researchers have captured articles that support the team’s understanding of the basic components and characteristics of MR systems and how MR systems were evaluated or selected in the past. The literature review approach follows a five-step process for research: Source Identification, Source Selection, Source Assessment, Source Analysis, and Presentation.

The purpose of this literature review is to gain a collective understanding and inform the research team and the stakeholder of existing literature and concepts on gaming engines and MR technologies. Furthermore, three primary goals have been established for the literature review: Close the knowledge gap pertaining to MR systems, build foundational understanding of key terms and characteristics of MR systems, and identify historical evaluation and selection methodologies for MR systems.

## B. FINDINGS

Over the course of the literature review, the research team made three findings. First, the research team built foundational knowledge to communicate with stakeholders and validated the need for research. Secondly, the research team discovered historical selection methodologies which can be used to support our research. Finally, the research team identified key observations describing test and evaluation techniques used by industry and gaming for MR systems.

### 1. Foundational Knowledge and Common language

Through literature, the research team built a strong foundational understanding encompassing the spectrum of mixed reality that enabled them to communicate with stakeholders in a common language. In the process the research team identified a gap in research pertaining to selection criteria within DOD. Moving forward, the research team is confident it can provide the stakeholder value in the MR realm. Furthermore, researchers and stakeholders now have a common language bridging the gap in ontology. Bridging this gap allows the research team to better understand the stakeholder needs and answer those needs with market solutions.

The research team recognized the gap in academic, peer-reviewed research regarding mixed reality technology. As a result, the research team should continue to review industry sources to remain abreast of emerging technologies. The research team will continue to use the methodology described in this literature review to find relevant sources. The research team learned that AR/VR systems are currently being employed by multiple industries with successful outcome including military use. In fact, the DOD recently awarded the Joint Enterprise Defense Infrastructure (JEDI) contract to Microsoft which includes providing AR capability using their HoloLens (Haselton 2019). Although the JEDI contract sounds promising, AR/VR is in its infancy stage; its capabilities and uses are still being imagined by industry. Furthermore, while multiple alternatives exist for viable solutions, ultimately, the research team will filter its choice through the criteria of the stakeholder’s needs.

The team’s research assesses the gaming industry is the driver behind introducing the most current and leading technologies in the MR space. The research team also believes this gap is significant to the project because most relevant solutions will likely come from technologies that currently exist in the gaming industry. However, the gaming industry as source poses a challenge because of its secretive nature. Companies invest significant amounts of money in protecting proprietary information and technologies. Their secrecy makes it difficult to find literature that gives definitive answers on how to define the technologies, their capabilities, constraints, or how the industry intends to use them. This proprietary information poses a challenge that must be addressed.

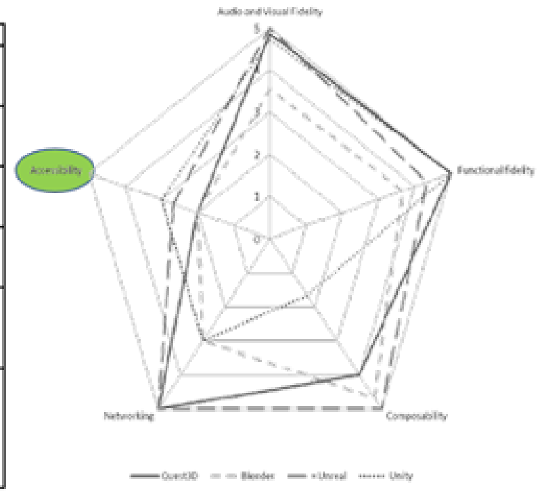
### 2. Selection Methodology

Existing literature on selection methodology provides the research team a foundation that may be used as a framework in future research. There are several selection methodologies offered by different literatures. One such article, “An Engine Selection Methodology for High Fidelity Serious Games” developed a sound selection methodology and evaluation design in selecting the appropriate gaming engine (Petridis et al. 2010). Petridis et al. (2010 p.33) provides a selection method that considers the importance of selecting the correct gaming engine and serves “as a starting point for a wider project towards overcoming particular issues with respect to composability.” The selection criteria of gaming engines in the context of serious games describes categories for ranking gaming engine that could provide utility as the research team moves forward with developing a research selection methodology. Petridis et al. (2010) used a qualitative approach, incorporating binary, ordinal, and nominal evaluation factors to select a gaming engine. As illustrated in Figure 2 Petridis et al. Selection Methodology, evaluation criteria include binary, gradient, and subjective components.

TABLE V. ACCESSIBILITY

| | Metrics | Quest3D | Blender | Unreal | Unity |
|---|---|---|---|---|---|
| *Learning Curve* | | Steep Learning Curve | Medium | Medium | Medium |
| *Docs and Support* | Docs Quality | Docs and tutorial | Docs and tutorial | Docs and tutorials | Docs and Tutorials |
| | Technical Support | Yes | No | Yes | Yes |
| | Community Support | Yes | Yes | Yes | Yes |
| *Licensing* | | Creative, Power and VR Editions | Open Source | Commercial Licence. | Indie and Pro Version |
| *Cost* | | Cost is between 1249, 2499, 9999 euros | free | free for educational purposes | 149 and 1099 Euros respectively |

**Evaluation Metric: Accessibility**




**Combined Metrics**

Figure 2. Petridis et al. Selection Methodology. Source: Petridis et al (2010).

The study performed by Petridis et al. subjectively evaluates rather than quantifies the differences in gaming engines. The radar graph in Figure 2 shows where gaming engines outperform each other but do not quantify which gaming engine is the best. Ultimately, Petridis's et al. research focused on a binary approach that evaluated gaming engines based on various features. This binary approach was useful to the research because it allowed researchers to reduce many alternatives to a more manageable handful of alternatives based on available features.

In contrast to the binary selection methodology, Pattrasitidecha used shades of gray to rank the different categories (Pattrasitidecha 2014). This provided a sliding scale for how well each software performed. This scaled approach provides fine-tuned fidelity that determines the value of each category while assessing performance of each alternative. Pattrasitidecha's research could prove useful in developing a selection methodology for gaming engines. Even though the article focuses on mobile gaming engines, the comparison matrix provides a structured method for comparing multiple different engines. This comparison matrix could be tailored to incorporate the stakeholder's requirements and weighted needs to assist with a selection methodology for gaming engines. There are also similarities between Pattrasitidecha's comparison matrix and the systems engineering process of quantifying different value criterions. However, the primary difference between Pattrasitidecha's approach and the approach typically used in systems engineering is the absence of weighted values for each criterion.

Both selection methodologies provide value for the stakeholder because in some cases, a binary "Yes/No" approach is appropriate, and at other times, researchers need the ability to rank the importance of the different selection criterions and evaluate the performance of each system against those categories.

The research team required a selection methodology that could adapt to JSOC's increasingly fluid environment. Therefore, the binary approach and the scaled approach can be blended to create a selection methodology. The research team may be able to develop a weighted selection methodology that incorporates ideas from both Pattrasitidecha and Petridis et al. The result may be able to provide the flexibility needed in JSOC's operational environment while also advancing selection methodologies.

### 3. Evaluation

For this literature review, evaluation pertains to the means and techniques that MR systems and designs were tested and evaluated by the military, industry, and gaming sectors. The review of these articles as they relate to evaluation gives insight and refinement to a selection methodology for gaming engines. Existing literature on evaluation criteria provides the capstone team a foundation that may be used as a framework for future research. The research team identified three observations within this framework. First, through experimentation, researchers can collect quantitative and qualitative data distinguishing system characteristics and performance. Krichenbaur et al. showed how systems can be evaluated and demonstrates user understanding in a digital environment (Krichenbaur et al. 2018). This technique can be applied in solving the stakeholder's problem. The second observation reveals user feedback can be employed as an effective approach to measure system performance (Tremblay et al. 2011). Therefore, stakeholder feedback can be used to validate key characteristics important to system designs. These characteristics can then be transformed into data points for analysis and integrated into a selection methodology. The third observation are the comparative analyses executed by Mat et al. (2014) and James Burrow (2019). The analysis shows characteristics found in gaming engines. Most important to the stakeholder is the inclusion of the direct comparison of Unity and Unreal game engines amongst other.

In addition to works by Mat et al. (2014) and James Burrow (2019), the research team found works by Xinogalos useful in the team's research. "Overview and Comparative Analysis of Game Engines for Desktop and Mobile Devices" is an article that seeks to educate the reader on how to make an informed decision when determining which gaming engine best meets their need. Essentially, the article is closely related to the topic of this capstone project in that the capstone team seeks to conduct a comparison of gaming engines in order to ensure JSOC is informed when making a decision on which gaming engine is the most suitable to meet their needs. The article flows intuitively by sections. The article has five sections; they are the introduction, related work, the game engines selected for the comparison, the framework used to compare the alternatives, and the results of the selection (Xinogalos 2017, 22). Section four of Xinogalo's paper provides the framework that is used to compare the gaming engines. The framework is composed of the following eight comparable categories: audiovisual fidelity, functional fidelity, composability, developer toolkits, accessibility, networking, development features, and deployment platforms. Each of the eight categories have two or more sub-categories that are evaluated. The results of each category is displayed in separate tables that illustrate to the reader which game engines are superior as it relates to the category being compared (Xinogalos 2017, 23-25). Figure 3 Xinogalos' Accessibility Feature Comparison highlights the method used for our analysis.

| ***Features/ Game Engines*** | **Game Maker** | **Jmonkey** | **Marmalade** | **Ogre3D** | **Shiva** | **Sio2** | **Turbulenz** | **Unity** | **Unreal Engine 4** |
|---|---|---|---|---|---|---|---|---|---|
| **1. Usability** | | | | | | | | | |
| 1.1 Easy of learning | | | | | | | | | |
| 1.1.1. Tutorials[2] | 3 | 4 | 4[2] | 4[2] | 3[2] | 3[2] | 2 | 5[2] | 4 |
| 1.1.2. Examples[2] | 2 | 4 | 4[2] | 4[2] | 3[2] | 3[2] | 3 | 5[2] | 4 |
| 1.2 Docs and Support[2] | | | | | | | | | |
| 1.2.1. Docs Quality[6] | Docs and Tutoria ls in official | Docs and Tutoria ls in official | Docs and Tutorials in official site | Docs and Tutoria ls in official | Docs and Tutorials in official site | Docs and Tutorials in official site | Docs and Tutorials in official site, not | Docs and Tutorial s in official | A subset of tutorials is at Unreal |
| 1.2.2. Technical Support[6,2] | x | x | √ (not in free version) | X | √ (Basic and Advance | √ (not in free version) | √ | √ (Professi onal | √ |
| 1.2.3. Community Support | 3 | 3 | 3[2] | 4[2] | 3 | 3[2] | 2 | 5[2] | 4 |
| **2. Price**[6,2] | | | | | | | | | |
| 2.1. Free | Studio Free | √ | Free version | √ | Web Edition | Trial version | √ | Personal version | √ |
| 2.2. Open Source | x | √ | x[2] | √ | x[2] | Certified Develope | √ | Professi onal | √ |
| 2.3. Price/Year | Studio Profess ional version : 149,99 $+299, 99$, | - | Indie version: 499$, Plus version: 1500$, | - | Basic new version: 200$ + tax, Advance d new version: | SIO2 Certified Develope r version: 1999$, SIO2 Licensed | - | 395$: Plus version, 1500$ Professi onal version | 5% Commis sion after first 3000$ of the first game / 3 |

Figure 3. Xinogalos’ Accessibility Feature Comparison. Source: Xingalos (2017).

Ultimately, the team used the functions found in the framework as a baseline. With the baseline functions identified the team tailored a new functions list to match JSOCs stated needed. The team used this tailored list to develop requirements for the gaming engine. The research team can build on the comparative analysis to highlight differences between alternatives.

The research team discovered that previous researchers evaluated gaming engines based on criteria in the commercial sector that has not yet been applied within a DOD context or to a comparison that could provide value to JSOC. While certain evaluation criteria used in the commercial sector can inform this selection methodology, there are factors unique to the DOD, such as compatibility with existing systems, networking, and operational security considerations, that provide justification for further exploration to identify evaluation criteria within the DOD context. While existing literature can inform gaming engine selection, this research team must consider additional variables specific to JSOC and the DOD when developing evaluation criteria.

# III. FRAMEWORKS AND RESEARCH DESIGN

The purpose of this chapter is to describe the methodology used to conduct research, identify criteria for comparison, and develop a selection methodology for MR gaming engines. This chapter is composed of two primary sections: theoretical framework and research design. In theoretical framework, the team explains the basis of reasoning used to develop their approach to solve JSOC’s need. In the second section, we explain why our research design is based on the System’s Engineering “V” and the subsequent decomposition of critical tasks into three major categories: Problem Understanding, Requirements Definition, and Value Analysis.

## A. THEORETICAL FRAMEWORK

The research team’s theoretical framework is grounded in inductive and material-based reasoning. In the article “Reasoning in Engineering Design,” Joshua Summers describes inductive reasoning as a “class of reasoning that seeks to generate appropriate design knowledge based upon the given set of design variables and design specifications” (Summers 2005, 5). Figure 4 Design Knowledge displays Summer’s supplemental inductive pattern. The project team will apply Summer’s model as a reasoning framework to reach overall conclusions to the problem.

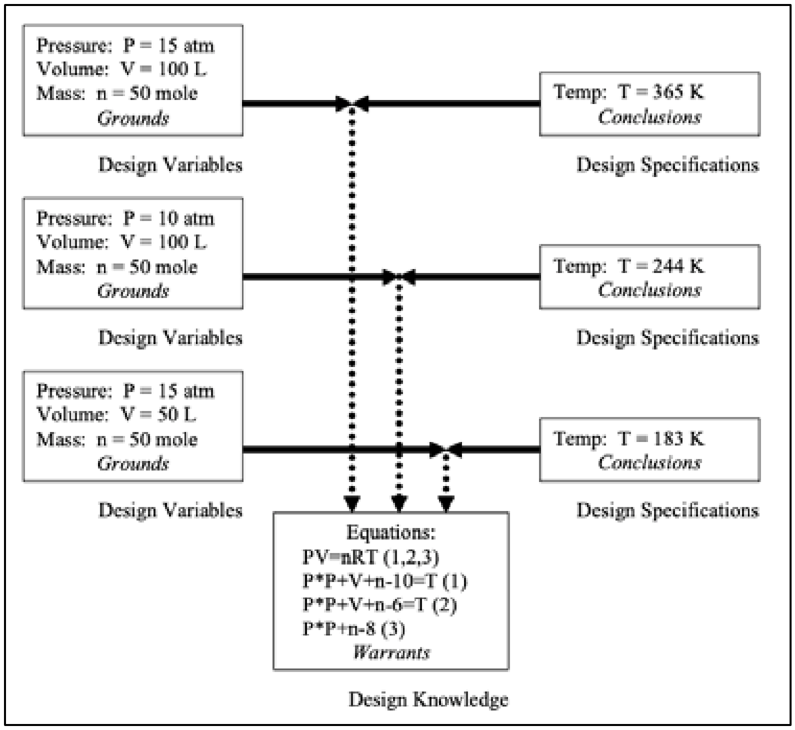



Figure 4. Design Knowledge. Source: Summers (2005).

Design knowledge is developed using data collected from design variables and specifications. Design specifications or conclusions are developed from primary stakeholder input on what functions the MR system must accomplish. The research team will collect design variables (grounds) and design specifications (conclusions). Design variables or grounds will be collected from gaming engine developers in the form of system specifications. Design variables and specifications required the transformation of qualitative data into quantitative data. Once transformed, variables and specifications can be compared and evaluated to support design knowledge. The design knowledge will then be translated into a repeatable, verifiable, and traceable tool for the selection of alternatives. Figure 5 Theoretical Framework displays a modified theoretical framework that directly addresses how the research team will use Summers' model to reach a solution.

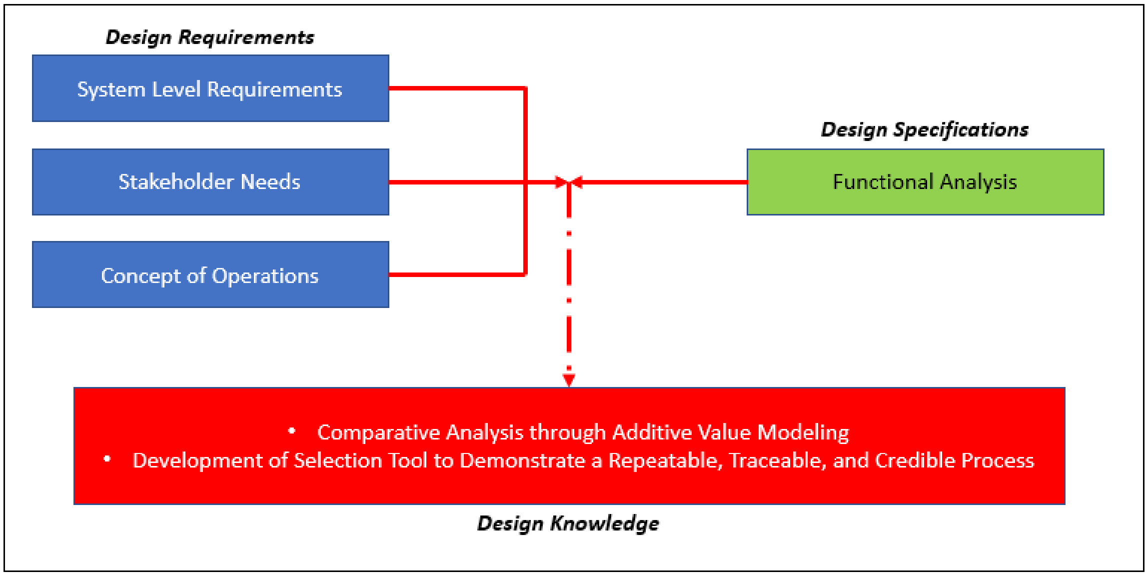



Figure 5. Theoretical Framework. Adapted from Summers (2005).

The research team determined that material-based reasoning (Worsley et al. 2016) is appropriate, based on their research in available market gaming engines. Therefore, the analysis and selection must be founded on the physical characteristics of market gaming engines. Furthermore, any design solutions developed must be bound by the physical (material) limitations and capabilities of the gaming engines.

The research team developed an expression to visualize their theoretical framework: Design Requirements + Design Specifications = Design Knowledge. Design requirements will be produced via stakeholder analysis and operational analysis. Design specifications will be produced through functional analysis. Design knowledge is defined as the results of comparative analysis between design variables and specifications. The research team applies this theoretical framework to generate design knowledge on MR gaming engines.

## B. MODEL-BASED SYSTEMS ENGINEERING

This research is conducted in an environment where stakeholder needs, system requirements, and emerging technologies require an ability to communicate system specifications into an easily understood format. Model-Based Systems Engineering

(MBSE) allows the research team to capture stakeholder needs and system functions into traceable system requirements, using diagrams and tables to describe behaviors and attributes of the system. The National Defense Industrial Association (NDIA) defines Model-Based Systems Engineering as “an approach to engineering that uses models as an integral part of the technical baseline that includes the requirements, analysis, design, implementation, and verification of a capability, system, and/or product throughout the acquisition life cycle” (National Defense Industrial Association [NDIA] 2011, 9). MBSE enables the research team to capture stakeholder needs and functional requirements, produce diagrams explaining the various behaviors of the system, and provide a creditable and traceable process in line with the systems engineering “V.”

To conduct MBSE, the research team had to select a software tool and modeling language. The research team chose Cameo Enterprise Architecture (CAMEO) because it provides a powerful tool to record and update requirements as stakeholder needs evolve and technology changes. Using this program allowed the research team flexibility and the ability to adapt more rapidly to newly discovered technologies. Cameo connects all elements of the system together, so any change to requirements or system function updates in real time across dozens of system diagrams, tables, and models. Cameo allows the research team to make changes efficiently across dozens of diagrams, tables, and models.

The research team chose the Systems Modeling Language (SysML) as our chosen modeling language because it is the modeling language the research team is most familiar with. SysML is “a broad and richly expressive graphical modeling language, enabling you to visualize and communicate the essential aspects of a system’s design: structure, behavior, requirements, and Parametrics (mathematical models)” (Delligatti 2014, 11). SysML thus provides the research team with the ability to depict complex system behavior, attributes, and requirements into a graphical model that stakeholders and advisors can easily understand and visualize. Ultimately, SysML allows the research team to translate raw needs and system functions into diagrams that explain the various behaviors and functions of the system.

## C. RESEARCH DESIGN

The research team referenced the traditional systems engineering, Figure 6 System's Engineering "V" (Department of Transportation 2007) and mixed methods to collect and analyze data to develop a selection methodology for JSOC.

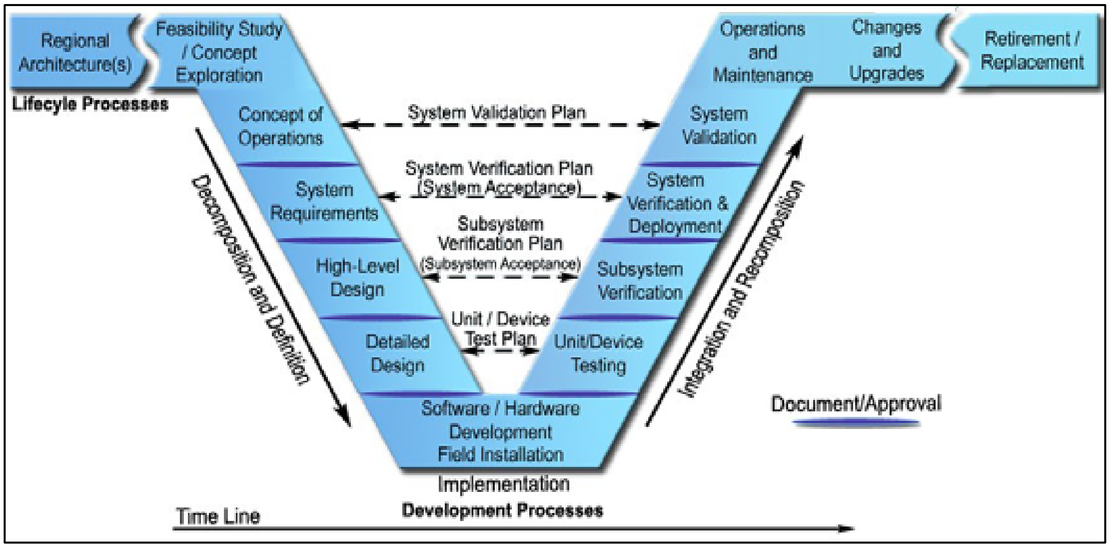



Figure 6. Systems Engineering "V." Source: Department of Transportation (2007).

The Systems Engineering "V" (SEV) provided a framework that guides the selection of a suitable alternative from a set of options. The International Council of Systems Engineering (INCOSE) Handbook defines the Systems Engineering "V" as "a sequential method used to visualize various key areas for SE focus, particularly during the concept and development stages" (INCOSE 2015, 33). The research project seeks to help JSOC identify and select a gaming engine by examining needs and sensitivities prior to the selection of a gaming engine. The MITRE Corporation, a non-profit manager of federally funded research and development centers, describes the SEV as "common representation of the systems engineering life cycle…" where the "…left side of the V represents concept development and the decomposition of requirements into functions and physical entities that can be architected, designed, and developed" (MITRE 2014, 2). The SEV is useful to the research team because the research project aligns with concept development and decomposition of requirements for architecting a future system. In focusing on the SEV's advantages, the International Council on Systems Engineering (INCOSE) states that the SEV "…highlights the need for continuous validation with the stakeholders…" (INCOSE 2015, 33). Stakeholder involvement throughout the research project is crucial because validation and criticism of the research finding will drive the subsequent direction of the research team.

The SEV model provides a structured approach to solving complex engineering problems. The research team operates on the left side of this SEV model, conducting decomposition and definition of the operational concept, system requirements, and system design. In attempting to solve an ill-structured problem, the project team required a hybrid application of this model. This hybrid application enabled the research team to adapt the SEV to the current problem. The research team was then unconstrained by iterative steps, allowing freedom of movement up and down the model, based on communication with JSOC and advisor input. The SEV is accepted by the NPS systems engineering department and the research team has practice implementing this model. Therefore, a hybridized SEV was used throughout this research project.

The tasks in this effort are categorized in three major project phases: Problem Understanding, Requirements Definition, and Value Analysis. The research team will provide JSOC with a deliverable at the conclusion of each of these major phases, as well as In Progress Reviews (IPRs) throughout the project. At the completion of the Value Analysis, the research team is postured to present JSOC with a credible, repeatable, and traceable methodology to better inform MR gaming engine selection.

### 1. Problem Understanding

The research team will analyze the problem space by conducting stakeholder analysis, literature review, and operational experiences. The research team will employ an Ishikawa diagram and stakeholder analysis to define the problem and progress through the Feasibility Study and Concept Exploration phases of the SEV Model (Figure 6). The literature review will provide a foundation that gives the research team general knowledge within this field of study. The research team will use operational experiences and JSOC input to develop an Operational Illustration. The operational illustration will provide a depiction of intended use of the gaming engine by JSOC and how the gaming engine is integrated into the larger system of systems.

### 2. Requirements Definition

This phase consists of research data Requirements Definition: This phase consists of research data collection on market gaming engines and high-level requirements definition. Data will be collected and categorized in two separate areas: 1) data relating to the problem area such as use-case parameters and system requirements generated by JSOC; or 2) data on state-of-the-market gaming engines such as manufacturer's system specifications. The research team will establish regularly scheduled IPRs to remain in constant continuous communications with JSOC. At the conclusion of this phase, researchers will categorize and code system specification data on available market gaming engines and use that data, along with JSOC input, to develop high-level system requirements.

### 3. Value Analysis

During value analysis, the research team considered stakeholder needs, operational needs, and functional allocation to inform Measures of Performance (MOP) and Measures of Effectiveness (MOE) in the form of a value hierarchy. The value hierarchy allows the research team to make tradeoff decisions between conflicting or unobtainable requirements. These tradeoff decisions will then establish the selection criteria to be used for Value Modeling and alternative comparison.

The research team uses Multiobjective Decision Analysis and Additive Value Modeling as the basis for producing a credible, repeatable, traceable selection tool to compare alternatives. Dr. George Parnell defines Multiobjective Decision Analysis as "a technique that focuses directly on complex decisions, multiple objectives, and uncertainty" (Parnell and Trainor 2009, 2). Parnell and Trainor further describe the Additive Value Model with the equation, $v(x) = \sum_{i=1}^{n} w_i v_i(x_i)$ "where *v(x)* is the alternative's value, *i* = 1 to *n* is the number of the value measure (attribute), xi is the alternative's score on the ith value measure, $v_i(x_i)$ = is the single dimensional value of a score of $x_i$, $w_i$ is the weight of the $i$th value measure, and $\sum_{i=1}^{n} w_i = 1$ $(all\ weights\ sum\ to\ one)$" (Parnell and Trainor 2009, 3). Within this research report, this model will be referred to as the MR Gaming Engine Value Model defined in Figure 7. This research will utilize Dr. Parnell's Swing Weight Matrix to weigh different alternatives using level of importance (defined by the stakeholder) and level of variance between alternatives to elicit swing weights (Parnell and Trainor 2009). The MR Gaming Engine Value Model and Parnell Method for eliciting swing weights will be discussed in depth in Chapter IV as the research team walks through the Value Modeling process.

**MR Gaming Engine Value Model**

Total Value of a Gaming Engine = $v(\mathbf{x}) = \sum_{i=1}^{n} w_i v_i(x_i)$

Figure 7. MR Gaming Engine Value Model. Adapted from Parnell and Trainor (2009).

## D. DELIVERABLES

The research team developed primary deliverables at the onset of this project to ensure research remained focused and delivered quality results to JSOC. Based on the research design, the research team developed the following deliverables:

1. Operational illustration.
2. Selection criteria for state-of-the-market gaming engines.
3. A repeatable, traceable selection methodology utilizing the MR Gaming Engine Value Model.

Furthermore, the research team identifies several outcomes of a successful research project. These strategic outcomes include a selection methodology that:

1. Satisfactorily incorporates desired system characteristics.

2. Weighs system characteristics in accordance with importance to the stakeholder.

3. Applies to commercial gaming engines currently used for system development.

4. Applies to the current stakeholder and other DOD entities.

5. Provides a clear evaluation metric of gaming engines.

THIS PAGE INTENTIONALLY LEFT BLANK

# IV. ANALYSIS AND FINDINGS

## A. INTRODUCTION

This chapter discusses results from activities involved with different phases of our study, which includes Problem Understanding to Requirements Definition, and, finally into Value Analysis. The research team first sought to understand JSOC's problem, using several techniques to identify causes, effects, and key stakeholders. Then, the research team defined requirements inherent to gaming engines within the DOD. Finally, the research team applied Value Modeling to develop a traceable, repeatable scoring method to support alternatives selection in future MR acquisitions. The value model allows for an analysis of multiple alternatives with multiple criteria by providing a tool that gives program offices the ability to assess multiple gaming engines in a quantitative and comparative method. The result of this value model provides decision makers a rational for choosing which gaming engine best fulfills the stakeholder needs.

## B. PROBLEM UNDERSTANDING

Entering the capstone process, the research team quickly identified a large amount of uncertainty surrounding the proposed problem. According to Professor Gregory Miller, a Senior Lecturer at the Naval Postgraduate School, "The second most important tool used by the systems engineer is that of defining the problem. If the correct problem is defined, then every action thereafter is efficiently focused on a solution that is wanted by the customer" (Miller, 2019a). What were the stakeholders trying to understand? What was pertinent to solving the problem? What were the possibilities of the potential solution space? At the beginning of the research project, none of these questions were clear. However, the need for an organized and structured approach to define the problem was overwhelmingly evident.

Therefore, the research team utilized a system engineering V process to begin to understand the problem. The team used several tools from the "V" process to explore the problem area including problem definition and bounding, stakeholder analysis, incorporation of research, scenario-based assessments, and user need identification.

Through iterative and transparent communication with the stakeholder, the research team successfully transformed an unwieldy and unbounded problem into one that was manageable and solvable. To properly understand the problem, the research team structured this phase into three categories: 1) Defining and Bounding the Problem; 2) Stakeholder Analysis and Needs; and 3) Operational Concept Description.

### 1. Defining and Bounding the Problem

#### *a. Incorporation of Research*

As identified in Chapter I of this report, the literature review provided a better understanding of game engines and mixed reality technologies. This review led to three findings which the research team used to help shape their understanding of the problem. In summary, the research team gained foundational knowledge of gaming engines, identified gaming engine selection methodologies previously proposed, and gathered important observations describing test and evaluation techniques used by industry for MR systems. Because the literature review was the research team's first academic and in-depth technical exposure to both MR technologies and gaming engines, it formed the research team's first attempt at problem understanding. This knowledge then formed the research team's initial understanding of and potential bias of toward MR and gaming engine technologies.

#### *b. Tools and Techniques*

The research team required a clear understanding of the problem and delineation of the problem from others within the MR mission planning tool domain. They also required a focused, or bounded, subset of the problem space that was feasible to solve with the time and resources available. Stakeholder engagement and root cause analysis are techniques that define and bound problems.

The research team performed stakeholder engagement through meetings, emails, and document sharing. Through this interaction, the research team established shared understanding and trust with the stakeholder. This communication allowed the research team to identify the current situation and the desired situation (or end state).

*Current Situation:* *no comparison of state of the market gaming engines*

*Desired Situation:* *comparison results detailing where value is within the gaming engine and which gaming engine best meets JSOC's need*

The team developed a communicated plan of action to ensure that our research moved the stakeholder from their current situation to their desired situation. This communication was iterative throughout the project research which fostered transparency between the research teams' goals and stakeholder needs.

The two techniques the team used to help define and bound the problem were the Ishikawa cause and effect diagram and the "Five Why's" (Miller, 2019a) root cause technique. The Ishikawa diagram, also called a fishbone diagram, enabled the research team to better understand the linkage between the stakeholder's current situation and desired situation. The research team also used the "Five Why's" technique to identify root cause through exploration of cause and effect relationships and identification of the underlying problem. Both techniques aided the research team in scoping the problem space and preventing the team from solving a symptom of a problem rather than the true problem itself.

MR technology is a broad and emerging technology which is the product of many systems. The knowledge gained from these techniques and repeated communication with the stakeholder supported bounding the problem. With direction and concurrence from the stakeholders, the team focused its efforts on the gaming engine as the primary focus of this research project. The stakeholders and the research team determined the MR system in aggregate was out of scope. Bounding the problem to the gaming engine refined the research team's understanding of the current and desired situations.

#### *c. System Context Diagram*

The research team used context diagrams to better visualize the gaming engine's interactions with external systems. Miller explains, "Graphical methods are a great way to capture relationships and our ideas about boundaries, boundary conditions, interfaces, and defining what is inside or outside of our system…" (Miller, 2019b). The research team developed Figure 8 Gaming Engine System Context Diagram to highlight inputs and outputs between the gaming engine and external systems. Of note, the external systems highlighted by the research team include non-human systems and human stakeholders. It was important for the team to understand the stakeholders' inputs and effects on the gaming engine in the same view as non-human systems so that the team could discern the high-level purpose of the gaming engine. The system context diagram defined the boundaries of the system and relationship between entities, both internal and external, enabling a clear understanding of the gaming engine's role within the MR System of Systems (SoS).

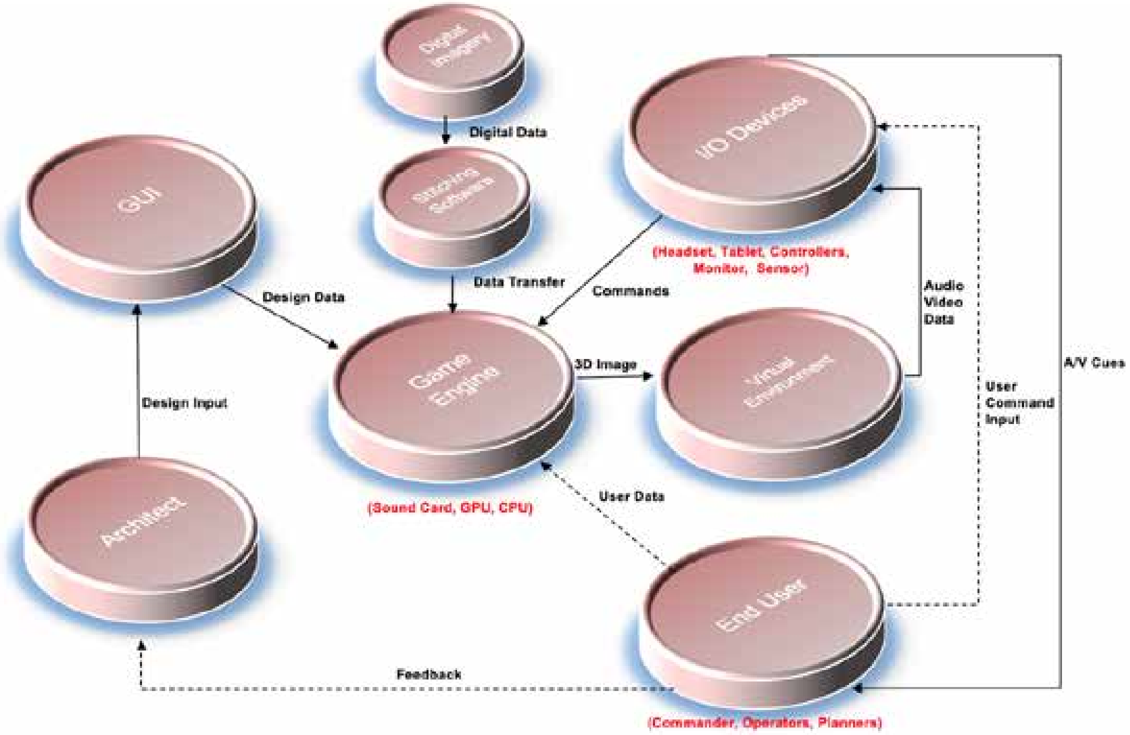



Figure 8. Gaming Engine System Context Diagram

### 2. Stakeholder Analysis and Needs

#### *a. Stakeholder Analysis*

Stakeholder analysis is essential because it identifies unique and competing needs between the different entities involved in the system's development. Because interested parties are identified early, stakeholder analysis enables a forum for engagement and resolution between conflicting interests. The research team first identified all possible parties that could benefit or could be affected by the MR planning system. Next, the research team categorized each stakeholder, identifying needs and concerns from the perspective of each, and prioritizing them by level of influence on this research project. JSOC validated the stakeholder analysis, identifying several areas of the initial analysis that differed from their assessment. Table 1 Stakeholder Analysis displays the results of the stakeholder analysis for the MR gaming engine. This figure also describes the complexity and uniqueness of the JSOC organization and their associated acquisition process. The research team used JSOC’s MR Mission Planning Development Team Lead as the point of contact for all acquisition and stakeholder feedback. Additionally, JSOC places significant trust into the special operators or users of their systems. This decision makes sense in that the users of JSOC systems are flush with operational experience, extremely technically proficient, and drive the requirements of most JSOC acquisitions. In this aspect of MR and gaming engines, the special operators have much more influence than the organizational leadership, JSOC.

Table 1. Stakeholder Analysis

| Stakeholder Analysis | | | | | |
|---|---|---|---|---|---|
| **Priority** | **Stakeholder** | **Type** | **Needs** | **Concerns** | **Notes** |
| 1 | Special Operator | User | - Precise Rehearsals<br>- Improved Situational understanding<br>- Tactical Advantage | - Geographical Disperse users<br>- Timeliness<br>- Relevancy<br>- OPSEC | Primary user of the system. Negotiates the system to rehearse the mission prior to execution. The relevance is dependent upon the system and other stakeholders to generate an accurate representative model of the target environment. |
| 2 | Planner | User | - Precise Planning<br>- Real Time Feedback<br>- Improved Situational Awareness and shared understanding | - Planning Speed, Accuracy, and Fidelity<br>- OPSEC | Primary user of the system. Develop initial plan, collect operator feedback, and re-plan the mission. Relies upon Comms/Arch team for target template and operators/commanders for planning constraints. |
| 3 | Commander | User, Approver, Oversight | - Battlefield Geometry<br>- Risk Assessment<br>- Informed Decision Making | - Timeliness<br>- Accuracy<br>- Risk Picture | Secondary user of the system. Provides input to planners/operators and will use output of Operators and Planners mission rehearsal to will ultimately approve mission. |
| 4 | Program Manager | Acquirer | - Stable Requirements<br>- Predictable funding stream<br>- Stable and predictable timeline | - Unclear requirements and changing requirements<br>- Foreign ownership of Contractor | Overall Acquirer of system. Responsible for delivering a system that meets warfighter needs within cost, on schedule, and at or above the performance parameters. |
| 4a | Software Developer | Integrator | - Requirements<br>- System Vision<br>- Coding Language | - Feasibility<br>- Fast Pace Technology Changes,<br>- Changing requirements | Primary integrator of technology. Writes APIs, ensures VR environments integrate with existing platforms. |
| 4b | Hardware Manufacturer | Supplier | - Performance Specs<br>- System Limitations | - Buy-American Compliance<br>- Requirements, Counterfeit hardware avoidance | Provides commercial hardware for use by system which meets primary and secondary user's intent. |
| 5 | Comms and Architecture Support Team | System Enabler, System Maintainer | - Maximize Interoperability with existing DoD Infrastructure,<br>- Secured high fidelity transmissions | - Policy/Regulation<br>- OPSEC | Supports the system and users by providing and maintaining a networking capability and enabling data flow across the system. |
| 6 | Data and Platform Support Team | System Enabler, Platform Creation, Data integration | - Input data to generate models<br>- Guidance from intel shop<br>- Guidance from planners,<br>- Commands from operators,<br>- Structure for Output | - Accuracy, fidelity currency of Inputs<br>- Clear and coordinated guidance from intel and planning cells | Supports the system and users through the translation of data into a representative, virtual environment. |
| 7 | JSOC | Higher Headquarters Capability Owner Requirements Owner | - Fill Capability Gap<br>- Tactical Advantage<br>- Increased operational flexibility and responsiveness | -OPSEC<br>-Feasibility<br>- ROI | Overall responsible for approval, use, and application of the system. Serves as a strategic entity. Delegates detailed requirement decisions to users, planners and program manager. |

The stakeholder analysis resulted in a refined understanding of the stakeholders' needs and desires. First, the analysis identified three stakeholders as 'users' of the system. The research team identified the Special Operators who would be donning VR headsets and performing rehearsals inside the virtual environment as the most important user of the MR system. Through prioritization, the three most important stakeholders were all users of the system. This understanding is important because it means the users, vice the enablers, designers, or support staff, will dominate the development of the system. As a result, the research team more heavily weighed the needs and concerns of the users.

The team members that represent JSOC's MR Mission Planning Tool Development Team are comprised of five primary entities within JSOC: 1) Military Communications, 2) Military End Users and Operators, 3) Contractor Developers, 4) Software Engineers/Data Scientists, and 5) Civil/Military Security. Military Communications provide input to vendors and test/integrate the VR capability into existing systems. The Military End Users and Operators test the system and provide feedback to military communications. The Contractor Developers serve as the vendors on contract to provide the user interface (UI) and specific features requested by military communications. The Software Engineers and Data Scientists complement military communications and ensures data can integrate into the VR system. Lastly, Civil/Military Security assesses vulnerabilities in the system's hardware and software. Thus, this development team is the primary conduit of input on stakeholder wants and needs for the MR gaming engine represented in this project.

#### *b. Stakeholder Needs*

Stakeholder needs directly contribute to requirements. The challenge for many system engineers and this research team is translating primitive needs into effective needs. Primitive needs (or wants) are the result of the opinions of stakeholders. Conversely, effective needs are supported by evidence and are worth additional cost and risk to meet the need. (Miller, 2019a). The research team worked with the stakeholders to refine the wants described in the stakeholder analysis and transform them into effective needs. Table 2 Stakeholder Needs captures the refined needs of JSOC.

Table 2. Stakeholder Needs

| # | NAME | TEXT |
|---|---|---|
| **1** | **Security** | **The gaming engine must be able to operate within the DoD's secure environment.** |
| 1.1 | Closed-Loop Network | Gaming engine must be capable of operating within a closed-loop, secure network. |
| 1.2 | Manufacturer/Developer Origins | The gaming engine developers must not have significant ties to U.S. adversaries. |
| **2** | **Interoperability** | **The gaming engine must be interoperable with certain data inputs and external DoD systems.** |
| 2.1 | Read/accept data inputs | JSOC requires gaming engine to be able to accept and identify multiple file formats of data inputs entering the system. |
| 2.2 | Sensor Fusion | JSOC believes sensor fusion is ill-defined, but is necessary for the accomplishment of mission. |
| 2.3 | Android Compatibility | Since Android Team Awareness Kit (ATAK) is an android-based system, the gaming engine must be compatible with multiple languages. |
| 2.4 | Multi-platform support | The gaming engine must be able to render across a broad range of platforms (Windows-based PCs, Android devices). |
| 2.5 | JSOC Network Compatibility | The gaming engine must be capable of communicating within the JSOC's network infrastructure. |
| 2.6 | Bluetooth compatibility | Local communications between headset/controllers and CPU must be capable of transmitting securely via Bluetooth. |
| **3** | **Gaming Engine Performance** | **The gaming engine must be able to perform virtual reality in a real-time environment over a secure network.** |
| 3.1 | Frame Rendering | Gaming engine must be able to render data inputs into a 3D, virtual environment. |
| 3.2 | CPU Usage | Gaming engine must be able to manage CPU. |
| 3.3 | GPU Usage | Gaming engine must be able to manage GPU. |
| 3.4 | Network Usage | The gaming engine must work within the constraints of the current network bandwidth limitations. |
| 3.5 | Machine Learning | The gaming engine must be capable of producing automated "opposing forces" that can act independently and make adaptive decisions based on user actions and within the general scenario and enemy's tactics. |
| 3.6 | Response Time | Users must be able to interact as close to real-time as possible from different geographical areas. |
| 3.7 | Hard-wired CPU into Network | The CPU used must be hard-wired into the network. |
| 3.8 | Virtual Reality | The gaming engine must render virtual reality environments. |

The research team identified three broad categories of needs espoused by the stakeholders. Security is an essential need so that JSOC may protect its systems and users from adversaries. JSOC, as a subordinate to the DOD, needs to maintain a secure digital environment. The MR mission planning tool, and naturally the gaming engine, must also maintain or enhance that security posture. JSOC also needs a system that is interoperable with the DOD information technology infrastructure. Therefore, the gaming engine must support system interoperability and also be interoperable with components of the MR mission planning tool already selected by JSOC. Finally, JSOC needs a gaming engine that meets performance objectives.

As stated previously, stakeholder needs will directly support the requirements used to procure a gaming engine that supports JSOC's MR mission planning tool. Ultimately, the system's functional requirements must trace back to each of these effective stakeholder needs.

### 3. Operational Concept Description

Operational concepts aid the research team in problem understanding by capturing high-level expectations of the system. To describe the operational concepts, the research team used scenarios, vignettes, and an operational illustration. Each description follows the stakeholder needs previously generated by the research team.

#### *a. Scenarios*

Scenarios are useful tools to help present and validate a common picture, create discussion between stakeholders, and identify key decision points. The research team created the scenario below to present an initial picture of the system fulfilling its need in its intended operational environment. The research team then presented an initial scenario to JSOC for discussion, modification, and feedback. The scenario below is the output of this interaction between the research team and the stakeholders.

*A high value target is identified in a small, remote village within Afghanistan. A small mission unit (SMU) has been tasked to conduct an extraction of the target within 96 hours. Due to time constraints and geographic dispersion, building a physical mock-up of the small village is not feasible. Therefore, the SMU requires a battlespace in which to conduct mission planning and* *rehearsals while geographically separated from planners and commanders. System developers develop the mixed reality environment mock-up of the real-world village. Developers receive data packages from various ISR platforms. Data packages can include aerial photographs, full motion video (FMV), and ground-based imagery among other methods. These data packages must be normalized into data that is compatible with the system. The system developer team will use this data to design a mock-up of the real-life location within the gaming engine. The gaming engine will allow the developer team to render 3D animation, 2D animation, and introduce operational variables through scripting and machine learning. Aspects of the rendered mixed reality world are tailorable to meet each operation's needs. After the mixed reality world is rendered by the gaming engine, information must be communicated in real-time to the geographically-dispersed units (operators, planners, and commanders). Due to the sensitive nature of the information being transmitted, data must be communicated over the DOD's secure network. The rendered mixed reality world must also be compatible with existing DOD mission command systems for commanders and planners to view within existing operations centers. With the rendered mixed reality world shared amongst users, planners can conduct mission planning while the SMU conducts rehearsals in real time. This mixed reality world facilitates shared understanding of battlefield conditions. To increase fidelity, system developers continue to provide updated battlefield visualizations based on changes to the real-world operating environment. The developer provides periodic updates until support is no longer requested from unit commanders. As a result of this mixed reality world, operators were able to experience the battlefield environment countless times before mission execution; planners were able to accurately forecast and project resources to support the capture of the high value target; commanders were able to provide clear intent and understand mission risks prior to execution. The fog of war is reduced, and operational uncertainties are mitigated.*

#### *b. Vignettes*

The researchers incorporated vignettes as a method to apply context from a real-world situation that JSOC could encounter. Vignettes allowed the research team, with stakeholder input and approval, to sequentially list activities from the start to the end of that specific situation.

Through the development of vignettes, it became clear that the team needed a structure to effectively communicate how the system would interact within JSOC. The research team selected Military Decision-Making Process (MDMP) as the standard planning tool used in military operations. As such, JSOC would use some form of planning doctrine like MDMP to organize, plan, and rehearse upcoming operations.

Throughout the vignette process, the team saw little to no deviation from the standardized MDMP doctrinal steps. Additionally, the system of interest would easily be adopted as an intelligence and rehearsal tool. Essentially, the vignette process validates the operational need of a MR system within JSOC.

(1) Vignette. Blue Sky

- *1.0 Operational Need*
  - *1.1 Warning Order Issued*
  - *1.2 Forces aligned for Mission*
  - *1.3 Resources allocated to effort*
- *2.0 Mission Analysis (MDMP).*
  - *2.1 Gather intelligence*
    - *2.1.1 Assess available rendering sources*
    - *2.1.2 Compile usable and relevant operational intelligence sources (Imagery, HUMINT, SIGINT, Open Source, Satellite)*
    - *2.1.3 Identify rendering source gaps*
    - *2.1.4 Task Assets to gather intelligence to fill gaps*
  - *2.2 Analysis of Intel*
  - *2.3 Generate a Common Operating Picture (COP) – Architecture Team*
    - *2.3.1 Pull raw data into MR system*
    - *2.3.2 Create MR Framework*
      - *2.3.2.1 Spatial relationships*
      - *2.3.2.2 Define environment boundaries*
      - *2.3.2.3 Provide threat composition*
    - *2.3.3 Refine MR environment*
      - *2.3.3.1 Application of Color and Texture*
      - *2.3.3.2 Inclusion of lighting*
      - *2.3.3.3 Application of environmental data*
      - *2.3.3.4 Application of threat disposition*
  - *2.4 Planner Assessment of Rendering/MR Environment*
    - *2.4.1 Commander Verifies Area Operations and Interest*
    - *2.4.2 Detail present to meet mission – granularity, usable product*
    - *2.4.3 MR environment/framework is accepted by Commander as suitable for additional operational planning and rehearsal (Reference to IPB)*
      - *2.4.3.1 Defining operational environment*
      - *2.4.3.2 Areas of interest*
      - *2.4.3.3 Characteristics of the environment*
- *3.0 Mission Planning*
  - *3.1 COA generation*
    - *3.1.1 Planners create unique scenarios to support planning process*
    - *3.1.2 Planners incorporate enemy COAs*
    - *3.1.3 Planners incorporate staff estimates*

*3.2 COA Analysis and Comparison*

*3.2.1 Planners run scenarios within multiple COAs*

*3.2.2 Planners record lessons learned*

*3.2.3 Replay scenarios for the Commander*

*3.2.4 Planners highlight force movements via friendly and enemy graphics*

*3.2.5 Commander provides input into the scenarios*

*3.2.6 Planners incorporate guidance into scenario*

*3.2.7 Planners replay updated scenario for Commander*

*3.2.8 Planners input graphical control measures and resources*

*3.3 COA Approval*

*3.3.1 Recommended plan packaged for transmission to approval authority*

*3.3.2 COA Approved*

*3.3.3 Planners generate key tasks and generate synchronization matrix*

*3.3.4 Overlay operational graphics*

*3.3.5 Operational Plan is finalized*

*3.3.6 Sharable graphics/imagery created to support OPORD*

*4.0 Rehearsals*

*4.1 Comms team orchestrates communication pipeline*

*4.2 Comms team establishes virtual environment*

*4.3 Comms team organizes player/operators to task and purpose/MOS*

*4.4 Disjointed operators perform mission*

*4.5 Receive a briefing*

*4.6 Talk about the plan*

*4.7 Do dry runs team level*

*4.8 Dry runs at a platoon level*

*5.0 Mission Rehearsals*

*6.0 FRAGO*

*7.0 Continuous subordinate rehearsal*

*8.0 SIGINT Updates (ongoing)*

*9.0 Mission initiation*

### c. Operational Illustration

An English adage wisely teaches that 'A picture is worth a thousand words.' Similar to scenarios and vignettes, an operational illustration helps an audience define the framework, provide context, and identify users and components of the MR gaming engine. The research team created Figure 9 MR Mission Planning Tool Operational Illustration to enhance understanding the role the gaming engine played in the larger system. From this illustration, the team captured how raw data fed into the gaming engine, the outputs required of the gaming engine, and the points at which system users interacted with the gaming engine. This operational illustration also provided the stakeholder with a visual depiction of the system in its operational environment in a simple, easily understood graphic.

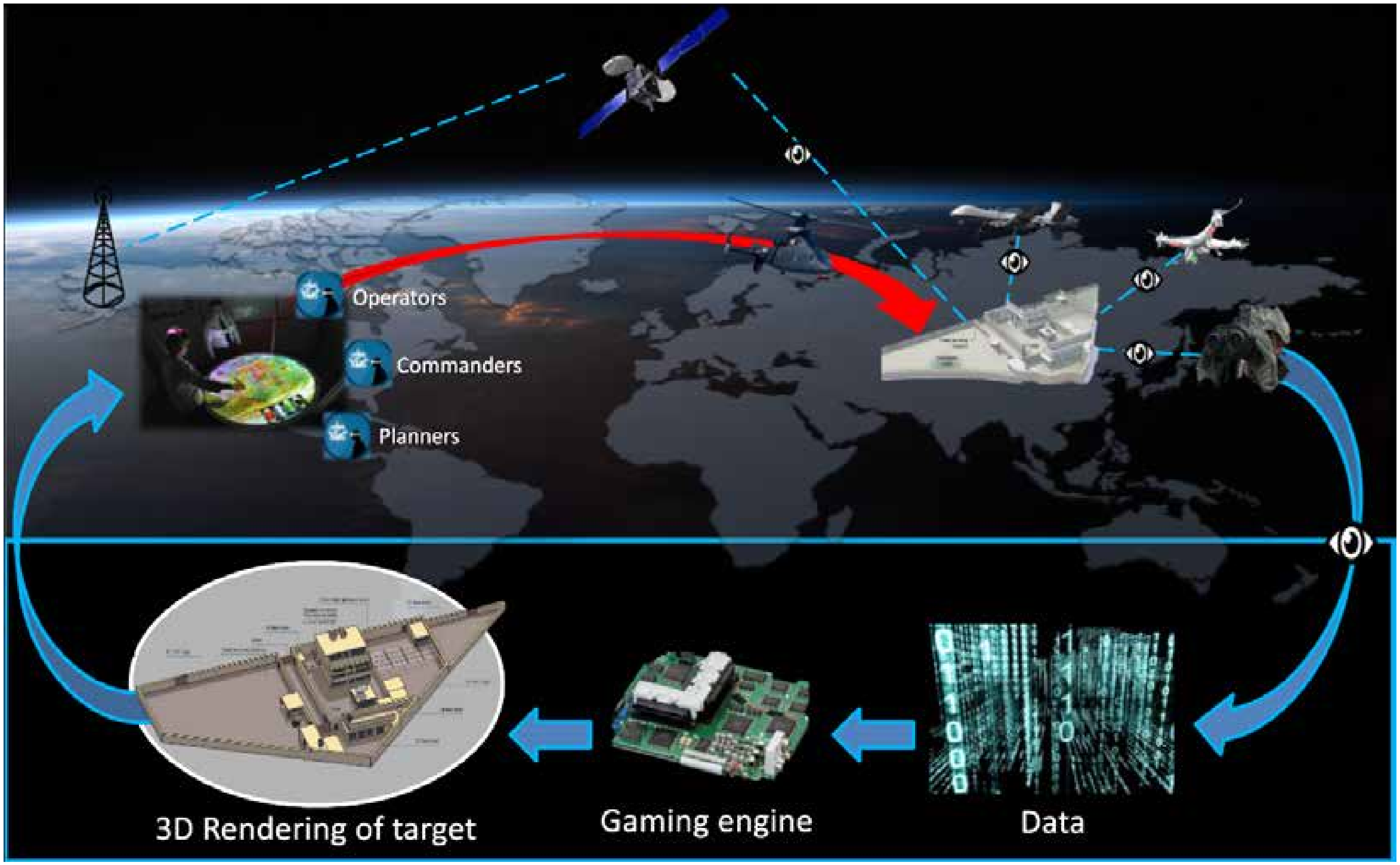



Figure 9. MR Mission Planning tool Operational Illustration

## C. REQUIREMENTS DEFINITION

After conducting an operational analysis, the research team used stakeholder needs and knowledge on gaming engines obtained from the literature review to conduct a functional analysis of the gaming engine's core functions. A functional analysis supported the next step in the System's Engineering "V" process, High-Level Requirements Definition, allowing the research team to develop system requirements that trace back to the stakeholder's needs and the functions inherent to gaming engines.

### 1. Functional Analysis

Based on the Operational Analysis and prevailing literature on gaming engines, the research team developed a functional hierarchy which identifies a gaming engine's core functions within the stakeholder's intended use. Figure 10 Gaming Engine Functional Hierarchy Displayed in SysML Block Definition Diagram displays the Functional Hierarchy in SysML Block Definition Diagram (BDD) format.

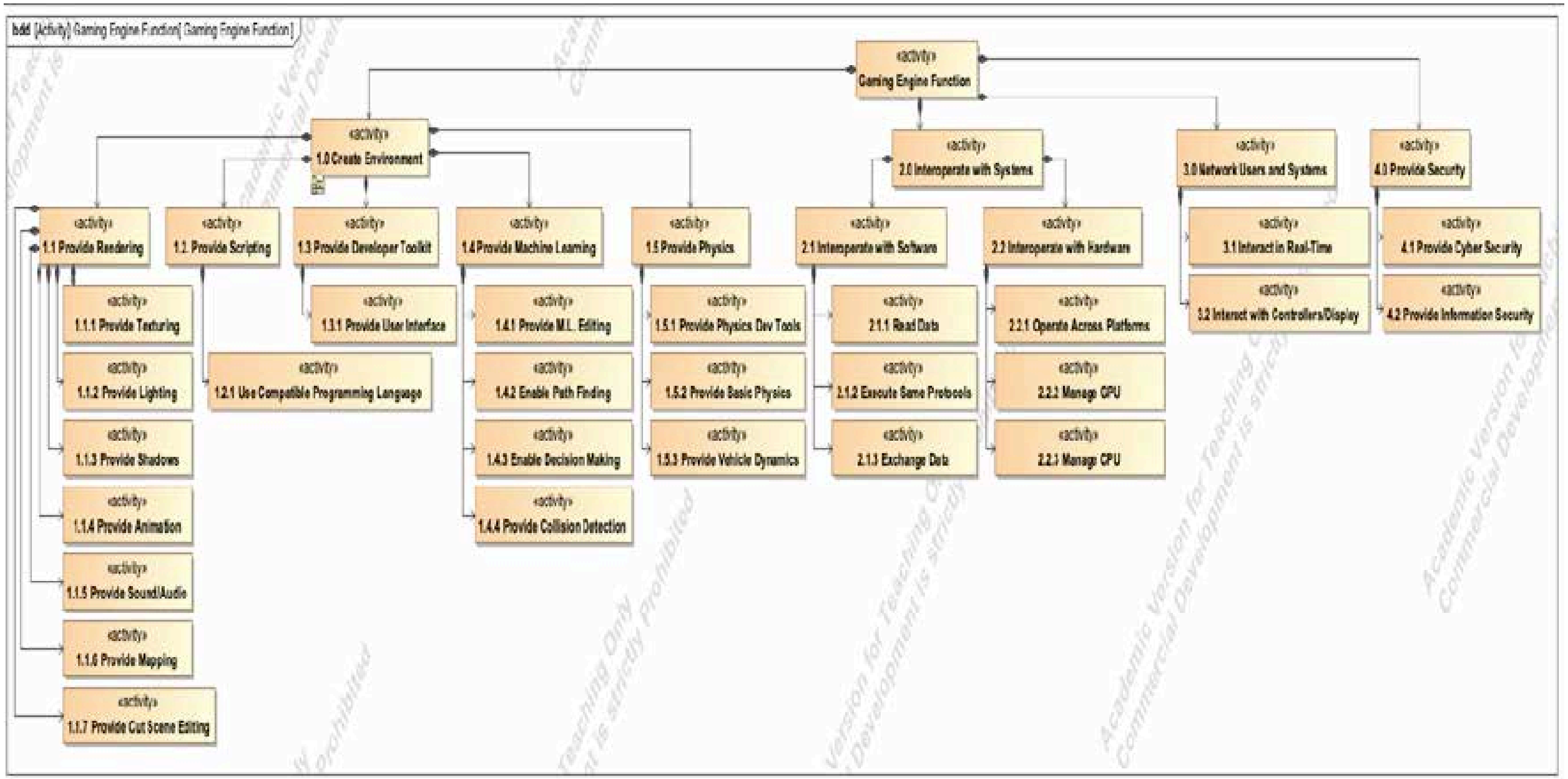



Figure 10. Gaming Engine Functional Hierarchy Displayed in SysML Block Definition Diagram

The gaming engine has four top-level core functions: create environment, interoperate with systems, network users and systems, and provide security. The research team then decomposed the system's functions to provide a better understanding of the gaming engine and generate system requirements.

### 2. Requirements Generation

Using the functional hierarchy above as the primary basis, the research team began generating and defining the gaming engine's functional requirements. These gaming engine requirements are defined and categorized in the following sections.

#### *a. SR-1.0 Create Environment:*

As the gaming engine must provide a developer toolkit to create a user-interface and virtual environment, it will create a fully immersive, realistic 3-D mixed reality environment. The created environment must include rendered 2-D and 3-D images and animations, machine learning capability, and scripting. This environment must also include physics capability that mimics the real world, including vehicle and human dynamics, that are adjustable based on the scenario. Tables 3 and 4 display the subsequent functional requirements associated with SR-1.0 (Create Environment).

Table 3. Create Environment

| | |
|---|---|
| SR-1.1 Provide Rendering | The engine must render the environment at 2K resolution and a frames per second (FPS) of 30 or greater. The gaming engine will take stitched data of a real-world location and render virtual objects and a realistic, immersive 3-D virtual environment to match the user's perspective of the real world. |
| SR-1.1.1 Provide Texturing | The gaming engine will provide basic, multi-textural, and procedural texture rendering (Xinogalos 2017, 26). The gaming engine must add images and textures to the surface of objects to create a realistic environment that displays objects realistically. |
| SR-1.1.2 Provide Lighting | The gaming engine's lighting feature must enable the creation of four lighting types: ambient, directional, omni, and spotlight (Xinogalos 2017, 26). The lighting feature is instrumental in replicating the refraction of light on other surfaces and replicating the use of night vision goggles. |
| SR-1.1.3 Provide Shadows | The gaming engine shall provide the ability to render shadows within the environment, generating a realistic sense of depth, location, and motion for the user. |
| SR-1.1.4 Provide Animation | The gaming engine must animate objects, users, and the environment into the 3-D rendered world. Animation includes forward and inverse kinematics, keyframe, skeletal, and morphing animation, and blending capabilities (Xinogalos 2017, 26). |
| SR-1.1.5 Provide Sound/Audio | The gaming engine must create spatial audio reproduction that allows the user to identify the sound's origin, distance, and direction as one would in the real world. |
| SR-1.1.6 Provide Mapping | The gaming engine will provide developers the ability to edit the environment and enhance user experience using a map editor. The map editor must allow the developer to use bump mapping to simulate imperfections on surfaces, parallax mapping to provide the illusion of depth within the virtual environment, and normal mapping. |
| SR-1.1.7 Provide Cut Scene Editing | The gaming engine will contain a cut scene editor that developers can use to introduce various elements of a military scenario into the virtual environment. |
| SR-1.2 Provide Scripting | The gaming engine must provide the ability for scenario developers to create high-level logic that replicates any operational scenario the user may experience in real life. The gaming engine must allow developers to script entire scenarios into the virtual environment and introduce different objects and variables into the environment. |
| SR-1.2.1 Compatible Language | The preferred programming language is C, C++, or Java. Other programming languages (such as C#) may be accepted, but must be translatable into C, C++, or Java. |

Table 4. Create Environment (Continued)

| | |
|---|---|
| SR-1.3 Provide Developer Toolkit | The gaming engine shall contain a developer-friendly toolkit. The kit must be suitable for both 2D and 3D graphics. |
| SR-1.3.1 Provide User Interface | The developer will have design freedom to create a user interface (UI) for the user that meets the user's needs. |
| SR-1.4 Provide Machine Learning | The game engine must be capable of generating avatars that can interact in the developed scenario, learn, and improve their interactions with that scenario. |
| SR-1.4.1 Machine Learning Editor | The gaming engine must have an Integrated Development Environment (IDE) that provides the ability to edit the machine learning to fit into specific situatio |
| SR-1.4.2 Path Finding | The gaming engine must incorporate path finding algorithms into its software. |
| SR-1.4.3 Decision Making | The game engine must use a machine learning algorithm that allows nonplayable characters (NPC's) to simulate believable behavior and decision making. |
| SR-1.4.4 Collision Detection | The gaming engine must have the ability to conduct collision detection. Collision detection concerns the detection of collisions between objects in the VR environment. This allows characters to interact with environments, terrain, objects, and each other realistically. |
| SR-1.5 Physics | The gaming engine will replicate real-world physics into a virtual environment. The gaming engine will set parameters for human physical motor limitations, weather affects, and other considerations such as: projectiles, air resistance, gravity, etc. |
| SR-1.5.1 Physics Development Tools | The gaming engine will incorporate a physics toolkit that allows developers to manipulate in-game physics. |
| SR-1.5.2 Basic Physics | The gaming engine must incorporate gravity, Newton's Laws, and other physics concepts into the virtual reality. |
| SR-1.5.3 Vehicle Dynamics | The gaming engine must be able to simulate the physics of different objects such as: aircraft, ground vehicles, Unmanned Aircraft System(UAS), etc. The gaming engine will support the variables of these objects and their affects in and on the MR system. |

#### *b. SR-2.0 Interoperate With Systems*

The gaming engine shall be interoperable with relevant DOD systems, data feeds, and hardware. The cooperation of the gaming engine and all ancillary software and hardware must facilitate the user in the creation of an MR environment. Table 5 displays the subsequent functional requirements for SR-2.0 (Interoperate with Systems).

Table 5. Interoperate with Systems Subordinate Functional Requirements

| | |
|---|---|
| SR-2.1 Interoperate with Software | The gaming engine must operate within the DOD environment and interact with existing DOD software. |
| SR-2.1.1 Read Data | The gaming engine must be able to interpret the stitched data inputs and formulate visual output. |
| SR-2.1.2 Execute Protocols | The gaming engine must be able to function with the other relevant DOD system software components. |
| SR-2.1.3 Exchange Data | The gaming engine must be able to exchange data as outputs to other DOD systems. |
| SR-2.2 Interoperate with Hardware | The gaming engine shall operate on the MR system and network. |
| SR-2.2.1 Operate Across Platforms | The system must be compatible of cross-platform play between Android and Windows based operating systems. |
| SR-2.2.2 Manage GPU | The gaming engine must be able to access and manage the GPU hardware capabilities. |
| SR-2.2.3 Manage CPU | The gaming engine must be able to access and manage the CPU hardware capabilities. |

#### *c. SR-3 Network users and systems:*

The gaming engine must exchange data between users within the virtual environment. Table 6 displays the subsequent functional requirements for SR-3.0 (Network users and systems).

Table 6. Network Users and Systems Subordinate Functional Requirements

| | |
|---|---|
| SR-3.1 Interact in Real-Time | Gaming engine must have a network module that enables real-time interaction between multiple users. The network shall initially support 12 users and be scalable to an increase in users. The network module shall support multiple geographically separated users, both individuals and groups, across multiple continents. |
| SR-3.2 Interact with Controllers/Display | The gaming engine must accept inputs and send outputs to external controller devices, displays, and hardware. |

#### d. *SR-4 Provide Security:*

The gaming engine developer and related software must comply with information and cyber security regulations outlined in DOD Instruction (DoDI) 8310.01, dated 2015. Table 7 displays the subsequent functional requirements for SR-4.0 (Provide Security).

Table 7. Provide Security

| | |
|---|---|
| SR-4.1 Provide Cyber Security | The gaming engine must adhere to DOD cyber security related policies. |
| SR-4.2 Provide Information Security | The engine shall prevent unauthorized use, authenticate all users and information, and log/record all developer input activity. |

### 3. Stakeholder Needs Traceability

The research team used model-based systems engineering (MBSE) to ensure traceability between functional requirements and stakeholder needs. This traceability matrix allowed the research team to validate functional requirements against the stakeholder’s effective needs. Tables 8 and 9 display the SysML Requirements Matrix that depicts this functional requirement to stakeholder need traceability.

Table 8. SysML Requirements Matrix

| | Stakeholder Needs | SN-1. Security | SN-1.1 Closed-Loop Network | SN-1.2 Manufacturer/Developer Origins | SN-2. Interoperability | SN-2.1 Read/accept data inputs | SN-2.2 Sensor Fusion | SN-2.3 Android Compatibility | SN-2.4 Multi-platform support | SN-2.5 JSOC Network Compatibility | SN-2.6 Bluetooth Compatibility | SN-3. Gaming Engine Performance | SN-3.1 Frame Rendering | SN-3.2 CPU Usage | SN-3.3 GPU Usage | SN-3.4 Network Usage | SN-3.5 Machine Learning | SN-3.6 Response Time | SN-3.7 Hard-wired CPU into Network | SN-3.8 Virtual Reality |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **System Requirements** | | **1** | **5** | **2** | **1** | **6** | **3** | **5** | **5** | **3** | **3** | **1** | **10** | **2** | **11** | **2** | **6** | **3** | **6** | **11** |
| SR-1 Create Environment | **7** | | | | | | | | | | | X | X | | X | | X | X | X | X |
| SR-1.1 Provide Rendering | **3** | | | | | | | | | | | | X | | X | | | | | X |
| SR-1.1.1 Provide Texturing | **2** | | | | | | | | | | | | X | | X | | | | | |
| SR-1.1.2 Provide Lighting | **2** | | | | | | | | | | | | X | | X | | | | | |
| SR-1.1.3 Provide Shadows | **2** | | | | | | | | | | | | X | | X | | | | | |
| SR-1.1.4 Provide Animation | **2** | | | | | | | | | | | | X | | X | | | | | |
| SR-1.1.5 Provide Sound/Audio | **2** | | | | | | | | | | | | X | | X | | | | | |
| SR-1.1.6 Provide Mapping | **2** | | | | | | | | | | | | X | | X | | | | | |
| SR-1.1.7 Provide Cut Scene Editor | **2** | | | | | | | | | | | | X | | X | | | | | |
| SR-1.2 Provide Scripting | **1** | | | | | | | | | | | | | | | | | | | X |
| SR-1.2.1 Use Compatible Programming Language | **3** | | | | | X | X | X | | | | | | | | | | | | |
| SR-1.3 Provide Developer Toolkit | | | | | | | | | | | | | | | | | | | | |
| SR-1.3.1 Provide User Interface | | | | | | | | | | | | | | | | | | | | |
| SR-1.4 Provide Machine Learning | **1** | | | | | | | | | | | | | | | | X | | | |
| SR-1.4.1 Provide M.L. Editing | **1** | | | | | | | | | | | | | | | | X | | | |
| SR-1.4.2 Enable Path Finding | **1** | | | | | | | | | | | | | | | | X | | | |
| SR-1.4.3 Enable Decision Making | **1** | | | | | | | | | | | | | | | | X | | | |
| SR-1.4.4 Provide Collision Detection | **1** | | | | | | | | | | | | | | | | X | | | |
| SR-1.5 Provide Physics | **1** | | | | | | | | | | | | | | | | | | | X |
| SR-1.5.1 Provide Physics Dev Tools | **1** | | | | | | | | | | | | | | | | | | | X |
| SR-1.5.2 Provide Physics | **1** | | | | | | | | | | | | | | | | | | | X |
| SR-1.5.3 Provide Vehicle Dynamics | **1** | | | | | | | | | | | | | | | | | | | X |

Table 9. SysML Requirements Matrix (Continued)

| | Stakeholder Needs | SN-1. Security | SN-1.1 Closed-Loop Network | SN-1.2 Manufacturer/Developer Origins | SN-2. Interoperability | SN-2.1 Read/accept data inputs | SN-2.2 Sensor Fusion | SN-2.3 Android Compatibility | SN-2.4 Multi-platform support | SN-2.5 JSOC Network Compatibility | SN-2.6 Bluetooth Compatibility | SN-3. Gaming Engine Performance | SN-3.1 Frame Rendering | SN-3.2 CPU Usage | SN-3.3 GPU Usage | SN-3.4 Network Usage | SN-3.5 Machine Learning | SN-3.6 Response Time | SN-3.7 Hard-wired CPU into Network | SN-3.8 Virtual Reality |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SR-2 Interoperate with Systems | 1 | | | | X | | | | | | | | | | | | | | | |
| SR-2.1 Interoperate with Software | 4 | | | | | X | X | X | X | | | | | | | | | | | |
| SR-2.1.1 Read Data | 3 | | | | | X | X | X | | | | | | | | | | | | |
| SR-2.1.2 Execute Same Protocols | 4 | | | | | X | X | X | X | | | | | | | | | | | |
| SR-2.1.3 Exchange Data | 5 | | X | | | X | | | X | | X | | | X | X | | | | X | |
| SR-2.2 Interoperate with Hardware | 6 | | | | | | | | X | | X | | | | | | | | X | X |
| SR-2.2.1 Operate across platforms | 2 | | | | | | | X | X | | | | | | | | | | | |
| SR-2.2.2 Manage GPU | 3 | | | | | | | | | | | | X | | X | | | | | X |
| SR-2.2.3 Manage CPU | 1 | | | | | | | | | | | | | X | | | | | | |
| SR-3 Network users and systems | 5 | | X | | | X | | | | X | | | | | | X | | | | X |
| SR-3.1 Interact in real-time | 4 | | | | | | | X | | | | | | | | X | | X | | X |
| SR-3.2 Interact with controllers/display | 4 | | | | | X | | | | | X | | | | | | | X | | X |
| SR-4 Provide Security | 5 | X | X | X | | | | | | X | | | | | | | | | X | |
| SR-4.1 Provide Cyber Security | 2 | | X | | | | | | | | | | | | | | | | X | |
| SR-4.2 Provide Information Security | 4 | | X | X | | | | | | X | | | | | | | | | X | |

The research team concluded from the requirements matrix that the system requirements satisfied all stakeholder needs. Most importantly, the research team identified one critical requirement not addressed by the stakeholder: Provide developer tool kit. Without the developer toolkit, operational scenarios and missions cannot be developed within the gaming engine, rendering it ineffective for the stakeholder's purposes. Therefore, the research team identified it as a critical function of a gaming engine. Due to the importance of this function, the stakeholder should re-evaluate the importance of a developer toolkit on the acquisition of a gaming engine. The usefulness of the developer toolkit can serve as a discriminator when comparing gaming engines.

## D. VALUE ANALYSIS

The value analysis section of this report contains three primary sections: 1) Value Hierarchy, 2) Value Modeling Process, and 3) Decision Matrix. The value hierarchy describes the overall system objective, measures of effectiveness (MOE) and measures of performance (MOP) with associated threshold and objectives. In the value modeling process, the research team applied the Parnell method and used additive value modeling to compare gaming engines. The primary purpose of the value model is to provide a decision matrix as a traceable, repeatable selection tool for the stakeholder. Figure 11 describes the research team's value analysis process in six steps to create a selection tool for the stakeholder.

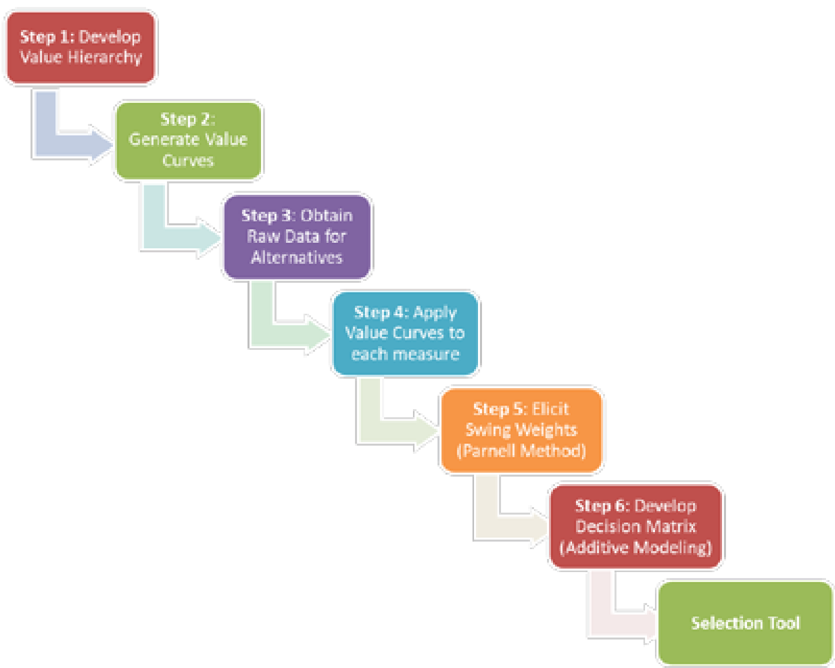



Figure 11. Value Modeling Process

### 1. Value Hierarchy

The overall system objective is to provide a digital mission planning environment. To provide value and satisfy mission requirements, the gaming engine must: 1) Increase immersive experience, 2) Support interoperability, 3) Support networkability, 4) Provide security, and 5) Provide supportability. These high-level values are the system's Measures of Effectiveness (MoE's). Figure 12 denotes all of the value characteristics identified.

The MoE's were decomposed further into measurable attributes or Measures of Performance (MOPs). This research team used MOPs as the primary basis for gaming engine comparison. Many of the MOP's identified involve the gaming engine as a critical component of the MR system as a whole and are affected by variables outside the scope of the gaming engine itself. Collecting measurement data on each MOP is outside the scope of this research project because many MOPs involve variables outside the gaming engine itself. Examples of out of scope variables are the MR system hardware and the DOD's network infrastructure. Regardless, the research team wanted to capture these criteria because each are critical considerations when acquiring an MR system for the DOD.

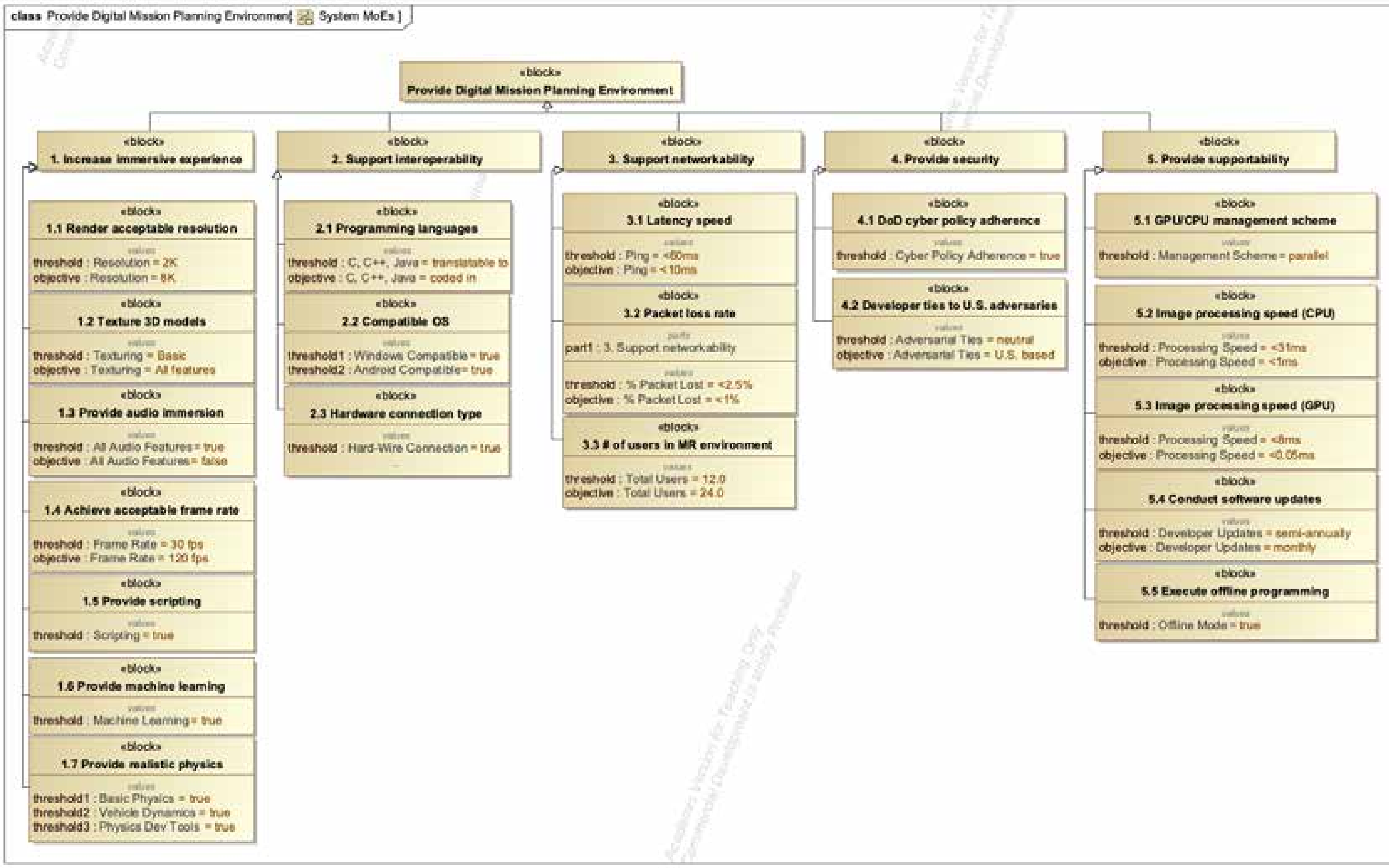



Figure 12. Classification Diagram for System MOE-Provide Digital Mission Planning Environment

### *a. MOE 1- Increase immersive experience*

The gaming engine must increase immersive experience through graphical rendering, texturing, audio immersion, scripting, machine learning, and realistic in-game physics. Table 10 identifies and describes each MOP's threshold and objective parameters.

Table 10. Measure of Effectiveness, Increase Immersive Experience

| **Measure of Effectiveness** | **Measures of Performance** | **Threshold Values** | **Objective Values** | **Description** |
|---|---|---|---|---|
| **1. Increase Immersive Experience** | 1.1. Render at acceptable resolution | 2K | 8K | The baseline resolution for VR is 2K. However, 8K is preferred to keep up with emerging technology and increased immersion. |
| | 1.2. Texture 3D models | Basic Texturing | Basic, Multi-textural, and procedural texturing features | The gaming engine must provide basic texturing. However multi-textural, and procedural texture rendering is preferred. |
| | 1.3. Provide audio immersion | Binary: Y/N | | The gaming engine must provide 3D audio capability, allowing the user to identify the source direction of a sound, and streaming audio capability since it enables user immersion. |
| | 1.4. Achieve acceptable Framerate | 30 FPS | 120 FPS | The gaming engine must perform at 30 frames per second (FPS) at a bare minimum. A higher FPS up to 120 FPS is preferred since it minimizes motion sickness. |
| | 1.5. Provide scripting | Binary: Y/N | | The game engine must provide the ability to script scenarios and introduce variables into the environment. |
| | 1.6. Provide machine learning | Binary: Y/N | | The game engine must be capable of machine learning to improve the scenario. |
| | 1.7. Provide realistic physics | Binary: Y/N | | The gaming engine must replicate real-world physics into a virtual environment through the use of discrete values. |

### *b. MOE 2- Support Interoperability*

Since the gaming engine must work within the context of the DOD’s architecture, it must be interoperable with existing hardware and software. Table 11 identifies and describes each MOP’s threshold and objective parameters.

Table 11. Measure of Effectiveness, Support Interoperability

| Measure of Effectiveness | Measures of Performance | Threshold Values | Objective Values | Description |
|---|---|---|---|---|
| **2. Support Interoperability** | 2.1 Programing languages | Translatable to C, C++, Java | Coded in C, C++, Java | Since the Android Team Awareness Kit (ATAK) is coded in C, C++, and Java, the gaming engine must be translatable to those languages. A gaming engine coded in these languages is preferred. |
| | 2.2 Compatible Operating Systems (OS) | Binary: Y/N | | The gaming engine must be compatible with Android (ATAK) and Windows (prevailing DOD operating system). |
| | Hardware connection type | Binary: Y/N | | The gaming engine must support hard-wire connection for operational security (OPSEC) reasons. |

### *c. MOE 3- Support Networkability*

The gaming engine must support the real-time interaction between users in a virtual environment across geographically dispersed areas. Latency speed and packet loss rate are critical to measuring network performance within a virtual environment, while the number of users in the MR environment addresses the operational use of the system by the stakeholder. Table 12 identifies and describes each MOP’s threshold and objective parameters.

Table 12. Measure of Effectiveness, Support Networkability

| Measure of Effectiveness | Measures of Performance | Threshold Values | Objective Values | Description |
|---|---|---|---|---|
| **3. Support Networkability** | 3.1 Latency speed | <60ms | <10ms | The gaming engine must support a network latency (ping) rate of at most 60 milliseconds (ms) in order to create an immersive experience for geographically separated users. For the most immersive experience, 10ms latency is preferred. |
| | 3.2. Packet loss tolerance | 1% | >2.5% | The gaming engine must be able to operate with at least 1.0% packet loss to create an immersive experience for geographically separated users. 2.5% packet loss tolerance is preferred. Decrease packet loss results in a smoother experience (Beigbeder et al. 2004, 21). Packet loss tolerance is a measure of resiliency and the ability of the gaming engine to handle loss of packets and continue to provide an immersive environment for the user. |
| | 3.3 Number of users in MR environment | 12 users (1 Small Mission Unit (SMU)) | 24 users (2 SMU's) | The gaming engine must support multiple users within the same environment. |

### *d. MOE 4- Support DOD Security*

Since the gaming engine is software, it must adhere to the most current DOD cyber security policies. Due to the sensitive nature of the mission information used for rehearsal within the environment, the developer's ties to adversarial nations must also be considered. Table 13 identifies and describes each of the MOP's threshold and objective parameters.

Table 13. Measure of Effectiveness, Support DOD Security

| Measure of Effectiveness | Measures of Performance | Threshold Values | Objective Values | Description |
|---|---|---|---|---|
| **4. Support DOD Security** | 4.1. DOD Cyber Policy Adherence | Binary: Y/N | [redacted] | The gaming engine must adhere to cyber security related policies and issuances created by the Deputy CIO for Cyber Security. The engine must be able to reside and operate on a closed network and function without linkage to a parent company. |
| | 4.2. Developer ties to adversaries. | Neutral Country | U.S. based company | To ensure operational and informational security, gaming engine developer must not have ties to countries adversarial to the U.S. |

### *e. MOE 5- Provide Supportability*

As with any system, supportability must be considered at the onset of gaming engine acquisition; most of the supportability considerations involve system upgradeability considerations. The CPU/GPU management scheme and image processing speeds are concerned with ensuring the selected gaming engine conforms to future technology, while software update frequency is concerned with the amount of developer support received throughout the life-cycle of the gaming engine. Table 14 denotes each MOP's threshold and objective parameters.

Table 14. Measure of Effectiveness, Provide Supportability

| Measure of Effectiveness | Measures of Performance | Threshold Values | Objective Values | Description |
|---|---|---|---|---|
| 5. Provide Supportability | 5.1 GPU and CPU management scheme | CPU/GPU parallel | | The gaming engine must perform using traditional CPU and GPU management schemes. However, using CPU/GPU more effective management scheme is preferred. |
| | 5.2 Image processing speed-CPU | 31 ms | 1 ms | The gaming image processing algorithms must enable speed at or below 31ms. (Baek et al., 2013) |
| | 5.3 Image processing speed-GPU | 8 ms | .05 ms | The gaming image processing algorithms must enable speeds at or below 8 ms. (Baek et al., 2013) |
| | 5.4. Conduct software updates | Developers push updates semi-annually. | Developers push updates monthly. | Gaming engine developer team must proactively conduct software updates that add features, fix bugs, and otherwise refine the gaming engine as technology improves. Gaming engine should be updated with technology on par with prevailing MR technology in the gaming industry. |
| | 5.5 Executable offline programming | Binary: Y/N | | System developers must be able to create and manipulate environment while offline. |

Each binary MOP is considered screening criteria for gaming engines. In other words, the gaming engine must have these attributes to be considered for selection. The research team recognizes that most of the binary criteria could be decomposed further with research; however, the primary stakeholder views these criteria as binary. Therefore, the research team applied the remaining 11 nonbinary MOPs to the value modeling process described in the next section.

### 2. Value Modeling Process

The research team applied the Parnell Method (Parnell and Trainor, 2009) to the 11 MOPs identified as nonbinary in the value hierarchy. First, the research team generated value curves for each MOP. Second, the research team generated data on the two leading gaming engines in industry (Unity and Unreal) and translated this data to value scores using each respective MOP value curve. Third, the research team elicited swing weights using stakeholder preferences and variation between each evaluation measure's threshold and objective parameters. Finally, the research team applied the swing weights and values using an additive model to determine which gaming engine performed best. The equation for the additive model is shown in Figure 13 MR Gaming Engine Value Model.

**MR Gaming Engine Value Model**

**Total Value of a Gaming Engine =** $v(\mathbf{x}) = \sum_{i=1}^{n} w_i v_i(x_i)$

Figure 13. MR Gaming Engine Value Model. Adapted from Parnell and Trainor (2009).

This additive model is captured in a decision matrix. This decision matrix is ultimately the comparison tool the research team will provide to JSOC to support future decision making. Although some of the data used was notional, this process may be used as a framework when more data on different gaming engines can be attained.

#### *a. Generated Value*

The research team assigned values to each of the 11 MOPs. The creation of value curves assigned a relative value to raw data obtained through open source documents. In cases where the data was ordinal, the research team created classifications to transform this ordinal data into nominal data. Figure 14 displays the value curve for achieving acceptable frame rate as an example. This value curve shows that a frame rate of 90 Frames Per Second

(FPS) provides a value of 8 to the stakeholder. The complete list of MOP value curves are found in Appendix B.

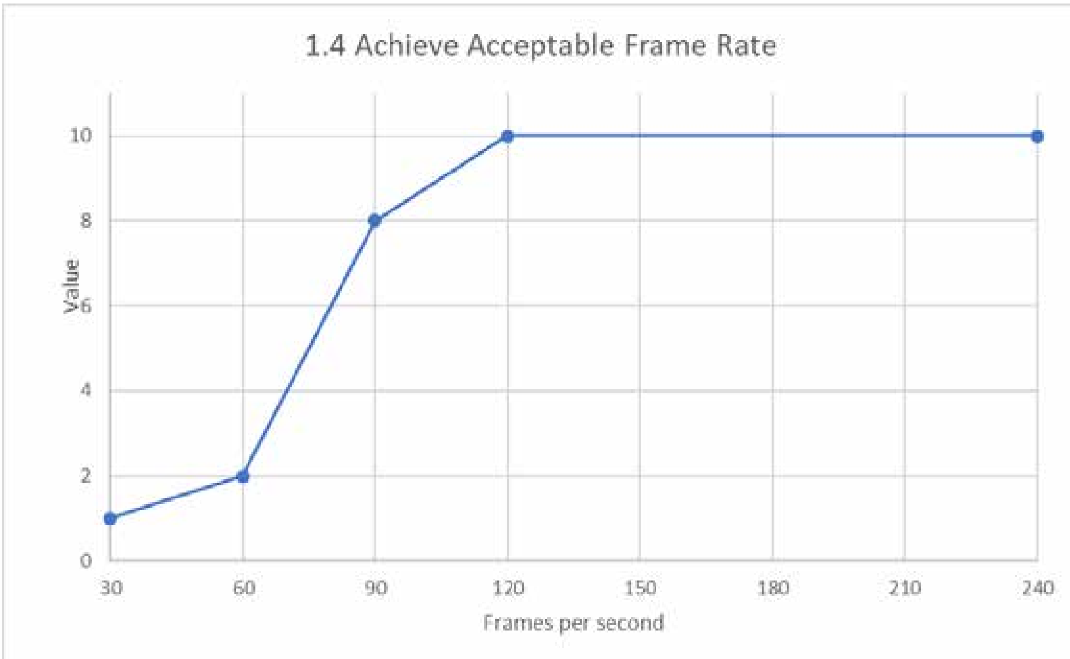



Figure 14. Measure of Performance Value Curve for 1.4 Achieve Acceptable Frame Rate

With values appropriately defined, the research team sought to collect data on market gaming engines for comparison. A literature search listed Unity and Unreal as the current top gaming engines on the market. The research team conducted research into the performance of both these gaming engines and determined that each meets the basic criteria necessary to meet JSOC's needs. Table 15 displays the raw data ($\mathcal{X}_i$) collected for each gaming engine.

Table 15. Raw Data Collected for Gaming Engines ($\mathcal{X}_i$)

| Evaluation Measure | Objective | Unity | Unreal |
|---|---|---|---|
| 1.1. Render at acceptable resolution | max | 8K | 8K |
| 1.2. Texture 3D models | max | 3 | 3 |
| 1.4. Achieve acceptable Framerate | max | 50* | 60* |
| 2.1 Programing languages | min | Translatable | Coded in |
| 3.1 Latency speed | min | 1ms** | 1ms** |
| 3.2. Packet loss tolerance | min | 2.5%** | 2.5%** |
| 3.3 Number of users in MR environment | max | 24** | 24** |
| 4.2. Developer ties to adversaries. | min | U.S. (San Francisco, CA) | U.S. (Cary, NC) |
| 5.2 Image processing speed-CPU | min | 1ms** | 1ms** |
| 5.3 Image processing speed-GPU | min | 0.05ms** | 0.05ms** |
| 5.4. Conduct software updates | min | Monthly (Unity3D 2020) | Semi-annual (SnapFiles 2018) |

* Generated using "Fastest" setting for Unity 5 and "Low" setting for Unreal Engine 4 when player is "idle" (Weebly n.d.)

** Notionally generated measures that depend on variables outside of gaming engine itself

The research team applied the value curves to the raw data, producing a value score for each attribute of the gaming engine. "Ten" represents the highest value and "one" represents the lowest value. This application allowed the research team to transform different data sets and units of measure into a common value scale. Table 16 displays the values ($\mathcal{V}_i$ ($\mathcal{X}_i$)) for each gaming engine.

Table 16. Applied Value Curves ($\mathcal{V}_i(\mathcal{X}_i)$)

| Evaluation Measure | Objective | Unity | Unreal |
|---|---|---|---|
| 1.1. Render at acceptable resolution | max | 10 | 10 |
| 1.2. Texture 3D models | max | 10 | 10 |
| 1.4. Achieve acceptable Framerate | max | 1.8 | 2 |
| 2.1 Programing languages | min | 4 | 10 |
| 3.1 Latency speed | min | 10 | 10 |
| 3.2. Packet loss tolerance | min | 10 | 10 |
| 3.3 Number of users in MR environment | max | 10 | 10 |
| 4.2. Developer ties to adversaries. | min | 10 | 10 |
| 5.2 Image processing speed-CPU | min | 10 | 10 |
| 5.3 Image processing speed-GPU | min | 10 | 10 |
| 5.4. Conduct software updates | min | 10 | 1 |

#### *b. Weighted Values*

Swing weight incorporates level of variance and importance of each evaluation measure to provide a ranked weighting from most important to least important. Weighing evaluation measures is necessary to differentiate each by overall significance. To obtain weighted values or swing weights, the research team needed to determine the variation in measure ranges and level of importance to the stakeholder. Initially, the research team calculated the variance in measure ranges between each MOP's threshold and objective parameters from the Figure 12 Value Hierarchy using the equation

$$\text{Variance in Range} = \frac{\text{max value} - \text{min value}}{\text{Average value}}.$$

The research team classified variation in range for each MOP into three categories: Low (0-99%), Medium (100-149%), and High (150% or more). Classifying variance in range into these three categories along with level of importance discussed below helps elicit swing weights between the different MOP's. Table 17 depicts the variation in range for each MOP.

Table 17. Variation in Range

| Evaluation Measure | Objective | Variation in Range | Classification |
|---|---|---|---|
| 1.1. Render at acceptable resolution | max | 120% | MED |
| 1.2. Texture 3D models | max | 100% | MED |
| 1.4. Achieve acceptable Framerate | max | 120% | MED |
| 2.1 Programing languages | min | 67% | LOW |
| 3.1 Latency speed | min | 143% | MED |
| 3.2. Packet loss tolerance | min | 86% | LOW |
| 3.3 Number of users in MR environment | max | 67% | LOW |
| 4.2. Developer ties to adversaries. | min | 100% | MED |
| 5.2 Image processing speed-CPU | min | 188% | HIGH |
| 5.3 Image processing speed-GPU | min | 198% | HIGH |
| 5.4. Conduct software updates | min | 143% | MED |

The research team identified level of importance for each MOP by using a Stakeholder Feedback Rubric. JSOC's MR Mission Planning Development Team provided the level of importance (10 being most important and 1 being least important) for each of the different attributes. Table 18 displays the Stakeholder Feedback Rubric with JSOC's level of importance for each MOP.

Table 18. Stakeholder Feedback Rubric

| | | |
|---|---|---|
| Please rank the following categories from 1 to 10. 1 is least important and 10 is most important. Cells can contain the same values. | | |
| Increase Immersive Experience | | |
| 1.1 | Render Acceptable Resolution | 10 |
| 1.2 | Texture 3D Models | 7 |
| 1.3 | Provide Audio Immersion | 5 |
| 1.4 | Achieve Acceptable Frame Rate | 7 |
| 1.5 | Provide Scripting | 5 |
| 1.6 | Provide Machine Learning | 5 |
| 1.7 | Provide Realistic Physics | 9 |
| Support Interoperability | | |
| 2.1 | Programming Languages | 10 |
| 2.2 | Compatible OS | 10 |
| 2.3 | Hardware Connection Type | 5 |
| Support Networkability | | |
| 3.1 | Latency Speed | 8 |
| 3.2 | Packet Loss Tolerance | 8 |
| 3.3 | Users in MR Environment | 8 |
| Security | | |
| 4.1 | DOD Cyber Policy Adherence | 7 |
| 4.2 | Developer Ties to U.S. Adversaries | 9 |
| Supportability | | |
| 5.1 | GPU/CPU Management Scheme | 8 |
| 5.2 | Image Processing Speed (CPU) | 8 |
| 5.3 | Image Processing Speed (GPU) | 9 |
| 5.3 | Software Update Frequency | 7 |
| 5.5 | Offline Programming/Usage Abilities | 10 |

The research team then assigned the different importance rankings provided by JSOC into level of importance categories: Low (0-3), Medium (4-7), and High (8 or higher). Table 19 depicts the level of importance for the criteria used in value modeling.

Table 19. Level of Importance

| Evaluation Measure | Objective | Importance (Stakeholder Preference) | Importance Range Category |
| --- | --- | --- | --- |
| 1.1. Render at acceptable resolution | max | 10 | HIGH |
| 1.2. Texture 3D models | max | 7 | MED |
| 1.4. Achieve acceptable Framerate | max | 7 | MED |
| 2.1 Programing languages | min | 10 | HIGH |
| 3.1 Latency speed | min | 8 | HIGH |
| 3.2. Packet loss tolerance | min | 8 | HIGH |
| 3.3 Number of users in MR environment | max | 8 | HIGH |
| 4.2. Developer ties to adversaries. | min | 5 | MED |
| 5.2 Image processing speed-CPU | min | 8 | HIGH |
| 5.3 Image processing speed-GPU | min | 9 | HIGH |
| 5.4. Conduct software updates | min | 7 | MED |

With variation in range and level of importance categorized, the research team developed a Swing Weight Matrix to calculate the swing weights for each MOP. Swing weighting is a method for assigning "value measures based on importance and variation of the scales of the value measures" (Parnell and Trainor 2009, 4). Parnell's Swing Matrix (2009) was used as a general guideline in establishing consistency rules to our selection methodology. Table 20 displays the Swing Weight Matrix with each MOP in its respective score.

Table 20. Swing Weight Matrix. Source: Parnell and Trainor, 2009.

| | | **Level of importance of value measure** | | |
|---|---|---|---|---|
| | | **High** | **Medium** | **Low** |
| **Variation in measure range** | **High** | *5.2 Image processing speed-CPU*<br>*5.3 Image processing speed-GPU*<br>**100** | *5.4 Conduct software updates*<br>**50** | **25** |
| | **Medium** | *1.1 Render acceptable resolution*<br>*3.1 Latency speed*<br>**85** | *1.2 Texture 3D models*<br>*1.4 Achieve acceptable framerate*<br>*4.2 Developer ties to adversaries*<br>*5.4 Conduct software updates*<br>**45** | **5** |
| | **Low** | *2.1 Programming languages*<br>*3.2 Packet loss rate*<br>*3.3 Number of users in MR environment*<br>**60** | **30** | **1** |

The research team is now able to elicit swing weights for each MOP using the equation $w_i = \frac{f_i}{\sum_{i=1}^{n} f_i}$ (Parnell and Trainor 2009, 8). Table 21 shows the calculated weights for each MOP.

Table 21. Swing Weights for Evaluation Measures

| Evaluation Measure | Objective | Weights |
|---|---|---|
| 1.1. Render at acceptable resolution | max | 0.12 |
| 1.2. Texture 3D models | max | 0.06 |
| 1.4. Achieve acceptable Framerate | max | 0.06 |
| 2.1 Programing languages | min | 0.08 |
| 3.1 Latency speed | min | 0.12 |
| 3.2. Packet loss tolerance | min | 0.08 |
| 3.3 Number of users in MR environment | max | 0.08 |
| 4.2. Developer ties to adversaries. | min | 0.06 |
| 5.2 Image processing speed-CPU | min | 0.14 |
| 5.3 Image processing speed-GPU | min | 0.14 |
| 5.4. Conduct software updates | min | 0.06 |

### *c. Decision Matrix*

The final step in the value modeling process involves developing a Decision Matrix that uses the MR Gaming Engine Value Model equation. This decision matrix calculates which gaming engine performs the best among the different MOP's, given the stakeholder's level of importance and variation among the gaming engines themselves. Table 22 displays the Decision Matrix used in this comparison.

Table 22. Decision Matrix

| Evaluation Measure | Objective | Weights | Unity | Unreal |
|---|---|---|---|---|
| 1.1. Render at acceptable resolution | max | 0.12 | 1.16 | 1.16 |
| 1.2. Texture 3D models | max | 0.06 | 0.62 | 0.62 |
| 1.4. Achieve acceptable Framerate | max | 0.06 | 0.11 | 0.12 |
| 2.1 Programing languages | min | 0.08 | 0.33 | 0.82 |
| 3.1 Latency speed | min | 0.12 | 1.16 | 1.16 |
| 3.2. Packet loss tolerance | min | 0.08 | 0.82 | 0.82 |
| 3.3 Number of users in MR environment | max | 0.08 | 0.82 | 0.82 |
| 4.2. Developer ties to adversaries. | min | 0.06 | 0.62 | 0.62 |
| 5.2 Image processing speed-CPU | min | 0.14 | 1.37 | 1.37 |
| 5.3 Image processing speed-GPU | min | 0.14 | 1.37 | 1.37 |
| 5.4. Conduct software updates | min | 0.06 | 0.62 | 0.06 |
| Totals | | 1.00 | 9.00 | 8.95 |

### 3. Value Model Results

Based on the level of importance of each MOP, level of variance between the different gaming engines, and the data available to the research team, Unity outperforms Unity by a slim margin. Both gaming engines perform similarly in most of the evaluation measures. However, they differ in three evaluation measures: 1.4 Achieve Acceptable Frame Rate, 2.1 Programming languages, and 5.4 Conduct Software Updates. Unreal performs slightly better given the same hardware configuration, providing a better frame rate on average compared to Unity. Unreal is also programmed in C++, making it more compatible with existing DOD systems, such as the Android Team Awareness Kit (ATAK). Since Unreal is coded in a compatible language, JSOC does not have to allocate resources, time, or money to translating code between the MR gaming engine and existing DOD software. Unity outperformed Unreal in one key aspect related to system supportability: software update frequency. Unity conducts software updates much more frequently than Unreal, which may show a correlation to developer support throughout the system's life cycle.

### 4. MR Gaming Engine Selection Methodology

The research team was able to create and validate a selection methodology that can be used in the future. The selection methodology was the result of the research team's efforts to compare gaming engines for JSOC. Figure 15 provides a graphical depiction of this selection methodology. This graphic shows a sequential approach that can be used as a reference for the steps taken in this research. Furthermore, it serves as a guideline for the selection of a gaming engine or a means to compare other DOD systems in the future.

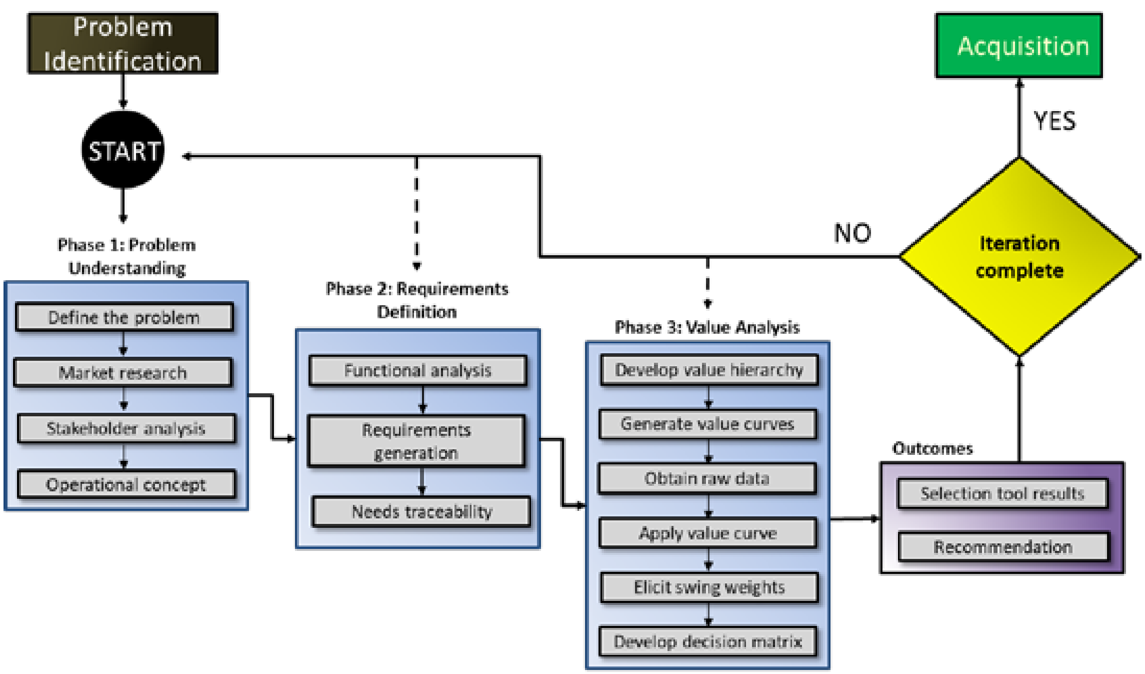



Figure 15. MR Gaming Engine Selection Methodology

This selection methodology generally follows the three phases of research discussed in Chapter III: Frameworks and Research Design: Problem Understanding, Requirements Definition, and Value Analysis. This selection methodology initiates when the acquirer has identified that there is a problem that requires the selection of alternatives. The acquirer then enters the first phase of our selection methodology: Problem Understanding. Within problem understanding, the acquirer must conduct market research to gain a foundational knowledge on the industry and system of interest. With proper baseline knowledge of the system of interest, in this case gaming engines, the acquirer can then define and bound the problem through cause-and-effect analysis. This involves gaining an understanding of the stakeholders who have an interest in the system, identifying the needs and wants of each. Then, the acquirer must conduct an operational analysis of the system in its intended environment. This involves the use of context diagrams, scenarios, vignettes, and the development of an operational illustration.

The acquirer then moves to Phase 2: Requirements Definition. The acquirer executes functional analysis to determine critical functions that the system must accomplish. System functions are translated into detailed requirements. These requirements should contain measures and criteria for success. Finally, the generated requirements are traced back to stakeholders' needs for requirements validation.

With a proper understanding of the system's functions and baseline requirements, the acquirer then conducts value analysis to determine the better alternative using multiobjective decision analysis and additive modeling that result in a selection tool. First, a value hierarchy must be developed. This value hierarchy must define the system's value to the stakeholder's problem. Each value measure must be developed with specific, measurable, attainable, realistic, and traceable MOP's to ensure a valid comparison. Second, a value curve for each value measure must be generated. The value curves normalize the different evaluation measures into a common ranking for scoring. Third, data must be collected on the performance of each alternative. This can be done through research, simulation, or testing. The fourth step involves applying the value curves to the raw data. This allows a comparison of the alternatives using a common scale for each evaluation measure. Fifth, swing weights are elicited using the Parnell Method (2009).

Swing weights incorporate variance in range and level of importance between the different evaluation measures. Sixth, a decision matrix is developed using additive modeling, incorporating both the swing weights and the value scores to determine which alternative performs better. This process generates a repeatable, traceable, selection tool that can be used in the acquisition of future DOD MR technologies.

The results of this value analysis is examined by the acquisition team to inform the acquisition decision. If the development of the system is iterative, the acquirer can circle back to whichever phase applicable. This overall process describes this research’s selection methodology used in the comparison of gaming engines.

THIS PAGE INTENTIONALLY LEFT BLANK

# V. CONCLUSIONS AND FUTURE RESEARCH

## A. CONCLUSION

The purpose of this research project was to develop a selection methodology to assess available market gaming engines which supported JSOC's selection of a MR mission planning tool. The research team has demonstrated that by, using value modeling and a system engineering approach, one can generate a repeatable, traceable selection methodology that supports the future acquisition of a MR mission planning tool. This methodology is viable because it can support changing requirements involved in emerging, state of the art technology. Furthermore, the selection methodology developed in this research has broader application for future DOD acquisitions within the MR spectrum of technologies. The systems engineering approach to selecting a suitable gaming engine for JSOC is applicable to other technical components of larger systems. The defined process offered in textbooks and other technical documents, coupled with this research's application to a system component, offers a road map for these other technical applications.

The research team addressed all of its research questions listed in Chapter I. The research demonstrated that value modeling can be applied to the selection of an MR system gaming engine which supports military mission planning. Additionally, this research identified several gaming engine characteristics that are of value to JSOC in future acquisitions of MR technologies. Lastly, this research identified several selection criteria that should be used for future JSOC MR system purchases. This report assessed two gaming engines, Unity and Unreal, that satisfy JSOC's need.

The root problem which motivated this project was an absence of a structured approach to the acquisition of a gaming engine for JSOC's MR mission planning tool, due to time constraints and operational needs. This research provides JSOC with a structured approach to shape an ill-defined problem, generate requirements, and develop an objective method to evaluate and compare systems, based on stakeholder feedback and variation between alternatives. While the comparison criteria and values may change as new technologies emerge, the overall process used in this research can still be applied.

## B. FINDINGS AND FUTURE WORK

The overall process for comparing criteria is straightforward—the math is easy. The hard part is allotting the necessary time, people, and resources toward defining the problem and producing quality requirements that both satisfy the operational need and drive the value modeling process. DOD organizations are pressured to produce solutions now versus doing dedicated research on the military applications of those solutions. Organizations must conduct analyses upfront for the value modeling process to work as intended.

Many gaming engines on the market currently satisfy JSOC's needs for an MR mission planning tool. Essentially, a gaming engine is a gaming engine, meaning the differences are marginally significant. While it is an important component of the MR system, the gaming engine should not be the sole basis for MR system comparison. Other considerations for selecting a gaming engine that should be considered include the system developers, the hardware and software security considerations, and interoperability within the DOD system architecture.

The system developer is a key component to MR mission planning and system selection consideration. The system developer touches all aspects of the system, from the software to the hardware and every interaction in between. Therefore, the system developer is a critical concern throughout the life cycle of the MR system. A thorough business case analysis should be conducted on the system developers to ensure they are not only capable of providing the best solution to the DOD currently, but are also capable of providing regular system updates and support throughout the life cycle of the MR system. The system developer can make or break the MR system. This research team's selection methodology could augment competitive prototyping as a possible solution to inform the selection of future MR systems (and their development teams) and reduce technology risk to the program office.

Further analysis should also be conducted on security considerations for MR technologies. Since MR solutions often involve Commercial-off-the-Shelf (COTS) technologies that have less stringent security requirements than the DOD, solutions should be generated as to how to adapt these technologies to the DOD's security requirements. There is also potential for developers to embed hidden code that feeds sensitive information back to adversarial nations; a concern that must be addressed upfront, especially within JSOC since they typically conduct operations with the highest security classifications.

The DOD has yet to clearly define interoperability within its own system's architecture. The initial interoperability requirements for the MR system seem simple—it must accept stitched data feeds and must be programmed using a programming language compatible with C, C++, and Java to properly interact with the ATAK software. However, expanding this technology as a robust mission planning tool requires interactions with countless legacy mission command systems, such as Joint Capabilities Release (JCR), Command Post of the Future (CPOF), and others to allow planners and commanders from various organizations to plan and rehearse within the system collaboratively. Mission command systems are just one subset of a countless amount of legacy DOD systems that the MR system must interoperate with to realize the full potential this technology could provide for future mission planning.

THIS PAGE INTENTIONALLY LEFT BLANK

# APPENDIX A. KEY TERMS AND DEFINITIONS

**Analysis of Alternatives (AoA) –** An Acquisition term modified for use in the specific application - a process applied during the early phase of system development to analyze and assess alternative solutions to a problem.

**Augmented Reality (AR) –** Live direct or indirect view of a physical, real-world environment whose elements are supplemented by computer-generated sensory input. Augmentation occurs in a real time and in semantic context with environmental elements" (Wallace and Kambouris 2017, 42).

**Bucket –** Represents the intersection of sectors (Military, Industry, Gaming) and categories (Foundational Knowledge, Evaluation, and Selection Methodologies) as depicted in Figure 4.

**Categories –** The topics of Foundational Knowledge, Evaluation, and Selection Methodologies.

**Commercial Industry –** Focuses on widespread production with the goal of selling a maximum number of products for a profit to a consumer base.

**Evaluation –** Process of defining characteristics of a subject (gaming engine) for the purposes of comparing and contrasting one or more subjects.

**Foundational Knowledge –** The historical knowledge widely accepted by the MR community regarding emerging technology.

**Gamification –** The use of game mechanics and experience design to digitally engage and motivate people to achieve their goals (Vega et al. 2017, 7).

**Gaming Engine –** A software system "responsible for the rendering, the physics, the Artificial Intelligence and other game mechanics" (Petridis et al. 2010, 29).

**Gaming industry –** A sector that was carved out of the commercial industry bucket that the research team determined would be useful for understanding and familiarizing with the most current technologies. The gaming industry is on the forefront of emerging MR technologies.

**Military industry –** Any MR usage that is tied to any sort of military application.

**Mixed reality –** The various devices, specifically displays, that encompass the reality-virtuality continuum (De Guzman 2018, 1).

**Rehearsals** – A prescribed military activity used by units and leaders at all military echelons to practice or rehearse the movements of a military operation. Rehearsals are ingrained in military culture as a necessary planning step to achieve a desired outcome. Rehearsals can range from formal to inform based on the time available to the unit or commander.

**Research team** – The authors of this literature review.

**Sand Table** – An element of planning utilized by military units and leaders for planning and rehearsal. A sand table is a scaled model representative of a desired location or objective. Sand tables allow military units to visualize an objective prior to performing actions on the objective.

**Sectors –** The fields of military, industry, and gaming.

**Selection Methodology** – A repeatable process for evaluating and selecting from a group of alternatives based on stakeholder needs and requirements.

**Stakeholder** – A DOD Acquisition specific term used to describe those persons or parties that are the recipient of benefits from an action or activity. The term can be used generically to refer to a specific person or organization or to multiple people or organization with differing interests.

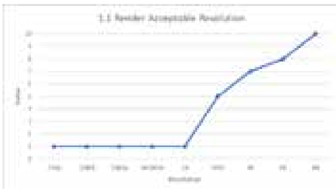
**Systems of Systems (SoS) –** A collection of independent systems, integrated into a larger system that delivers unique capabilities. The independent constituent systems collaborate to produce global behavior that they cannot produce alone.

**Warfighter –** A non-specific term that refers to members of the Armed Forces, including members of the Army, Marines, Navy, Air Force, and Space Force. The term is applied to military members who perform conventional military tasks and roles on the battlefield.

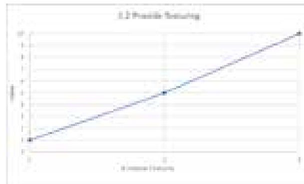
**Virtual Reality (VR) –** "A virtual rendition of reality" (Wallace and Kambouris 2017, 42)

# APPENDIX B. VALUE CURVES

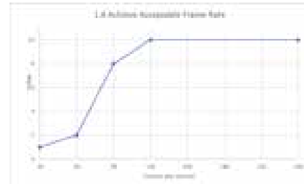
**Resolution**



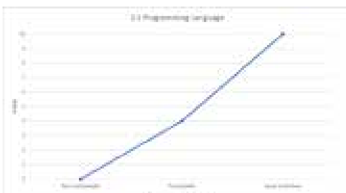
**Texturing**



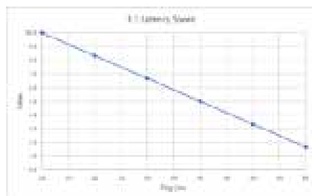
**Frame Rate**



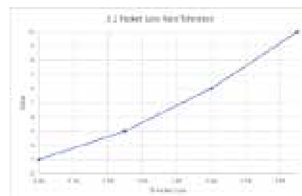
**Programming Language**



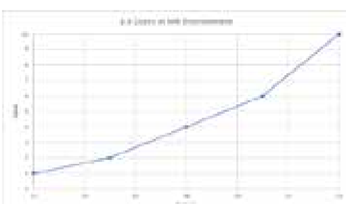
**Latency**



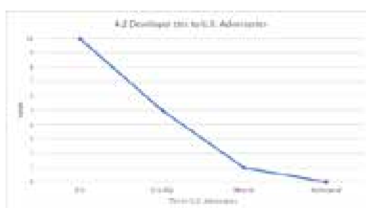
**Packet Loss Tolerance**



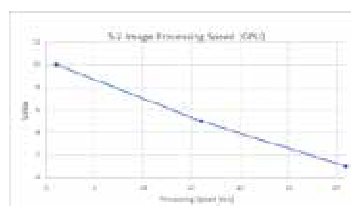
**Number of Users**



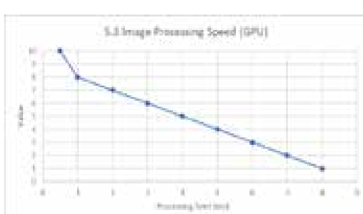
**Adversary Ties**



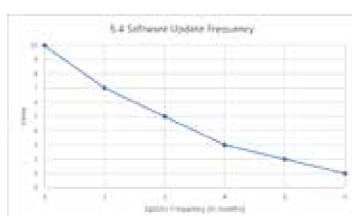
**CPU Image Processing**



**GPU Image Processing -**



**Software Update Frequency**



**RENDER ACCEPTABLE RESOLUTION**

Table 23. Render Acceptable Resolution Value Curve

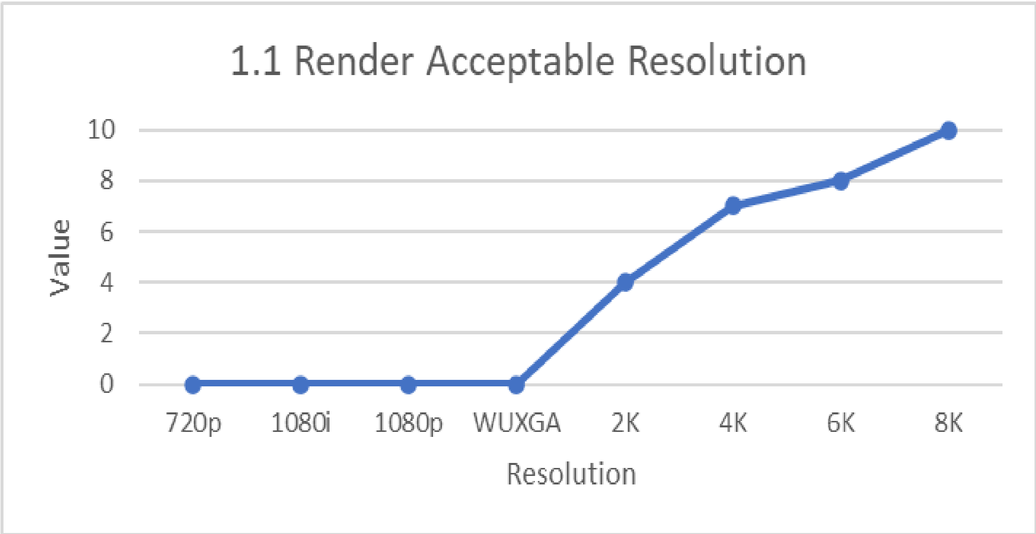



Table 24. Resolution Values

| Resolution | Value |
|---|---|
| 720p | 0 |
| 1080i | 0 |
| 1080p | 0 |
| WUXGA | 0 |
| 2K | 4 |
| 4K | 7 |
| 6K | 8 |
| 8K | 10 |

**Definition –** The number of pixels the gaming engine is capable of outputting to an optical device. Resolution is a contributor to immersion and higher resolution contributes to increased immersion. Measures are made in common terms of number of *horizontal* pixels *x vertical pixels* and are expressed in common language such as 1080p, 2K, 4K, 6K, and 8K

**Value Curve Shape –** The stakeholder perceives value through maximizing resolution. Resolutions of 720p, 1080i, 1080p, and WUXGA are acceptable to the stakeholder but are least desired and awarded a value of one. A resolution of 2K through 8K are more desirable and are awarded value according to the increased resolution. Stakeholder value is maximized at 8K resolution.

**TEXTURING**

Table 25. Provide Texturing Value Curve

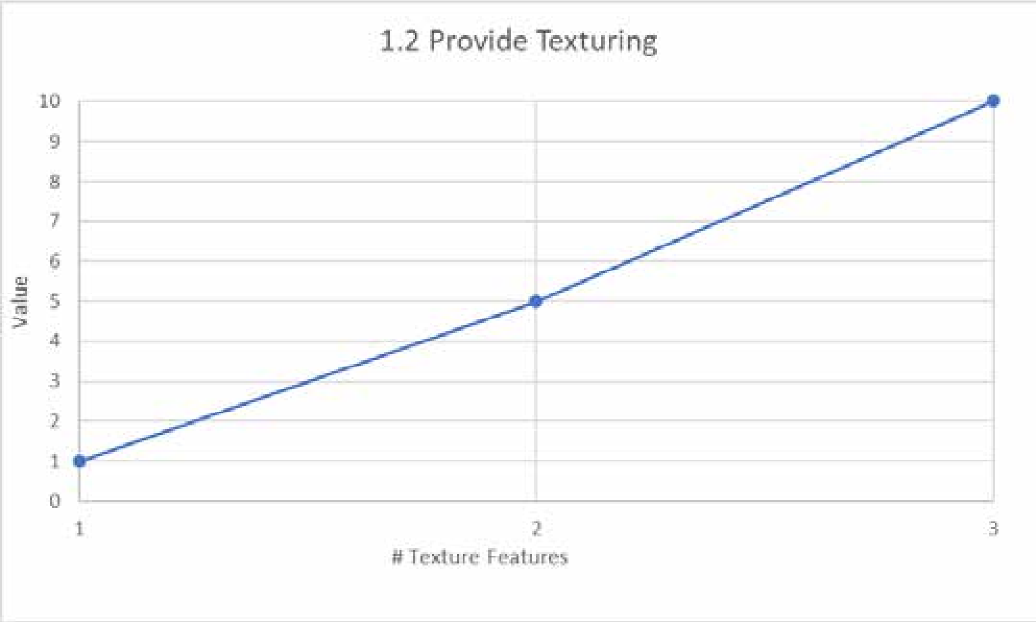



Table 26. Texturing Values

| Number of Texture Features | Value |
|---|---|
| 1 | 1 |
| 2 | 5 |
| 3 | 10 |

**Definition –** The number of features in the gaming engine that support three-dimensional texturing. This metric is quantified in three categories: Basic, Basic + Multi-textural, Basic + Multi + Procedural. More texture features increase the immersive experience and provide increased value.

**Value Curve Shape –** The stakeholder perceives value through maximizing the number of texture features. Gaming engines that provide zero texture features provide zero value and are excluded from consideration. A single texture feature receives minimal value while two and three texture features exponentially add value for the stakeholder.

**FRAME RATE**

Table 27. Achieve Acceptable Frame Rate Value Curve

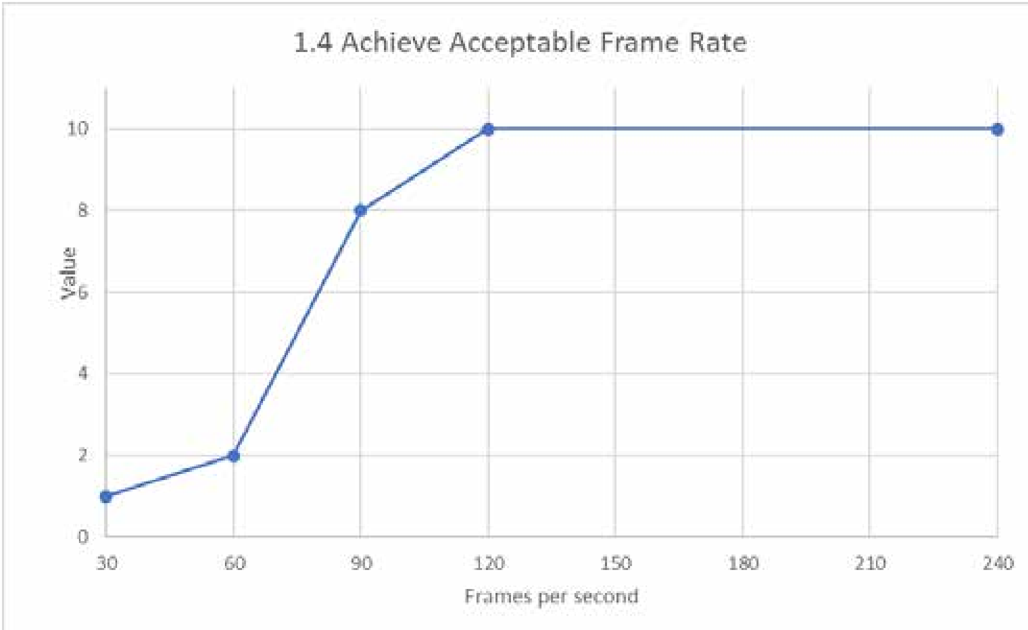



Table 28. Frame Rate Values

| Frames per second | Value |
|---|---|
| 30 | 1 |
| 60 | 2 |
| 90 | 8 |
| 120 | 10 |
| 240 | 10 |

**Definition –** The highest supported frame rate produced by the gaming engine. Frame rate contributes to a realistic virtual environment and ultimately immersion within the environment. This metric is quantified in increments of 30 frames per second. Higher frame rates increase the immersive experience and provide increased value.

**Value Curve Shape –** The stakeholder perceives value through maximizing the frame rate quantity. Gaming engines that provide less than 30 FPS provide zero value and are not considered. 30 and 60 FPS solutions offer limited value to the stakeholder. A 90 FPS solution offers significant value to the stakeholder as frame rates above 90 FPS decrease motion sickness for users within a virtual environment. Frame rates of 120 FPS and greater maximize stakeholder value.

**PROGRAMMING LANGUAGE**

Table 29. Programing Language Value Curve

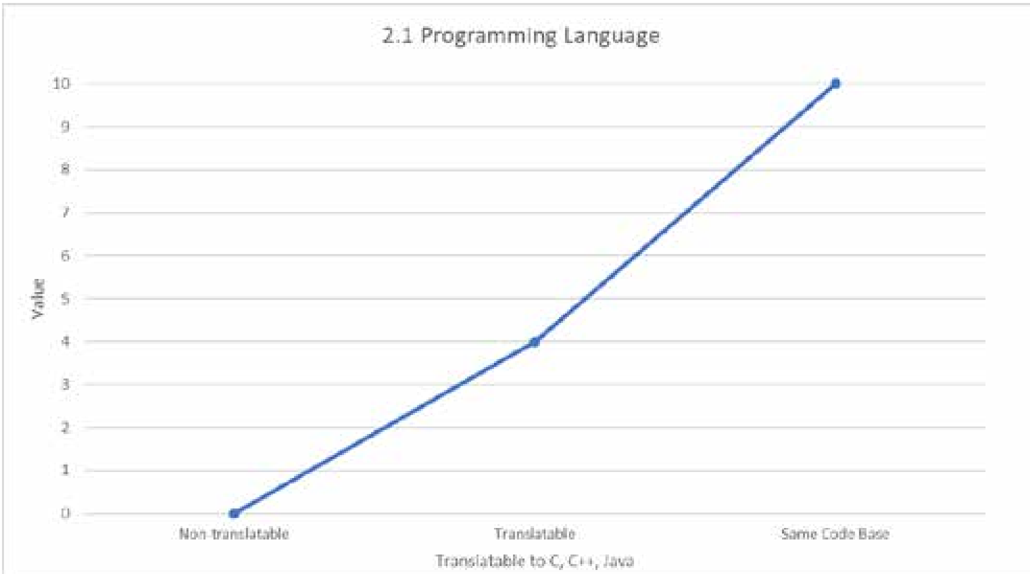



Table 30. Programing Language Values

| Programming Language | Value |
|---|---|
| Non-translatable | 0 |
| Translatable | 4 |
| Same Code Base | 10 |

**Definition** – The degree to which the gaming engine programing language supports existing project infrastructure. Programing Language is categorized in three pools: same code base, translatable to existing code through an application programming interface/translator, or non-translatable. The more easily code interacts with existing infrastructure the more valuable it is to the stakeholder.

**Value Curve Shape** – The stakeholder perceives value through minimizing the disruptions caused by incompatible code. The most preferred option is a gaming engine that shares a code base. Gaming engines with code that is not translatable provides zero value to the stakeholder and is not considered. Translatable code does have value to the stakeholder, but it is less than half as valuable as a gaming engine with the same code base as existing infrastructure. Translatable code can also be awarded, or decremented value based upon the ease of translating the code (measured as a value judgment of the stakeholder software development entity).

**LATENCY**

Table 31. Latency Speed Value Curve

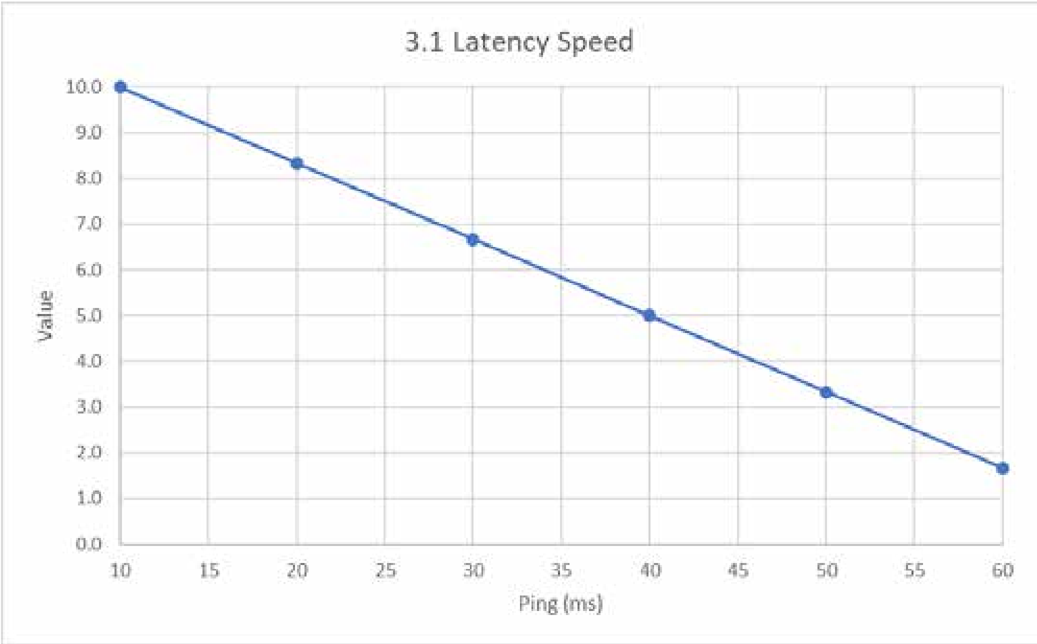



Table 32. Latency Speed Values

| Ping (in milliseconds) | Value |
|---|---|
| 60 | 1.7 |
| 50 | 3.3 |
| 40 | 5.0 |
| 30 | 6.7 |
| 20 | 8.3 |
| 10 | 10.0 |

**Definition –** The measure of network delay attributable to the gaming engine. Latency degrades the immersive experience and is undesirable. Latency is measured in tens of milliseconds (ms). Higher latency decreases value while low latency increases value.

**Value Curve Shape –** The stakeholder perceives value through minimizing latency. Stakeholder value is maximized at latency rates of 10ms and below. Gaming engine solutions that contribute to latency in excess of 60 ms are not suitable solutions. Because latency diminishes the quality of the virtual experience, each increment of degradation is worth an equal increment in value.

## PACKET LOSS TOLERANCE

Table 33. Packet Loss Tolerance Value Curve

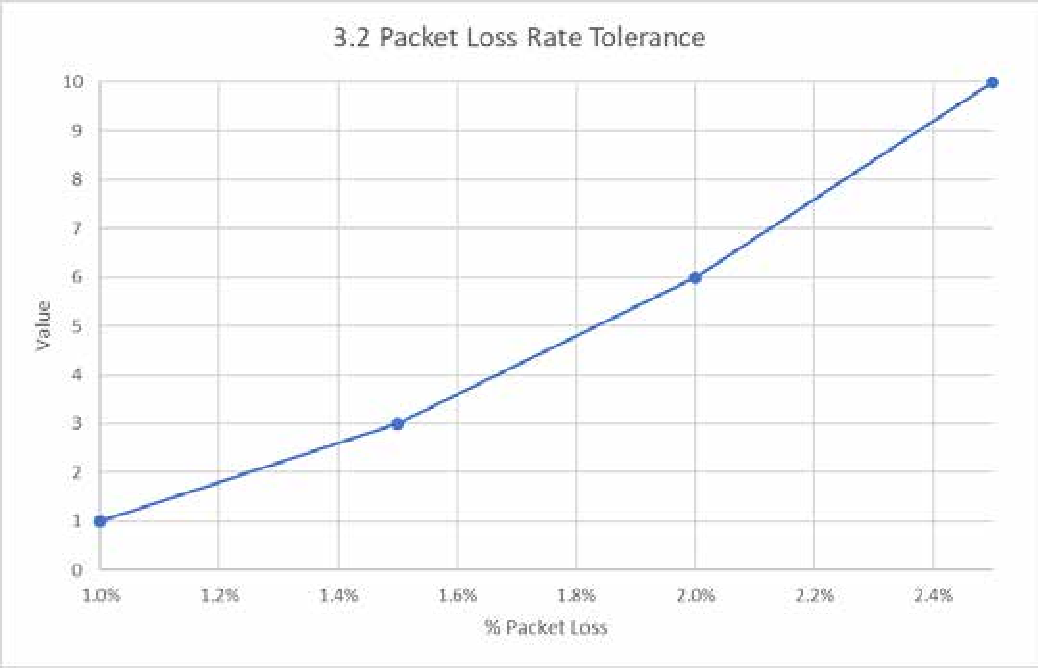



Table 34. Packet Loss Value

| Percent Packet Loss | Value |
|---|---|
| 2.5% | 10 |
| 2.0% | 6 |
| 1.5% | 3 |
| 1.0% | 1 |

**Definition –** A measure of resiliency attributable to the gaming engine's ability to manage and operate through packet loss. Packet loss degrades the immersive experience but is inevitable in geographically dispersed network. Packet loss is measured in the percent of data loss the gaming engine can tolerate before failure. Higher packet loss tolerance increases stakeholder value.

**Value Curve Shape –** The stakeholder perceives value through maximizing the packet loss a gaming engine can tolerate and still deliver an immersive experience for users. Stakeholder value is maximized at tolerance rates of 2.5% and above. Gaming engine solutions that cannot tolerate packet loss below 1% are not considered. Increases in packet loss tolerance are exponentially valuable to the stakeholder.

## NUMBER OF USERS

Table 35. Number of Users Value Curve

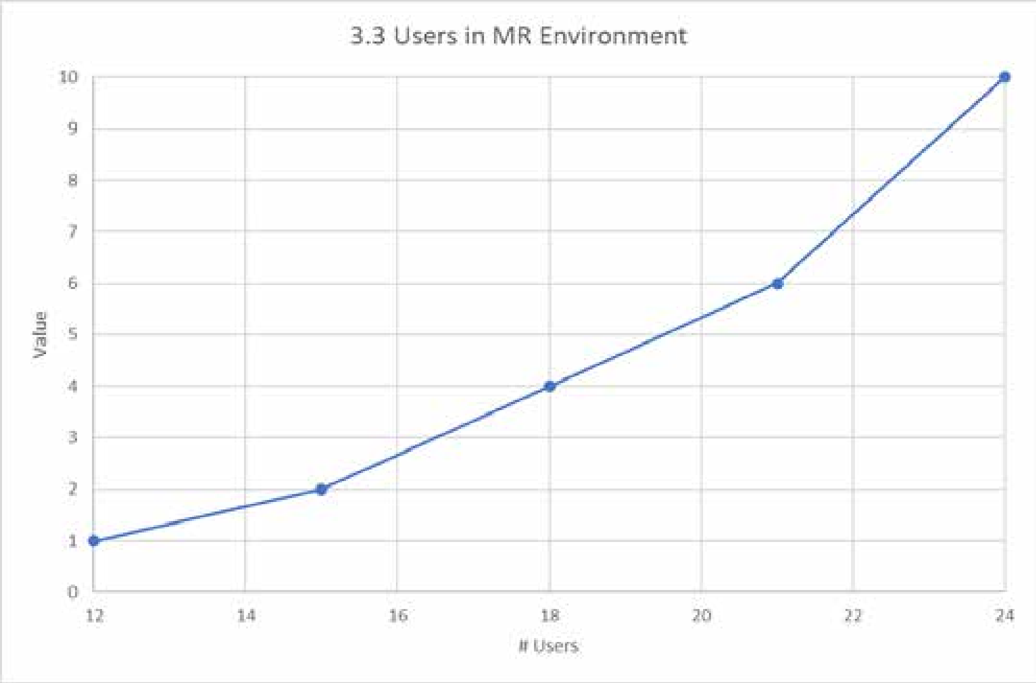



Table 36. Number of User Value

| Numbr of Users | Value |
|---|---|
| 12 | 1 |
| 15 | 2 |
| 18 | 4 |
| 21 | 6 |
| 24 | 10 |

**Definition –** The measure of the gaming engine’s ability to create a virtual environment for multiple users. Each additional user requires the gaming engine to render an additional and unique point of view. Larger numbers of users increase the realism of operating in a complex environment as well as prevent arbitrary restrictions within the virtual rehearsal environment. The ability for more users to operate within the environment increases value.

**Value Curve Shape –** The stakeholder perceives value through an increasing number of users. The stakeholder requires a system that supports a minimum of 12 users (size of one SMU team). Stakeholder value increases marginally as the gaming engine can accept additional users that perform command and support functions. Stakeholder value is maximized at 24 users (size of two SMU teams).

## DEVELOPER TIES TO U.S. ADVERSARIES

Table 37. Developer Ties to U.S. Adversaries Value Curve

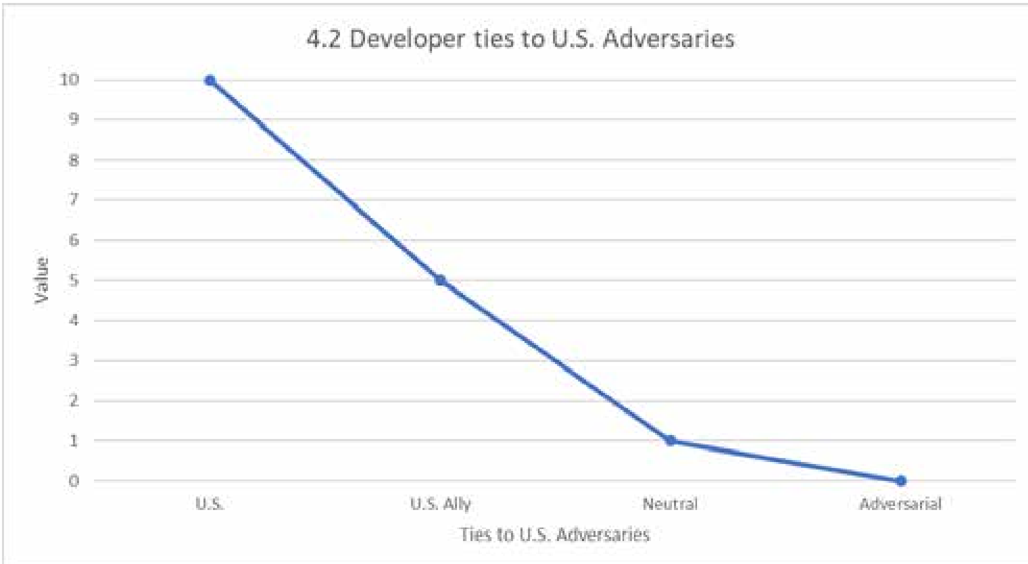



Table 38. U.S. Adversary Values

| Ties to U.S. Adversaries | Value |
|---|---|
| U.S. | 10 |
| U.S. Ally | 5 |
| Neutral | 1 |
| Adversarial | 0 |

**Definition** – The measure of association of the gaming engine developer with U.S. adversaries. This metric is quantified in four categories. Developers with any connection to a U.S. adversary are not considered. Stakeholder value is maximized when the gaming engine developer has no connection to a U.S. adversary.

**Value Curve Shape** – The stakeholder perceives value through the use of a gaming engine whose developer's interests align with U.S. values or whose developers do not have hostile intent to the DOD. Similar to how the DOD assesses intelligence sharing, U.S. companies are perceived by the stakeholder to have the most value. Gaming engines developed by Allies of the U.S. have moderate value and Nuetral developers have minimal value.

**IMAGE PROCESSING SPEED (CPU)**

Table 39. Image Processing Speed (CPU) Value Curve

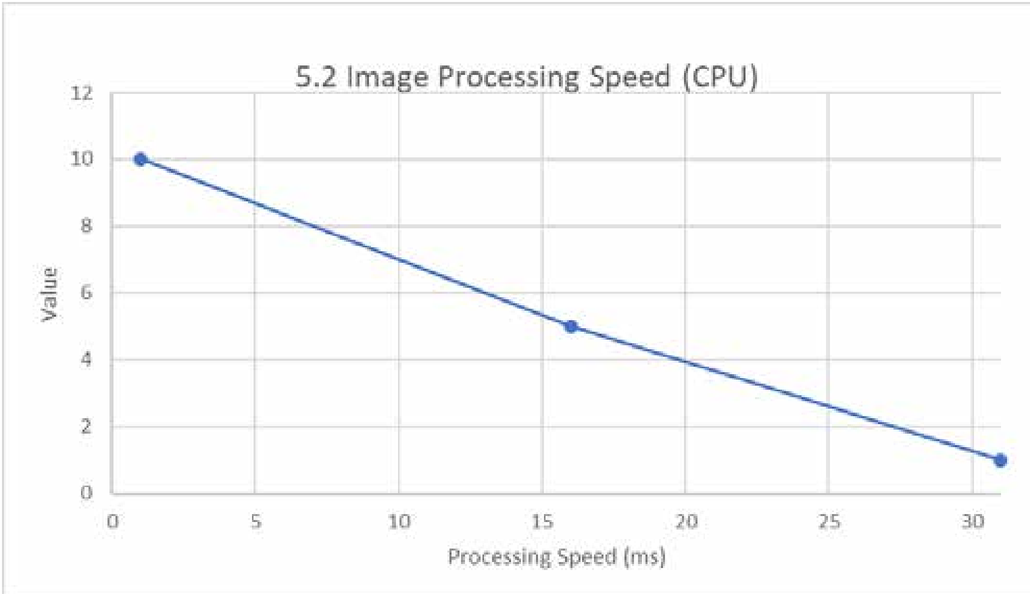



Table 40. CPU Processing Speed Values

| Processing Speed (in ms) | Value |
|---|---|
| 31 | 1 |
| 16 | 5 |
| 1 | 10 |

**Definition** – A measure of the gaming engine's algorithms to enable CPU image processing. This metric is measured in milliseconds. Algorithms that support faster processing speeds are more valuable to the stakeholder.

**Value Curve Shape** – The stakeholder perceives value through the gaming engine supporting increasing processing speed. Stakeholder value is maximized when the gaming engine algorithm enables processing speeds of 1 ms or greater. Gaming engines that do not support processing speeds at 31 ms and below are not considered.

## IMAGE PROCESSING SPEED (GPU)

Table 41. Image Processing Speed (GPU) Value Curve

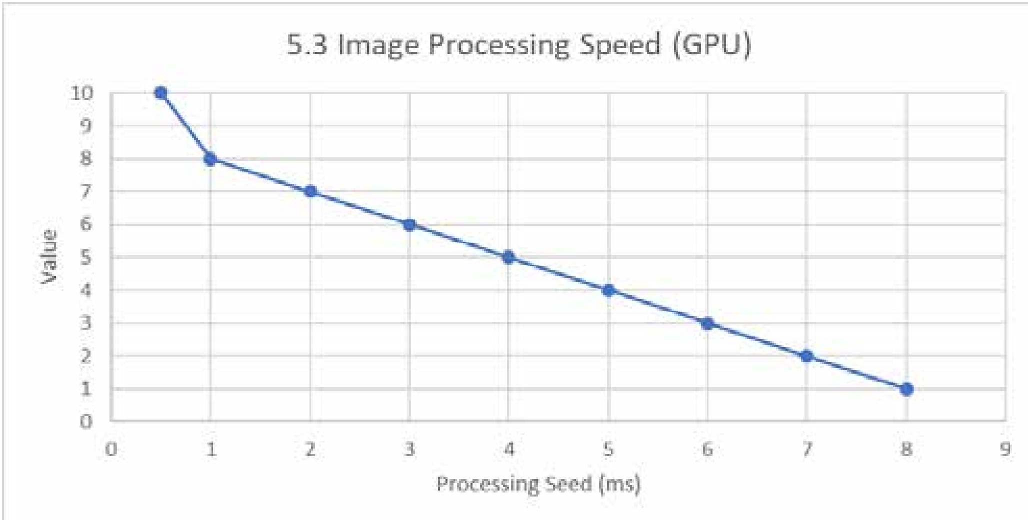



Table 42. Processing Speed (GPU) Values

| Processing Speed (in ms) | Value |
|---|---|
| 8 | 1 |
| 7 | 2 |
| 6 | 3 |
| 5 | 4 |
| 4 | 5 |
| 3 | 6 |
| 2 | 7 |
| 1 | 8 |
| 0.5 | 10 |

**Definition** – A measure of the gaming engine's algorithms to enable GPU image processing. This metric is measured in milliseconds. Algorithms that support faster processing speeds are more valuable to the stakeholder.

**Value Curve Shape** – The stakeholder perceives value through the gaming engine supporting increasing processing speed. Stakeholder value is maximized when the gaming engine algorithm enables processing speeds of 0.5 ms or greater. Gaming engines that do not support processing speeds at 8 ms and below are not considered.

SOFTWARE UPDATE FREQUENCY

Table 43. Software Update Frequency Value Curve

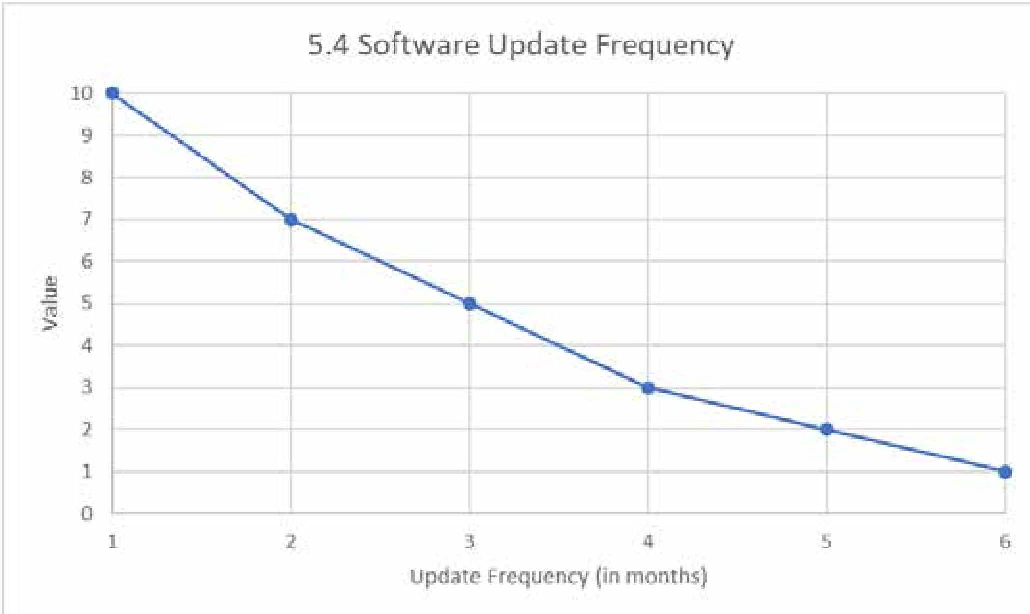



Table 44. Update Frequency Values

| Update Frequency (in months) | Value |
|---|---|
| 6 | 1 |
| 5 | 2 |
| 4 | 3 |
| 3 | 5 |
| 2 | 7 |
| 1 | 10 |

**Definition** - The interval between gaming engine software updates released by the developer. This metric is measured in months. Increased update frequency supports high performance and security objectives of the stakeholder. Stakeholder value increases as the interval between updates decreases.

**Value Curve Shape** – The stakeholder perceives value through decreased intervals between gaming engine software updates. Developer updates releases less than semi-annually do not support the stakeholder and are not considered. Decreasing frequency of updates exponentially increases stakeholder value. Stakeholder value is maximized with monthly software updates.

# LIST OF REFERENCES


Beigbeder, Tom, Rory Coughlan, Corey Lusher, John Plunkett, Emmanuel Agu, Mark Claypool. 2004. "The Effects of Loss and Latency on User Performance in Unreal Tournament 2003." Computer Science Department of Worcester Polytechnic Institute.

De Guzman, Jaybie A., Kanchana Thilakarathna, and Aruna Seneviratne. "Security and Privacy Approaches in Mixed Reality: A Literature Survey." ACM Computing Surveys (CSUR) 52, no. 6 (2018): 1–37.

Delligatti, Lenny. 2014. *SysML Distilled: A Brief Guide to the Systems Modeling Language.* New Jersey: Pearson Education, Inc.

Department of Transportation. 2007. *Systems Engineering for Intelligent Transportation Systems: An Introduction for Transportation Professionals.* Washington, DC: U.S. Department of Transportation, Federal Highway Administration.

Haselton, Todd. 2020. "How the Army Plans to Use Microsoft's High-Tech HoloLens Goggles On the Battlefield." Saturday 6. Accessed May 15, 2019. cnbc.com/2019/04/06/microsoft-hololens-2-army-plans-to-customize-as-ivas.html.

INCOSE, (International Council on Systems Engineering). 2015. *Systems Engineering Handbook*. John Wiley & Sons.

Krichenbauer, Max, Goshiro Yamamoto, Takafumi Taketom, Christian Sandor, and Kato Hirokazu. 2018. "Augmented Reality Versus Virtual Reality for 3D Object Manipulation." IEEE Transactions on Visualization and Computer Graphics (Institute of Electrical and Electronics Engineers) 24 (No. 2).

Miller, Gregory. 2019a. "Problems, Wants, Needs, Stakeholders." Class notes for SE3100: Fundamentals of Systems Engineering, Naval Postgraduate School, Monterey, CA. https://cle.nps.edu/access/content/group/f57bf35a-882f-4aeb-b7f0-4dbc2ba71985/Modules_Sp10/2%20Problem%20Statement%20and%20Stakeholders/problem%20and%20stakeholders.pptx

Miller, Gregory. 2019b. "Boundaries, Relationships to Peer Systems, Scenario-based Methods." Class notes for SE3100: Fundamentals of Systems Engineering, Naval Postgraduate School, Monterey, CA. https://cle.nps.edu/access/content/group/f57bf35a-882f-4aeb-b7f0-4dbc2ba71985/Modules_Sp10/3%20Boundaries%20etc/boundaries%20and%20scenarios.pptx

Miller, Gregory. 2019c. “Functional Analysis.” Class notes for SE3100: Fundamentals of Systems Engineering, Naval Postgraduate School, Monterey, CA. https://cle.nps.edu/access/content/group/f57bf35a-882f-4aeb-b7f0-4dbc2ba71985/Modules_Sp10/4%20Functional%20Analysis/functional%20analysis.pptx

MITRE, The Corporation. 2014. *System Engineering Guide.* Bedford, MA: MITRE Corporation Corporate Communications and Public Affairs.

National Defense Industrial Association (NDIA). 2011. *“Final Report of the Model Based Engineering (MBE) Subcommittee.”* 10 February 2011. https://www.ndia.org/-/media/sites/ndia/meetings-and-events/divisions/systems-engineering/modeling-and-simulation/reports/model-based-engineering.ashx

Parnell, Gregory S. and Timothy E. Trainor. 2009. “Using the Swing Weight Matrix to Weight Multiple Objectives.” In Proceedings of the INCOSE International Symposium, Singapore, July 19–23, 2009. https://doi.org/10.1002/j.2334-5837.2009.tb00949.x.

Pattrasitidecha, Akekarat. 2014. *Comparison and evaluation of 3D mobile game engines.* Goteborg, Sweden: Chalmers University of Technology.

Petridis, Panagiotis, Ian Dunwell, Sara de Freitas, and David Panzoli. 2010. “An Engine Selection Methodology for High Fidelity Serious Games.” In 2010 Second International Conference on Games and Virtual Worlds for Serious Applications, 27–34. IEEE. https://doi.org/10.1109/VS-GAMES.2010.26.

SnapFiles. 2018. “Unreal Engine 4 Version History Log.” Accessed 6 November 2020. https://www.snapfiles.com/apphistory/unreal_history.html.

Summers, Joshua D. (2005). “Reasoning in Engineering Design.” ASME 2005 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, September 2005.

Tremblay, Sébastien, Patrick Jeuniaux, Paul Romano, Jacques Lowe, and Richard Grenier. 2011. “A multi-perspective approach to the evaluation of a portable situation awareness support system in a simulated infantry operation.” In 2011 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), pp. 119–122. IEEE, 2011.

Unity Technologies. “Unity QA – LTS Releases – Unity.” Accessed 6 November 2020. https://unity3d.com/unity/qa/lts-releases.

Worsley, M., & Blikstein, P. (2016). “Reasoning Strategies in the Context of Engineering Design with Everyday Materials.” *Journal of Pre-College Engineering Education Research (J-PEER)*, 6(2), Article 4. https://doi.org/10.7771/2157-9288.1126

Xinogalos, S. (2017, December 4). “Overview and comparative Analysis of Game Engines for Desktop and Mobile Devices.” *International Journal of Serious Games*, 4, 21–36.

THIS PAGE INTENTIONALLY LEFT BLANK

# INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California