Literature Survey – Paper 1 :
https://www.um.edu.mt/library/oar/bitstream/123456789/82141/1/Automatic_game_script_generation_2015.pdf

This work creates a system that automatically generates PuzzleScript game level and rules using procedural methods. It shows that playable levels can be produced quickly and accurately.

Relation to GS: Since GS also has simple rule-based logic, this paper supports how such rules can be created and used effectively in game development.

Literature Survey – Paper 2:
https://ieeexplore.ieee.org/abstract/document/5972135

This paper presents a new method for developing a mobile game script engine using a slice-array technique. It solves memory, font, and performance issues found in traditional J2ME script systems.

Relation to GS: It supports GS by showing how improved script-engine design can make game logic faster, lighter, and easier to manage, similar to GS's goal of creating a simple rule-based scripting approach.

Literature Survey – Paper 3:
https://iopscience.iop.org/article/10.1088/1757-899X/434/1/012053/pdf

This paper shows how script-based knowledge representation can be used to design educational games, combining elements like roles, scenes, and conditions into a clear structure.

Relation to GS: It supports GS because both use simple script-like structures to organize game logic and make game design easier.

Literature Survey – Paper 4:
https://www.tandfonline.com/doi/epdf/10.1080/14626268.2019.1570942?needAccess=true

This paper introduces a script-centric approach for writing interactive game narratives, using play-through traces to help writers handle branching dialogues without needing programming skills.

Relation to GS: It connects well with GS because both aim to simplify game scripting and make narrative logic easier to manage, especially for non-technical users.

Literature Survey – Paper 5:
https://spawn-queue.acm.org/doi/pdf/10.1145/1483101.1483106

This article explains how improving game scripting languages can make game development faster and help create smoother, more efficient gameplay. It highlights the importance of clean, powerful script systems in modern games.

Relation to GS: It is relevant because GS also aims to provide a simple and efficient scripting method, showing how better script design can improve game logic and development.

Literature Survey – Paper 6:
https://peer.asee.org/21380.pdf

This paper introduces simple video game scripting engines that let students write small scripts to control game characters. The engines use a limited instruction set, making scripting easy to learn and experiment with.

Relation to GS: It relates well to GS because both aim to make game scripting simple and rule-based, helping users control game behavior without complex programming.

Literature Survey – Paper 7:
https://ieeexplore.ieee.org/abstract/document/10062557

This paper describes designing a 2D text-based adventure game in Unity, creating original scripts, characters, and interactions. The script controls dialogue, story flow, and puzzle progression.

Relation to GS: It is relevant because it shows how game behavior and story flow rely on custom scripts—similar to how GS defines actions and logic using simple, rule-based scripting.

Literature Survey – Paper 8:
https://www.proquest.com/openview/747f89dee4c80088d42f42c2f06abb04/1?pq-origsite=gscholar&cbl=5452619

This paper introduces CeGAS, a gamification authoring system that helps teachers create game-based learning activities using a simple script editor based on the 5E learning model. It allows users to design game content without dealing with technical complexity.

Relation to GS: It relates to GS because both aim to simplify game content creation through an easy scripting interface, showing how scripts can help non-programmers design structured game tasks and interactions.

Literature Survey – Paper 9:
https://iopscience.iop.org/article/10.1088/1742-6596/1992/3/032108/pdf

This paper discusses the design of a high-performance Java-based mobile game engine, focusing on architecture, module selection, and performance optimization. It analyzes user needs and technical constraints to build an efficient engine for mobile games.

Relation to GS: It is relevant because GS also depends on engine support to execute scripts. The paper provides background on how engines handle performance and modules, helping you justify the need for simple and efficient scripting systems.

Literature Survey – Paper 10:
https://www.um.edu.mt/library/oar/bitstream/123456789/27604/1/A_Domain-Specific_Embedded_Language_Approach_for_t-2-8.pdf

This paper presents a method for creating game scripting languages by embedding specialized scripts inside a general-purpose host language. It explains how fixed and adaptive scripts can be represented as data objects, allowing flexible behavior definition.

Relation to GS: It is relevant because GS is also a domain-specific scripting language. This paper supports the idea of designing simple, rule-based scripts to control game behavior and shows how scripting can be structured at different abstraction levels.

Literature Survey – Paper 11:
https://link.springer.com/article/10.1007/s11042-025-20899-8

This study analyzes the maintainability and software quality of popular open-source game engines using static code analysis. The results show that many widely used engines suffer from poor code quality, which affects long-term performance and sustainability.

Relation to GS: This paper supports the need for simple and well-structured scripting systems like GS. By reducing complex engine-level code and using clear game scripts, GS can help improve maintainability and ease of game logic development.

Literature Survey – Paper 12:
https://www.nature.com/articles/s41598-025-16830-8

This paper proposes a reputation-based incentive mechanism for Hierarchical Federated Learning to encourage reliable device participation. It uses reputation scores, blockchain, and game theory to improve training quality and system performance.

Relation to GS: It focuses on federated learning and incentive mechanisms rather than game scripting or game logic design. It does not contribute to scripting languages or rule-based game systems.

Literature Survey – Paper 13:
https://ieeexplore.ieee.org/document/6884801

This paper introduces an extensible description language designed to define game elements such as objects, behaviors, and rules in a structured and flexible way. The language allows games to be easily extended or modified without changing core engine code.

Relation to GS: It is closely related to GS because both focus on using a dedicated description or scripting language to represent game logic and behavior. This supports the

idea that game functionality can be controlled through simple, extensible scripts instead of complex programming.

Literature Survey – Paper 14:
https://www.researchgate.net/profile/Mubeen-Aslam-4/publication/342380760_A_Domain_Specific_Modeling_Language_for_Adventure_Educational_Games_and_Flow_Theory/links/5f2bae7a92851cd302dfbae2/A-Domain-Specific-Modeling-Language-for-Adventure-Educational-Games-and-Flow-Theory.pdf

This paper proposes an extended domain-specific modelling language (FA-GLiSMo) to design adventure educational games by combining game logic, structure, and flow theory. The language helps developers and educators collaboratively model game behavior and progression using clear domain concepts.

Relation to GS: It is closely related to GS because both use domain-specific languages to represent game logic and structure in a simplified form. The paper supports the idea that game behavior can be defined using structured models or scripts instead of complex engine code.

Literature Survey – Paper 15:

https://dl.ifip.org/db/conf/iwec/icec2006/Moreno-GerMSF06.pdf?utm_source=chatgpt.com

This paper presents a language-driven approach to video game development, where a custom domain-specific language (DSL) is created to define game logic and structure before building the engine and games themselves. The process shows how a specialized language can simplify game creation by separating game behavior definition from engine code.

Relation to GS: It strongly relates to your GS project because both focus on designing a custom game scripting language and using it as the basis for building games. This supports the idea that DSLs make game logic clearer and easier to maintain.

Literature survey -- paper 16:

https://dl.acm.org/doi/epdf/10.1145/3582437.3582467

Paper Description:
This paper studies PuzzleScript programs to understand how rule-based scripts are used to define game mechanics. By analyzing source code, it identifies common patterns, limitations, and the expressive power of the language.

Relation to GameScript:
The paper relates to GameScript as both use declarative, rule-based scripting for game logic. It supports the idea that simple rules can effectively represent complex game behavior.

Literature Survey – Paper 17:

https://www.researchgate.net/profile/Martin-Hanneghan/publication/267801341_Towards_a_Domain_Specific_Modelling_Language_for_Serious_Game_Design/links/5512d8900cf268a4aaeb3061/Towards-a-Domain-Specific-Modelling-Language-for-Serious-Game-Design.pdf

Paper Description:
This paper explains how Model-Driven Engineering can be used to simplify the design of serious games. It proposes a Domain Specific Modelling Language (DSML) that captures game design elements in a structured way, reducing development complexity.

Relation to GameScript:
This work relates to GameScript because both aim to simplify game development using high-level, domain-specific concepts. While the DSML uses models, GameScript uses rule-based scripting, but both focus on clarity, structure, and maintainability of game logic.

Literature Survey – Paper 18:

https://dl.acm.org/doi/epdf/10.1145/1401843.1401847

Paper Description:
This paper applies declarative processing techniques to game development using the state–effect pattern. It introduces a scripting language that can be compiled into declarative languages like SQL to improve performance and simplify game state handling.

Relation to GameScript:
This paper relates to GameScript as both use declarative, rule-based logic instead of procedural scripting. The separation of game state and behavior aligns with GameScript's rule and state evaluation approach, supporting clarity and maintainability.

Literature Survey – Paper 19:
https://dl.gi.de/items/ad00b5ed-c8a0-459d-8b98-1e8b5bea44ae

Paper Description:
This paper presents a model-driven framework to simplify serious game development. It uses a domain-specific modeling language and visual tools so educators can design games without writing code, which are then transformed into playable games.

Relation to GameScript:
This work relates to GameScript as both reduce development complexity using high-level abstractions. While GameScript uses rule-based scripting, this framework uses models, but both emphasize clarity, maintainability, and easier game logic design.

Literature Survey – Paper 20:
Hernandez.pdf

Paper Description:
This paper introduces Eberos GML2D, a graphical domain-specific language for designing 2D video games. It allows developers to define game entities, behaviors, states, and interactions using visual models, which are later converted into executable game code.

Relation to GameScript:
This work relates to GameScript as both simplify game development using domain-specific abstractions. While GameScript uses rule-based scripting, Eberos GML2D uses visual models, but both focus on clear, structured, and maintainable game behavior definition.