IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 11, April 2026
Impact Factor: 8.2

DreamPlay-AI : AI Game Generation

Asst. Prof. Achala. P. Narayankar, Pratik S Nalawade, Sarvesh K Gudekar, Ritesh M Noukudkar
Vaishnavi N Raktade, Priyanka P Devgonda.
Department of Computer Science and Engineering
Sant Gajanan Maharaj College of Engineering, Mahagaon Kolhapur, India

Abstract: *Game development usually feels out of reach for anyone who isn't a programmer or designer. If you don't know how to code or create graphics, you're left out—and that's a shame. DreamPlay AI changes the game by letting anyone make playable 2D games just by describing what they want in plain language. The platform taps into advanced Natural Language Processing (NLP) and Machine Learning (ML) to turn your words into structured game blueprints. Those blueprints cover all the building blocks: mechanics, rules, characters, setting, and what you're supposed to do. After DreamPlay AI processes your idea, it grabs one of its pre-built templates—maybe a shooter, a racer, a puzzle, or an endless runner—and uses it as a launchpad. Then, the system swaps in unique assets, so each game feels fresh and matches your description. It doesn't just change visuals like characters and backgrounds; it tweaks obstacles and objects too. You don't need to wrestle with confusing tools or code. Just type your idea, hit play, and you're testing your own game in seconds.Along the way, DreamPlay AI breaks down walls in education, entertainment, and rapid prototyping. Students, hobbyists, and people who never thought they could build games get to join in. It boosts creativity, speeds up development, and slashes the barriers keeping newcomers out. By marrying smart AI with simple design, DreamPlay AI opens the door to game creation for everyone..*

Keywords: DreamPlay AI, Game Generation, Asset Swapping, Natural Language Processing (NLP), Machine Learning (ML), 2D Game Development, Procedural Content Generation, No-Code Development

I. INTRODUCTION

Artificial intelligence has changed the game—literally and figuratively—when it comes to creating and playing with digital content. Out of all these new developments, AI-powered game generation really stands out. Suddenly, anyone can make their own games. Doesn't matter if they know how to code or not. The whole process is getting easier and faster, stripping away that old requirement for deep programming knowledge, design skills, and endless hours of work. People are demanding tools that let them skip straight to building things—no more wrestling with complicated software or spending weeks crafting a prototype. That's where DreamPlay AI comes in. It's got this clever setup: you just type out a prompt, something as simple as "make a monster shooting game," and it transforms your idea into a living, playable 2D game. Behind the curtain, it's mixing smart templates, a machine learning classifier, and logic for swapping assets—so it's not just slapping things together. It actually gets what you want, figures out the genre, nails the mechanics, and then spits out a working game. DreamPlay AI isn't just for techies. Students, developers, teachers—anyone can jump in, even if they've never built a game before. If you want to explain game logic in a classroom, it helps you do that in seconds. And here's the really exciting part: every prompt, every bit of feedback, teaches the system and makes it better. The whole project keeps improving the more people use it, creating this kind of feedback loop where the audience literally shapes the tool's growth. When you step back, it's pretty clear DreamPlay AI is opening up game creation to just about everyone. The project makes building games accessible, quick, and honestly, pretty fun. The fact you can use plain language to bring your ideas to life signals where AI-assisted creativity is

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 11, April 2026 Impact Factor: 8.2

heading. It's not just about technology—it's about unleashing imagination for anyone who's got an idea and wants to see it in action.

II. LITERATURE REVIEW

Artificial Intelligence (AI) is at the heart of modern game development. It's not just about adding a few clever tricks to make the game more challenging—it's fundamentally changing the way designers build games, automate processes, and bring worlds to life. A striking example is the rise of intelligent agents or Non-Playable Characters (NPCs) that behave almost like real players.Take Sharma's 2019 study. It digs into building a 2D AI agent for games that don't follow a set s cript, using Convolutional Neural Networks (CNNs). With these models, characters can handle decisions in the moment—should they jump, duck, or do something unexpected in a tough platformer? The result is characters that start to act more like humans, sensing and reacting on the fly. It proves that deep learning isn't just hype; it genuinely helps handle situations where every second is unpredictable. There's a catch, though: these models usually get stuck focusing on just a few game mechanics. So, they're smart in the right setting, but you can't plop them into any old game and expect magic. Looking at the bigger picture, Dhanikonda and Laxmi (2024) offer a snapshot of how AI has changed game development over time. They stress that AI isn't just a feature—it's become a driving force behind making games more realistic, interactive, and, frankly, just more fun. It shows up everywhere: smart character behavior, clever pathfinding, and deep game analytics that make each playthrough feel fresh. These upgrades make games more immersive, but let's be honest, pulling them off isn't easy. It takes a team with serious technical skills and the patience to wrangle different technologies into a seamless whole. Then there's Procedural Content Generation via Machine Learning (PCGML)—a concept that Summerville and colleagues unpacked back in 2015. Imagine AI models, from neural networks to Markov models and evolutionary algorithms, working behind the scenes to create new levels, maps, and even the rules of the game itself. Suddenly, developers spend less time on repetitive design, and players enjoy content that feels new every time. It's a big leap toward adaptive, creative game design. But it's not perfect. The hurdles are real: developers struggle with getting enough high-quality training data, making sure the content feels polished, and guaranteeing that AI-made levels are actually fun and playable. Roberts and Chen (2019) push this even further by looking at learning-based procedural content generation.

III. PROBLEM STATEMENT

Digital creativity tools have come a long way, and it's a lot easier now to access the world of game development. Still, even putting together a simple 2D game isn't exactly a walk in the park. You need some serious know-how—programming, making and handling game assets, sorting out game logic, dealing with collisions, and just getting all the parts to work together as a system. There's the core loop to design, sprites to manage, physics to make feel right, and levels to organize. All those steps take familiarity with game engines and at least some scripting skills, if not full-on programming. For plenty of students, casual hobbyists, or anyone outside the programming world, that learning curve can slam the brakes on their creative ambitions. Even with a flood of new AI helpers like Gemini, Copilot, and the ever-present ChatGPT, you mostly get isolated code snippets or quick chunks of logic. They don't spit out a fully-built, ready-to-play game. The process is still hands-on—you have to piece things together, troubleshoot what the AI gives you, and actually get the code to run inside a game engine or framework. These tools simply aren't built to tackle the whole package: picking the right graphics, dropping in systems like game mechanics, sorting how characters move, adapting art or assets to match a user's idea, and keeping gameplay consistent. Most of the time, you still need to debug or polish what you get before the game will really work, and, again, that takes technical chops. That's where the story really gets interesting—there's a clear gap that calls for a smarter system, something that doesn't just generate code but actually creates a full working game out of a user's idea. Imagine a platform that understands normal human language, figures out the kind of game you want to make, and automatically brings in everything you need: the player character, obstacles, enemies, a background, and anything else that makes your vision come to life. Using advances in machine learning, this system picks up your intentions, chooses fitting templates, and pieces together all the art, code, and logic

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 11, April 2026
Impact Factor: 8.2

to assemble a real, playable game. DreamPlay AI exists to fill this very gap. It takes natural language prompts, breaks down the requirements, and then—without needing you to write a single line of code—it crafts a fully functional 2D mini-game. The tool handles template selection and asset management in the background, so you get consistent gameplay and something you can use right away. The real value here? Game creation suddenly becomes accessible to everyone, whether you're a student, teacher, or a complete beginner. Your ideas can jump from your head to a working game in no time. It doesn't just save hours of frustration; it makes the whole creative process way more fun and open for people who've never touched code.

IV. PROPOSED SYSTEM OVERVIEW

DreamPlay AI is pretty cool—it takes a simple text prompt from the user and turns it into a fully playable 2D game, all without needing any programming know-how. You just type in your idea, like "Build a monster shooting game in a forest," and the system figures out exactly what you mean using artificial intelligence and a bunch of templates that are ready to go. First, DreamPlay AI uses Natural Language Processing to break down your sentence. It pinpoints things like the kind of game you want (in this case, a shooting game), where it happens (the forest), who you play as (probably a hero-type character), and who you fight (monsters). Once it's got all the important bits from your prompt, it uses a machine learning model to sort the game into a category, like "action shooting." With that, DreamPlay AI picks a template that matches—these templates already have the basics baked in: player movement, shooting, enemy behavior, scoring, and collision detection. You don't have to touch any code; the system does all the heavy lifting. After picking the right template, DreamPlay AI automatically finds assets that fit your theme. For a forest vibe, it chooses backgrounds like trees and grass, player sprites like a hunter or soldier, and monsters for enemies. It tweaks everything so the visuals match and don't look out of place. Now, the engine applies all the essential game logic—moving side to side, firing bullets, tracking when things collide, counting points when monsters get knocked out. It sets up rules for gameplay too, like how often monsters appear, how much health you start with, and how tough things get as you play. Every component comes together behind the scenes, and you're never stuck fiddling with settings or code. At the end, DreamPlay AI hands you a finished game file or an interface—just press the "Play" button and you're off. Basically, it feels like having a smart game builder that listens, understands your idea, and turns it into a game right away. It makes game-making easy for newbies, students, or anyone who's never coded before but wants to see their game ideas come to life.

V. SYSTEM ARCHITECTURE

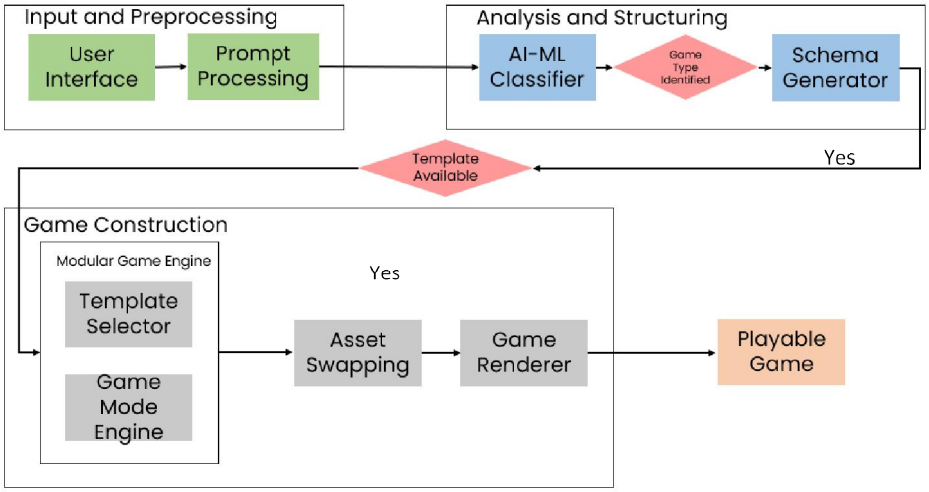


Fig. 1. System Architecture of DreamPlay AI

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 11, April 2026
Impact Factor: 8.2

VI. SYSTEM ARCHITECTURE OVERVIEW

DreamPlay AI takes a user's plain text prompt and turns it into a ready-to-play 2D game. It does this in three big stages: Input & Preprocessing, Analysis & Structuring, and Game Construction. It all starts with the user interface. Someone types in a request—maybe, "Build a monster shooting game in a forest." That's sent to the Prompt Processing module, which doesn't just accept the prompt at face value. First, it cleans up the text, strips out random noise or filler words, and pulls out key information—stuff like "monster," "shooting," "forest." It then restructures the cleaned prompt so the system can actually make sense of it. The point here is to cut through any confusion and really figure out what the user wants. Now comes the Analysis & Structuring phase. Here's where the AI-ML Classifier jumps in. It examines the processed prompt and figures out what kind of game you're asking for—shooting, racing, platformer, whatever. It even sorts out all the essentials: player characters, enemies, environments, main actions. If it decides you asked for a shooting game in a forest, for example, it confirms the request is valid and hands the details to the Schema Generator. The schema is like a blueprint—it spells out every piece of the game, from rules and object relationships to mechanics and needed assets. So, it'll lay out stuff like the player's ability to shoot, how and where enemies spawn, and what the forest background should look like. Once the schema is mapped out, the system checks if there's already a fitting template available. If there is, things move forward. The last stage is Game Construction. The Modular Game Engine takes over. First, the Template Selector picks the most suitable template—say, a shooting game template. Then the Game Mode Engine lays down the main gameplay logic: movement, shooting action, keeping score, collision detection—basically, everything that makes the game tick. After that, the Asset Swapping module brings life to the game, matching visuals with the theme. So for the forest shooter, it sets up a forest scene, puts in player sprites, and drops in monster enemies. The Game Renderer then ties all the parts together—game logic, assets, structure—and builds an actual, functional game. All this happens behind the scenes. When the dust settles, DreamPlay spits out a fully playable game that anyone can run right away, without needing to write code or debug anything. It's pretty much plug-and-play.

VII. IMPLEMENTATION DETAILS

DreamPlay AI is all about building a smart, adaptable system that lets anyone create 2D games just by describing what they want — no coding needed. Instead of locking everything into one specific game style, the real magic here is with these dynamic game modes. Think of it less like a single, fixed blueprint and more like a toolbox full of different logic modules. The system grabs whatever pieces fit your idea and snaps them together on the fly. It kicks off when you type in your game idea—maybe you want an old-school shooter or a fast-paced racing game. The system reads your prompt, picks out the important details, and figures out exactly what kind of game you're after and which features you'll want. From there, it fires up the right game mode inside its modular engine. Each game type comes with its own set of logic modules. So, if you ask for a shooter, the system can set it up as a run-and-gun arcade challenge or maybe a careful sniper test. For racing, the modules could handle things like endless tracks or dodging obstacles. None of these options are hardwired as complete, separate games—they're more like building blocks the AI can mix and match, depending on what you want. Once it knows which modules to use, the system connects them to a basic template. This template gives the game its overall shape, and then the logic modules bring it to life — things like player controls, scoring, spawning enemies for shooters, or adding speed and obstacle logic for racing games. Dividing the structure from the behavior is what makes DreamPlay AI so flexible. What really sets the system apart is that your prompt isn't just changing how the game looks. It actually tweaks how the core of the game works. That means using the same underlying template, you can get wildly different experiences. Tell it to make "a monster shooting game in a forest," and you'll get something totally different from "a sniper challenge in a desert." Even if the base is the same, the AI shifts mechanics, behaviors, and feel to match your idea. This kind of separation and dynamic mixing turns DreamPlay AI into a truly adaptive game creator, where your imagination sets the limits.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 11, April 2026
Impact Factor: 8.2

TABLE I: TECHNOLOGY STACK USED IN THE PROPOSED SYSTEM

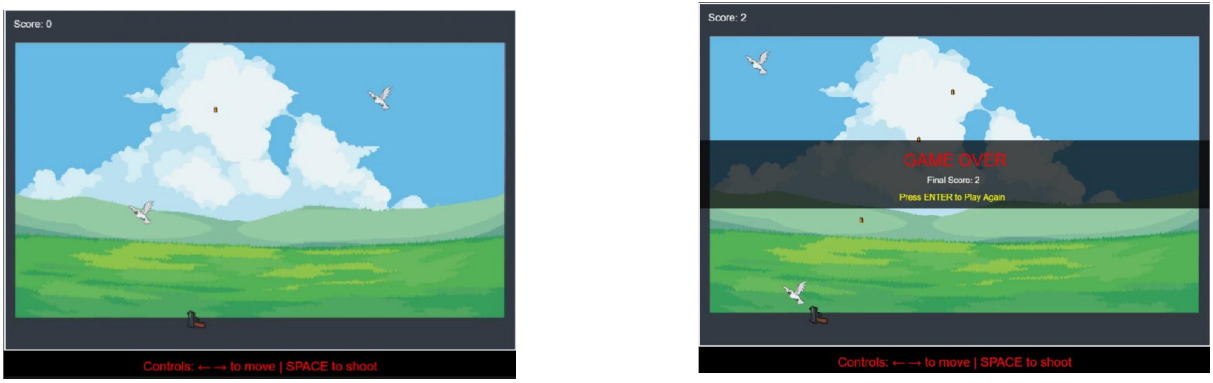
Layer	Technology and Purpose
Frontend Technologies	HTML5 Canvas, JavaScript, CSS
Backend Technologies	Python, Flask
AI / ML Libraries	NLP-based Text Classification, TF-IDF Vectorizer, Naive Bayes Classifier
Data	CSV-based training dataset, Schema-driven configuration
Hardware Requirements	Processor: Intel i3 or higher RAM: Minimum 4 GB Graphics: Integrated graphics (sufficient for 2D games)

VIII. EXPERIMENTAL RESULTS AND ANALYSIS

DreamPlay AI really shows its potential during experimental testing—the system managed to turn a plain text prompt into a fully functional 2D game, start to finish. Looking at the shooter game test, it's pretty impressive: you get a playable game where you control a character, try to hit moving targets (like birds), and there's a working shooting mechanism built right in. The results speak for themselves. The game's world loads properly—with a background scene, animated birds fluttering around, and score tracking that actually updates as you play. This means all the visual pieces and the game logic fit together without any obvious trouble. One thing that stands out is how the game gets built on the fly, straight from the user's prompt, with zero hand-coded intervention. DreamPlay AI figures out it's a shooter, drops in the core gameplay—character movement, shooting—and slaps on the right mechanics. On top of that, it puts clear on-screen instructions for how to move and shoot, so players know what to do. This shows the system isn't just making a game; it's thinking about the player experience, too. Asset management works well here. The system chooses the right backgrounds, main character sprites, and enemy birds, then places them where they need to go—all automatically, guided by the prompt. DreamPlay's asset swapping module keeps things adaptable, so each playthrough can swap in new visuals if the prompt changes. There's another detail: enemy birds behave differently during the game. They don't just fly in straight lines—they shift around, reacting to the gameplay, which shows that some AI-based logic is steering their movement and interactions. From start to finish, the gameplay feels complete. Everything functions as you'd expect: the game screen updates live, the score ticks up in real time, and when it's over, you get a "Game Over" screen. There's even a prompt to start again ("Press ENTER to Play Again"), which proves the game loop and state management are in place and reliable. Stepping back, DreamPlay AI pretty much delivers on its big promise: it automates game development while keeping things playable and interactive, and it does away with most of the manual coding, blending pre-set templates with live logic and assets. It lowers the bar for creating simple games, making things easy for folks who don't want to mess with the code themselves. That said, these results only scratch the surface. Right now, it works best for simple, basic games. There's real room to grow—think more complex game mechanics, fancier graphics, or support for whole new genres. With more development, DreamPlay AI could open up way more creative possibilities. For now, though, it's impressive to see a game spin up out of just a few words, and actually work.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 11, April 2026
Impact Factor: 8.2

RESULTS AND ANALYSIS



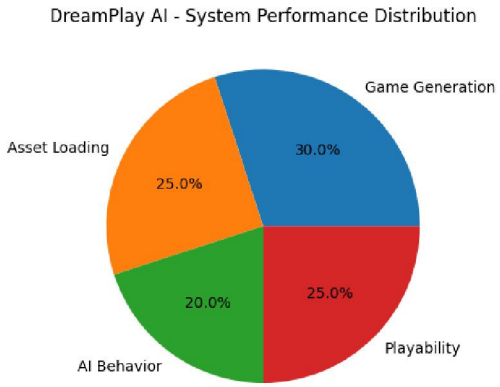


Fig 2. System Performance Distribution

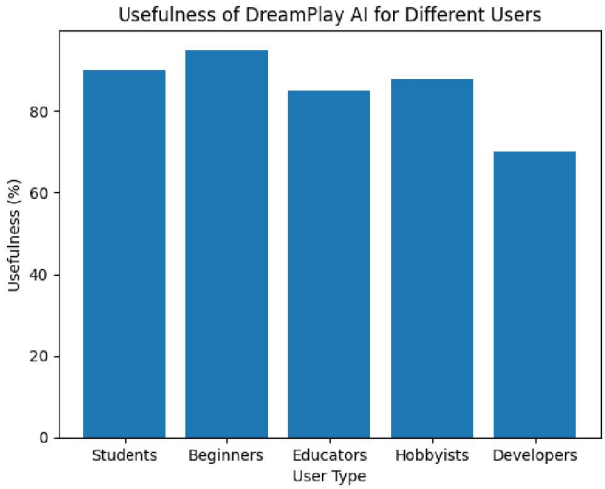


Fig 3. Usefulness graph for different Users

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 11, April 2026
Impact Factor: 8.2

IX. DISCUSSION

DreamPlay AI's experimental results show something pretty striking: when you bring together AI, automation, and template-driven game creation, you can really cut down the hassle and the time it takes to make a game from scratch. Right now, if you want to use typical game development tools, you need to know how to code, pick out every last asset yourself, and figure out all the tricky logic that holds your game together. It's a pile of work, and it's definitely not beginner-friendly. DreamPlay flips this on its head. Instead of grappling with code and chasing down bugs, users just type a natural language prompt, and—almost like magic—the system spits out a fully playable 2D game. You no longer need to wade through hours of programming or struggle with fiddly debugging. The difference in speed and effort is huge. Games pop into existence almost instantly, with hardly any manual input. What really makes this system tick is its AI-powered classification and the way it matches templates dynamically. It doesn't just throw together random elements; the AI listens to what the user actually wants (or types, really), picks out fitting game mechanics and assets on its own, and builds out the logic automatically. The result? Game creation feels much smoother and more reliable. The modular design helps a lot, too. You can swap in different game modes and behaviors on the fly, which means the platform isn't just fast—it's flexible and grows as your ideas get bigger. And then there's the way it handles graphics: asset swapping and the game rendering tools work together to make sure what you see always fits the theme you had in mind, down to the last detail. Not only does this make the games look better, it also keeps everything coherent—it's never a mismatch between gameplay and visuals. Maybe the best part is this: when DreamPlay finishes putting together your game, it's ready to play right away. You don't have to tweak anything or go back for another editing round. But—there's always a "but"—the system isn't perfect. Since it leans on a set of predefined templates, sometimes the variety or depth of the games can feel a bit limited. The way it pins down the genre and logic is only as good as the data and rules it started with, so if you ask for something super complicated or vague, it might miss the mark. And let's be honest—if you want features like multiplayer or advanced physics, that's still out of reach for now. Looking ahead, though, there's a lot of room for DreamPlay to grow. The plan is to sharpen those machine learning models so the AI gets even better at understanding prompts, widen the template library to cover more styles and genres, and start adding robust advanced mechanics. There's also interest in bringing in features like real-time tweaking, upgraded visuals, and even support for 3D games down the line. With steady improvement, DreamPlay could honestly change the way people approach game development, taking stumbling blocks out of the way and giving just about anyone a shot at making their own games with almost zero friction. It's a big step towards making game creation open to everyone, not just coders or design veterans.

REFERENCES

[1]. V. Sharma, "Implementation of a 2D A.I. Agent for Nondeterministic Games using Convolution Neural Network," 2019.
[2]. S. R. Dhanikonda and N. Laxmi, "Artificial Intelligence Usage in Game Development," 2024.
[3]. A. Summerville, S. Snodgrass, M. Guzdial, et al., "Procedural Content Generation via Machine Learning (PCGML)," 2015.
[4]. J. Roberts and K. Chen, "Learning-Based Procedural Content Generation," 2019.
[5]. IEEE, "IEEE Conference Author Guidelines," IEEE, 2023.