**Prompt Engineering – Utilizarea LLM-urilor în dezvoltarea proiectului**

În cadrul proiectului nostru 2D Endless Wave Survival Game, am utilizat modele de limbaj mari (LLM-uri) pentru a ne sprijini în diverse etape ale dezvoltării, în special în rezolvarea anumitor blocaje tehnice și în clarificarea unor concepte din Pygame. Deși am lucrat majoritar pe cont propriu, aceste modele au reprezentat o unealtă suplimentară utilă, mai ales în zone punctuale în care am întâmpinat dificultăți.

**Modelele utilizate**

Pe parcursul proiectului am apelat la trei LLM-uri diferite:  
  
- ChatGPT (OpenAI)  
- DeepSeek  
- Gemini (Google)  
  
Fiecare dintre acestea ne-a fost de ajutor în contexte diferite, în funcție de complexitatea întrebării sau de tipul de explicație pe care o căutam.

**Utilizări practice**

LLM-urile ne-au fost utile în principal pentru:  
- Fixarea bugurilor apărute în timpul dezvoltării, cum ar fi erori de runtime, probleme de inițializare a obiectelor, și clarificări legate de ordinea logică în cod.  
- Întrebări punctuale despre Pygame, cum ar fi: cum putem încărca sprite-uri dintr-un sprite sheet, cum putem anima un personaj în funcție de direcția de mers, sau cum să folosim hitbox-uri și coliziuni.  
- Sugestii de structură pentru organizarea clasei Enemy, inclusiv implementarea animațiilor de moarte și atac.  
- Clarificări legate de performanță sau bune practici, cum ar fi momentul potrivit pentru a elimina un sprite dintr-un grup sau cum să folosim pygame.time.get\_ticks() pentru cooldown-uri.

**Diferențe între LLM-uri**

Am observat că fiecare model are puncte forte ușor diferite:  
  
- ChatGPT a fost cel mai echilibrat în explicații clare și concrete. A fost util în special la depanare și când voiam să înțelegem „de ce nu merge” ceva. De asemenea, ne-a ajutat să înțelegem mai bine cum să organizăm structura proiectului.  
- DeepSeek s-a remarcat prin faptul că tinde să ofere soluții rapide și concise, fiind util mai ales în întrebări punctuale de cod, cum ar fi „cum se inversează o imagine în Pygame”.  
- Gemini a oferit uneori explicații detaliate și complete, dar în câteva cazuri a fost mai puțin precis când întrebările noastre erau legate strict de sintaxa Pygame. Totuși, a fost folositor pentru explicații generale legate de programare orientată pe obiect.

**Concluzie**

Per total, am folosit LLM-urile ca un instrument auxiliar, fără a depinde de ele pentru implementarea de la zero a funcționalităților majore. Ele au completat munca noastră prin oferirea de soluții la probleme punctuale și prin clarificarea unor concepte cu care nu eram familiarizați. Experiența de lucru cu aceste modele a fost una eficientă, mai ales când am știut cum să formulăm întrebările corect și să validăm răspunsurile obținute.