

Documentació DevChallenge5: Carta més alta

1. Introducció

El projecte "Carta més alta" és un minijoc en línia per a dos jugadors que competeixen seleccionant cartes d'una baralla, on guanya qui tria la carta amb major valor. Cada jugador rep 5 cartes aleatòries al començament de cada ronda, mostrades boca avall.

En seleccionar una carta, aquesta es revela, i al final de la ronda es mostren les cartes seleccionades per ambdós jugadors juntament amb el resultat (per exemple, "Jugador 1 guanya!"). El joc utilitza WebSockets per a la comunicació en temps real entre els clients i el servidor.

L'objectiu d'aquest DevChallenge és desenvolupar el joc utilitzant eines d'intel·ligència artificial (IA), com ChatGPT, DeepSeek i Grok, i documentar el procés de desenvolupament, posant èmfasi en com s'han utilitzat aquestes eines, els reptes trobats i les solucions implementades.

El desenvolupament es va iniciar amb ChatGPT i DeepSeek per establir les bases del joc i la interfície, i més endavant es va incorporar Grok per a correccions, millores específiques (com el repartiment aleatori de 5 cartes i la visualització de les cartes seleccionades).

2. Eines d'Intel·ligència Artificial Utilitzades

- **ChatGPT:**
 - **Ús principal:** Plantejament inicial del projecte, suggeriment de tecnologies (com WebSockets) i generació del codi base per a la lògica del joc i la comunicació amb el servidor.
 - **Interaccions:** Prompts amb pseudocodi per definir la lògica bàsica (crear baralla, repartir cartes, comparar valors) i intents d'integració amb un servidor asíncron, encara que amb resultats inicials poc satisfactoris.
- **DeepSeek:**
 - **Ús principal:** Generació de l'estructura HTML inicial i part de la lògica del client per a la interfície i la interacció amb les cartes.
 - **Interaccions:** Prompts per crear una interfície bàsica i funcional, tot i que el resultat inicial era molt simple i requeria millores.
- **Grok:**
 - **Ús principal:** Correcció de problemes específics, implementació de millores i tocs finals.
 - **Interaccions:** Converses iteratives amb prompts textuais detallats, incloent fragments de codi existents per a modificacions concretes i explicacions sobre tecnologies com WebSockets.
- **Metodologia d'ús de les IA:**
 - Només es van utilitzar prompts textuais amb ChatGPT, DeepSeek i Grok; no vaig fer servir autocompletadors de codi com Copilot.
 - Els prompts inicials amb ChatGPT i DeepSeek incloïen pseudocodi, com:
 1. Crear una baralla de cartes francesa (52 cartes).
 2. Barrejar la baralla de manera aleatòria.
 3. Assignar 5 cartes a cada jugador.

4. Cada jugador tria una carta (mostrada boca avall).
 5. Comparar les cartes seleccionades:
 - Determinar el guanyador segons el valor.
 - Retornar "Jugador 1 guanya", "Jugador 2 guanya" o "Empat".
- Amb Grok, els prompts eren més específics, incloent codi existent i sol·licitant correccions concretes (per exemple, "mostrar cartes boca avall amb back_black.svg i back_red.svg", "repartir 5 cartes aleatòries").
 - **Limitacions inicials:** ChatGPT i DeepSeek van generar codi funcional però amb errors, com repartir només 3 cartes o permetre seleccionar múltiples cartes. Grok va ser més efectiu per a correccions precises i millores avançades.

3. Tecnologies Utilitzades

El joc es va desenvolupar amb les següents tecnologies, algunes presents al codi inicial i altres suggerides o explicades per les IA:

- **Backend:**
 - **Node.js amb WebSocket (biblioteca ws):** Comunicació en temps real entre el servidor i els clients. ChatGPT va recomanar WebSockets, i Grok va explicar el seu funcionament (conexions persistents, esdeveniments com connection, message i close) per entendre la gestió de jugadors i missatges.
 - **JavaScript:** Lògica del servidor per gestionar connexions, seleccions de cartes i resultats.
- **Frontend:**
 - **HTML i CSS:** Estructura i estils de la interfície, amb suport per mostrar cartes com imatges SVG ({numero}_of_{palo}.svg) i cartes boca avall (back_black.svg, back_red.svg).
 - **JavaScript:** Lògica del client per gestionar la baralla, repartir cartes aleatòries, interactuar amb l'usuari i comunicar-se amb el servidor.
 - **Imatges SVG:** Representació visual de les cartes, emmagatzemades a la carpeta cartes/, amb noms com ace_of_spades.svg i cartes boca avall.
- **Investigació addicional:** No va ser necessari investigar tecnologies més enllà de les explicacions proporcionades per les IA:
 - ChatGPT va suggerir WebSockets i va proporcionar una configuració inicial per al servidor i el client.
 - Grok va explicar coses com implementar el mètode per barrejar les cartes de forma aleatòria amb l'algoritme Fisher-Yates (sort(() => Math.random() - 0.5)), evitant la necessitat de recerca externa.

4. Procés de Desenvolupament amb IA

4.1. Codi Inicial

El projecte va començar amb un codi base funcional proporcionat a través de prompts amb pseudocodi, que incloïa:

- server.js: Servidor WebSocket per gestionar connexions de dos jugadors, seleccions de cartes i resultats.

- index.html: Estructura bàsica de la interfície.
- styles.css: Estils per a les cartes i la interfície.
- client.js: Lògica del client, inicialment incompleta i amb una llista fixa de cartes.

Aquest codi permetia jugar, però tenia limitacions: usava una llista fixa de cartes (no aleatòria), no mostrava les cartes seleccionades al final de la ronda, i la interfície era molt bàsica.

4.2. Fases del Desenvolupament

4.2.1. Plantejament Inicial amb ChatGPT i DeepSeek

- **ChatGPT:**
 - Vaig utilitzar ChatGPT per definir l'enfocament del projecte, decidint utilitzar WebSockets per a la comunicació en temps real i una baralla de 52 cartes.
 - Va generar un codi inicial per a la lògica del joc (creació de la baralla, comparació de cartes) i la configuració del servidor WebSocket.
 - **Problemes:**
 - El codi inicial només repartia 3 cartes per jugador, en lloc de les 5 desitjades.
 - Permetia seleccionar múltiples cartes, provocant errors en la lògica del joc.
 - La integració amb el servidor era inestable, amb missatges que no es processaven correctament.
 - **Reformulacions:** Vaig haver de fer múltiples prompts per corregir aquests errors, com especificar el número correcte de cartes i limitar la selecció a una sola carta. Tot i això, les correccions eren lentes i sovint incomplertes.
- **DeepSeek:**
 - Vaig demanar a DeepSeek que generés l'estructura HTML (index.html) i part de la lògica del client (client.js) per mostrar cartes i gestionar la interacció.
 - **Problemes:** La interfície generada era molt bàsica (només contenedores simples sense estils avançats) i no suportava cartes boca avall.
 - **Reformulacions:** Vaig sol·licitar millores visuals, però DeepSeek no va entendre de cap manera el format de les cartes SVG ({numero}_of_{palo}.svg).

4.2.2. Millores amb Grok

Quan els resultats de ChatGPT i DeepSeek van mostrar limitacions, vaig passar a utilitzar Grok per a correccions i millores específiques. Les millores implementades inclouen:

1. Visualització de Cartes Seleccionades:

- **Requisit:** Mostrar les cartes seleccionades per ambdós jugadors al final de la ronda com imatges SVG.
- **Solució de Grok:** Va suggerir modificar server.js per enviar un missatge resultat|\${result}|\${carta1}|\${carta2} i actualitzar client.js per mostrar les cartes en els contenedors carta-seleccionada-jugador1 i carta-seleccionada-jugador2. També va proposar estils a styles.css per a les cartes seleccionades (80x120 píxels).
- **Reformulacions:** Inicialment, Grok va assumir que les cartes es mostraven amb text. Vaig aclarir que usaven imatges SVG boca avall (back_black.svg per al Jugador 1, back_red.svg per al Jugador 2), i la IA va ajustar client.js per utilitzar l'atribut dataset.carta i revelar les cartes al seleccionar-les.

- **Resultat:** Les cartes seleccionades es mostren correctament al final de cada ronda amb el format {numero}_of_{palo}.svg.

2. Format de Noms de Cartes:

- **Requisit:** Garantir que les cartes usessin el format {numero}_of_{palo}.svg (per exemple, ace_of_spades.svg).
- **Solució de Grok:** Va actualitzar la funció getCardValue a server.js per extreure el número de la carta (split per _of_) i va assegurar que client.js carregués les imatges SVG correctament.
- **Reformulacions:** Grok inicialment va usar un format incorrecte (ace_spades). Després d'especificar el format exacte, va corregir el codi per utilitzar _of_.
- **Resultat:** Totes les referències a cartes segueixen el format correcte.

3. Repartiment Aleatori de Cartes:

- **Requisit:** Repartir 5 cartes aleatòries per jugador en lloc d'una llista fixa.
- **Solució de Grok:** Va proposar crear una baralla completa de 52 cartes a client.js amb numbers i suits, i una funció dealCards per barrejar i repartir cartes. Inicialment va suggerir 4 cartes per jugador.
- **Reformulacions:** Vaig especificar que havien de ser 5 cartes, i Grok va canviar dealCards(4) a dealCards(5). La lògica de barajeo va funcionar correctament des del principi.
- **Resultat:** Cada jugador rep 5 cartes aleatòries al començament de cada ronda.

4.3. Iteracions i Reformulacions

- **Primers resultats de les IA:**
 - **ChatGPT:** Va generar un codi inicial funcional però amb errors (3 cartes en lloc de 5, selecció múltiple de cartes, integració inestable amb el servidor). Les correccions requerien múltiples prompts i eren lentes.
 - **DeepSeek:** Va produir una interfície HTML molt bàsica que no suportava cartes boca avall ni el format SVG correcte. Les millores visuals van ser difícils d'aconseguir.
 - **Grok:** Els codis inicials eren funcionals en un 80-90%. Les correccions (com el suport per a cartes boca avall i el format SVG) van ser ràpides i precises amb prompts específics.
- **Reformulacions de prompts:**
 - Amb ChatGPT i DeepSeek, vaig haver de reformular prompts repetidament per corregir errors com el nombre de cartes o la selecció múltiple, amb resultats lents.
 - Amb Grok, vaig especificar:
 - Cartes boca avall amb back_black.svg i back_red.svg.
 - Format exacte {numero}_of_{palo}.svg.
 - Repartiment de 5 cartes aleatòries en lloc de 4.
 - Exemple de prompt a Grok: "Modifica client.js per repartir 5 cartes aleatòries per jugador, mostrant-les boca avall amb back_black.svg per al Jugador 1 i back_red.svg per al Jugador 2, i revela la carta seleccionada amb el format {numero}_of_{palo}.svg."
- **Desafíos:**

- **ChatGPT i DeepSeek:** Errors com repartir poques cartes, permetre múltiples seleccions i mal enteniment del format SVG van ser difícils de corregir, ja que les IA no interpretaven bé els requeriments específics.
- **Grok:** El principal repte va ser assegurar que les cartes es mostressin boca avall i es revelessin correctament. Grok ho va resoldre amb l'atribut `dataset.carta`. També va afegir validació al servidor (`isValidCard`) per evitar cartes invàlides.
- **Comparació amb correcció manual:** Alguns errors (com limitar la selecció a una carta) haurien estat més ràpids de corregir manualment, però l'ús de la IA va permetre explorar solucions ràpides i entendre millor tecnologies com WebSockets.

5. Tecnologies Investigades

- **WebSockets:**
 - ChatGPT va recomanar WebSockets i va proporcionar una configuració inicial per a `server.js` i `client.js`, però amb errors.
 - Grok va explicar clarament el funcionament dels WebSockets (conexions bidireccionals persistents, esdeveniments com `connection`, `message`, `close`), evitant la necessitat de recerca externa.
- **Imatges SVG:**
 - DeepSeek va generar una interfície HTML bàsica, però no va gestionar bé les imatges SVG.
 - Grok va explicar com carregar SVG amb `` i `object-fit: contain`, i com canviar dinàmicament les imatges per mostrar cartes boca avall i revelar-les.
- **Barajeo aleatori:**
 - ChatGPT i Grok van suggerir l'algoritme Fisher-Yates (`sort() => Math.random() - 0.5`) per barrejar la baralla, que va funcionar sense necessitat d'investigació addicional.

6. Resultats i Reflexions

- **Funcionament:** El joc és completament jugable:
 - Dos jugadors es connecten via WebSocket.
 - Cada jugador rep 5 cartes aleatòries, mostrades boca avall (`back_black.svg` o `back_red.svg`).
 - En seleccionar una carta, aquesta es revela (per exemple, `ace_of_spades.svg`).
 - Al final de la ronda, es mostren les cartes seleccionades i el resultat (per exemple, "Jugador 1 guanya!").
 - Les rondes es reinicien amb noves cartes aleatòries després que ambdós jugadors voten.
- **Contribució de les IA:**
 - **ChatGPT:** Va establir les bases del joc, però amb errors que van requerir moltes iteracions.
 - **DeepSeek:** Va proporcionar una interfície HTML inicial, però limitada i poc estilitzada.
 - **Grok:** Va ser clau per corregir errors, implementar millores (visualització de cartes seleccionades, format SVG, repartiment de 5 cartes).

- **Leccions apreses:**
 - **Prompts clars:** És essencial especificar detalls precisos (per exemple, format SVG, cartes boca avall, 5 cartes) per evitar malentesos.
 - **Limitacions de la IA:** ChatGPT i DeepSeek van tenir problemes amb correccions específiques, mentre que Grok va ser més efectiva amb prompts detallats.
 - **Documentació contínua:** Prendre notes durant les interaccions amb la IA va facilitar la creació d'aquesta documentació.
 - **Coneixement previ:** El coneixement bàsic d'HTML, CSS i JavaScript va ajudar a entendre i ajustar el codi generat, especialment quan les IA no interpretaven correctament els requeriments.
 - **WebSockets:** La tecnologia, suggerida per ChatGPT i explicada per Grok, va ser fàcil d'implementar gràcies a les seves guies, tot i ser relativament nova per a mi.
 - **Punts febles de la IA:** El disseny visual de la pàgina. No hi ha manera de conseguir un resultat bonic, ja que per això fa falta més interacció humana per crear una pàgina amb un fons bonic i que quadri amb la temàtica.

7. Conclusió

El projecte "Carta més alta" es va desenvolupar amb èxit utilitzant ChatGPT, DeepSeek i Grok, complint el requisit d'apostar per eines d'IA. La combinació de WebSockets, JavaScript, HTML, CSS i SVG va permetre crear un joc interactiu i visualment atractiu. Les iteracions amb les IA van ser essencials per superar limitacions inicials (com errors en el repartiment de cartes o la gestió de WebSockets) i implementar millores com el repartiment aleatori de 5 cartes i la visualització de les seleccions finals. Aquest procés va demostrar el potencial de la IA per accelerar el desenvolupament, però també la importància de prompts clars i, en alguns casos, intervenció manual per a correccions específiques. La documentació contínua durant el procés va ser clau per estructurar aquesta presentació i reflexionar sobre el camí recorregut.