

chkn

CHKN (Chicken Race) pa 'chkn.sputnet.space' ar ett real-time spel dar du kan utmana vänner i antingen ett chicken run (vadslagning) eller en 5-kamp. Servern ar alltid domare och ar den enda sanningen for poang och state.

Om man spelar ensam far man mota en AI-bot som heter 'Sputnik' (samma bot som i Yatzy). Login och user-hantering sker via Authentik. Backend byggs i Node och frontend i Vite, samma stack och upplagg som Yatzy.

****Spelare****

Match = 2-N spelare, rekommendation 2-6. Allt ar real-time och servern styr flodet (authoritative server).

****Valuta****

Allt ackumuleras i en gemensam heltalsvaluta: 'CHKN-poang'.

0) Lobby

1. Skapa match (invite-länk/kod)
2. Valj lage: Chicken Run eller 5-kamp
3. Ready-check -> start

5-kamp: spelordning

1. Yatzy
2. Black Jack
3. Fragesport
4. Musikquiz
5. Texas Hold'em

1) Yatzy (finns redan)

Output: 'yatzyScore' (ex 0-375 beroende pa regler).

Konvertering till CHKN-poang:

Startstack efter Yatzy = 'yatzyScore * 10'

Spara: 'stack = yatzyScore * 10'

2) Black Jack (byggs)

Malt: skapa swing + taktik men snabbt.

Upplagg (MVP-vanligt):

- Du har 5 rundor.
- En runda: du kan betta pa 0-7 rutor (klassisk table layout).
- Bet per ruta: 10-100.
- Varje aktiv ruta ar en separat hand mot dealer.
- Efter 5 rundor: summera 'bj_delta' (netto vinst/forlust).
- Uppdatera stack: 'stack = stack + bj_delta'.

Regelknappar per hand: Hit / Stand / Double / Split (valfritt i MVP). Insurance kan hoppas over forst.

****Viktigt for roulette-steget****

Definiera for blackjack:

- 'bj_start = yatzyScore * 10'
- 'bj_end = stack_after_bj'
- 'bj_profit = max(0, bj_end - bj_start)'
- 'bj_loss = max(0, bj_start - bj_end)'

Mellanspel A) Roulette (mellan 2 och 3)

Bara blackjack-vinst far riskas.

Spelaren valjer:

- Farg: Rod eller Svart
- Belopp: 'bet <= bj_profit'

Utfall:

- Vinst: 'stack += bet'
- Forlust: 'stack -= bet'

Obs: las 'bj_profit' har som "roulette-tillaten pott" sa det inte gar att loop-maxa.

3) Fragesport (AI-fragor)

****Kategorival****

- 6 kategorier totalt.
- Round-robin: varje spelare valjer 2 kategorier var.
- Totalt blir det '2 * antal_spelare' kategori-val.

****Fragepaket****

- Generera fragor per valt kategori (ex 5 fragor per val).
- Svartighetsmix per kategori-batch: 20% latt, 60% medel, 20% svar.

****Poangmodell (forslag)****

Servern startar varje fraga med 'startTimestamp' och tar emot svar med 'serverReceivedTimestamp'.

Baspoang:

- Latt: 80
- Medel: 140
- Svar: 220

Tidsbonus:

'bonus = round(bas * clamp(1 - (t / T), 0, 1) * 0.75)'

dar 't' ar sekunder till korrekt svar och 'T' ar maxtid (ex 12s).

Totalpoang:

'points = bas + bonus'

Fel svar: 0 (ingen minus i MVP).

Uppdatera: 'stack += trivia_points'

Mellanspel B) Tarning hogre/lagre (mellan 3 och 4)

- Sla tarning 1 (1-6) synlig.
- Spelaren valjer hogre eller lagre.
- Insats 'bet <= stack'. Satt garna max, ex 25% av stack.
- Sla tarning 2.

Utfall:

- Vinst: 'stack += bet'
- Forlust: 'stack -= bet'
- Vid lika: push (0) rekommenderas.

4) Musikquiz

- 5 rundor.
- Spela ett klipp per rond (10-20s eller progressivt langre).
- Forst korrekt far poang.

Poangforslag:

- Grund: 200
- Tidsbonus likt trivia, men kortare 'T' (ex 8s)

Alternativt (om flera far gissa samtidigt):

- 1a: 200
- 2a: 100
- Ovriga: 0

Uppdatera: 'stack += music_points'

Praktiskt: tank pa rattigheter. Spotify previews om de finns, annars royalty-free bibliotek.

5) Texas Hold'em

- Spelarna startar med sin ackumulerade 'stack'.
- Turnering "turbo".
- En niva = alla har varit SB och BB exakt en gang (en orbit).
- Efter varje orbit: blinds dubblas.
- Spelet slutar nar en spelare har allt.

Startblinds:

- 'SB = max(10, round(medianStack / 200))'
- 'BB = 2 * SB'

Single-player: Sputnik

- 'Sputnik' ar en vanlig 'MatchPlayer' med 'isBot=true' och 'userId="bot:sputnik"'.
- Om match startar med 1 manniska -> servern auto-joinar Sputnik.
- Sputnik bypassar Authentik (intern identitet).

Bot-beteende (enkel men kul):

- Yatzy: ateranvand befintlig botlogik.
- Blackjack: basic strategy + liten risk-justering.
- Trivia: sannolikhet att svara ratt baserat pa svarighet + reaktionstid.
- Musikquiz: slumpad "igenkanning" med delay.
- Hold'em: tight-aggressive med randomness.

Datamodel (sa ni slipper 17 olika poang-sanningar)

Korna en 'match-orchestrator' som ager state och bara lanar UI fran varje minispel.

Core:

- 'Match { id, mode, status, createdAt }'
- 'MatchPlayer { matchId, userId, seat, stack, isConnected }'
- 'StageState { matchId, stage, stateJson, startedAt }'
- 'LedgerEntry { matchId, userId, stage, delta, reason, ts }'

Event-sourcing light (rekommenderat):

- Spara events: 'AnswerSubmitted', 'BetPlaced', 'HandResolved', 'StageCompleted'.
- Rebuild state ska vara deterministiskt.

Realtime & anti-fusk (MVP men stabilt)

- Servern ar authoritative.
- Servern skickar 'question shown at X'.
- Servern tar emot svar + timestamp (server time).
- Klienter far inte rakna poang.
- Rate limit pa svar/guess.
- Lasta stage transitions.

MVP-plan (i rat ordning)

1. Match Orchestrator + Lobby + Realtime
2. Integrera Yatzy-resultat -> 'stack'
3. Blackjack (5 rundor, bets 10-100, 1-7 rutor)
4. Roulette rod/svart pa 'bj_profit'
5. Trivia (kategori-val + AI-generator + snabbpoang)
6. Tarning hi/lo
7. Musikquiz (forst korrekt)
8. Hold'em turbo (orbit -> blinds * 2)

Drift

- Doman: 'chkn.sputnet.space'
- Auth: Authentik
- Backend: Node (samma upplagg som Yatzy)
- Frontend: Vite (samma upplagg som Yatzy)
- Proxy route: '/' -> Vite build (statisk)
- Proxy route: '/api' -> Node API
- Proxy route: '/socket' -> WebSocket upgrade