

# chkn

CHKN (Chicken Race) på 'chkn.sputnet.space' är ett real-time spel där du kan utmana vänner i antingen ett chicken run (vadslagning) eller en 5-kamp. Servern är alltid domare och är den enda sanningen för poäng och state.

Om man spelar ensam far man mot 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 är 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 på regler).

Konvertering till CHKN-poang:

Startstack efter Yatzy = 'yatzyScore \* 10'

Spara: 'stack = yatzyScore \* 10'

## 2) Black Jack (byggs)

Mål: skapa swing + taktik men snabbt.

Upplegg (MVP-vanligt):

- Du har 5 runder.
- En runda: du kan betta på 0-7 rutor (klassisk table layout).
- Bet per ruta: 10-100.
- Varje aktiv ruta är en separat hand mot dealer.
- Efter 5 runder: summa 'bj\_delta' (netto vinst/forlust).
- Uppdatera stack: 'stack = stack + bj\_delta'.

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

\*\*Viktigt för roulette-steget\*\*

Definiera för 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 får riskas.

Spelaren väljer:

- Färg: Röd eller Svart
- Belopp: 'bet <= bj\_profit'

Utfall:

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

Obs: las 'bj\_profit' har som "roulette-tillaten pott" så det inte gar att loop-maxa.

## 3) Fragesport (AI-fragor)

\*\*Kategorival\*\*

- 6 kategorier totalt.
- Round-robin: varje spelare väljer 2 kategorier var.
- Totalt blir det '2 \* antal\_spelare' kategori-val.

\*\*Fragepaket\*\*

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

\*\*Poängmodell (forslag)\*\*

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

Baspoäng:

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

Tidsbonus:

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

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

Totalpoäng:

'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 väljer hogre eller lagre.
- Insats 'bet <= stack'. Satt gärna max, ex 25% av stack.
- Sla tarning 2.

Utfall:

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

## 4) Musikquiz

- 5 runder.
- Spela ett klipp per rond (10-20s eller progressivt längre).
- Forst korrekt får poäng.

Poängforslag:

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

Alternativt (om flera får gissa samtidigt):

- 1a: 200
- 2a: 100
- Övriga: 0

Uppdatera: 'stack += music\_points'

Praktiskt: tank på rättigheter. 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 gång (en orbit).
- Efter varje orbit: blinds dubblas.
- Spelet slutar när en spelare har allt.

Startblinds:

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

## Single-player: Sputnik

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

Bot-beteende (enkel men kul):

- Yatzy: ateranvänd befintlig botlogik.
- Blackjack: basic strategy + liten risk-justering.
- Trivia: sannolikhet att svara rätt baserat på svarighet + reaktionstid.
- Musikquiz: slumpad "igenkänning" med delay.
- Hold'em: tight-aggressive med randomness.

## Datamodell (sa ni slipper 17 olika poäng-sanningar)

Korna en 'match-orchestrator' som agerar state och bara lanar UI från 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 är authoritative.
- Servern skickar 'question shown at X'.
- Servern tar emot svar + timestamp (server time).

Klienter får inte räkna poäng.

- Rate limit på svar/guess.

- Lasta stage transitions.

## MVP-plan (i rat ordning)

1. Match Orchestrator + Lobby + Realtime
2. Integrera Yatzy-resultat -> 'stack'
3. Blackjack (5 runder, bets 10-100, 1-7 rutor)
4. Roulette rod/svart pa 'bj\_profit'
5. Trivia (kategori-val + AI-generator + snabbpoäng)
6. Tarning hi/lo
7. Musikquiz (först 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