

Developer A

Sprint1

Developer B

Lav en feature branch: "continue_saved_game_[par]"

I toppen af `startSession` metoden, indsæt en linje der viser en velkomst til brugeren, fx: "Velkommen til Matador".

Anvend `TextUI` til dette.

Hvis der findes data, er det forventede output (nogenlunde):

Egon, 100000

Kjeld, 50

Benny, 3030

Hvis der ikke findes data, vil vi skal brugeren spørges om han vil køre med denne data.

- Kig på if-statementet i metoden og kopier dette ind det rigtige sted:

```
&& ui.promptBinary("Continue previously saved game? Y/N")
```

Kod metoden `promptBinary` i næste trin.

Vi mangler `promptBinary` i `TextUI` klassen. En metode der gør som `promptText` og `promptNumeric`, men i stedet for en `String` eller en `int`, returnerer en `boolean`.

1. Vis beskeden til brugeren (beskeden findes i metodens parameter)
2. Brug `nextLine` til at få fat i det brugeren taster
3. Brug et if-else statement til at finde ud af om brugeren har tastet "Y" eller "N".
4. Hvis brugeren ikke har tastet nogen af delene, kaldes metoden rekursivt.
5. Hvis brugeren har tastet "Y" eller "y" returneres `true`
6. Hvis brugeren har tastet "N" eller "n" returneres `false`

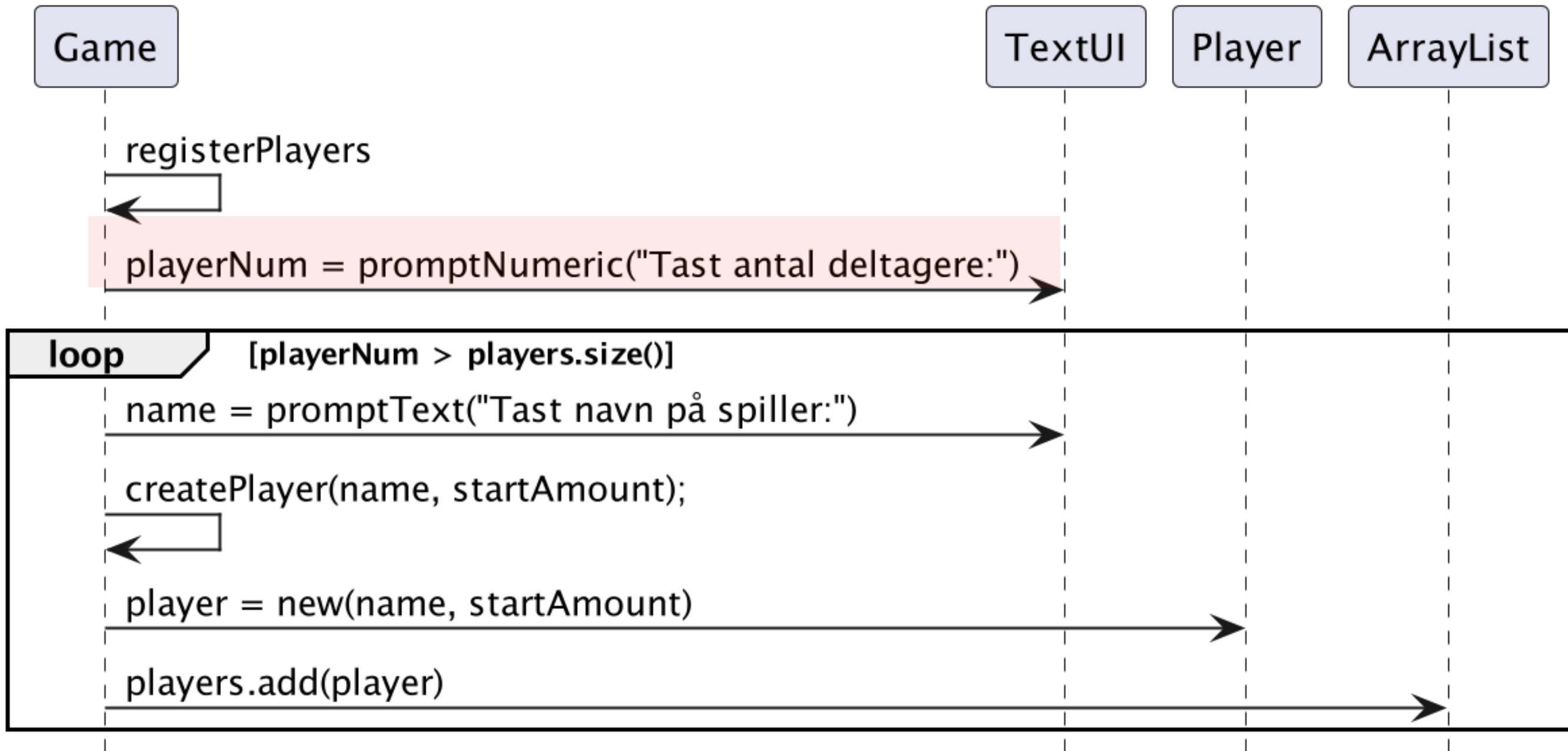
Git:



Git:



UC2: registerPlayers



Developer A

Sprint2

Developer B

Opret en branch: "UI_improvement_[par]"

Vi skal nu forbedre brugerdialogen i `registerPlayers`, ved at bede om antallet af spillere før vi går ind i loop'et. Støt jer til sekvensdiagrammet for uc2

- Uden for `while`-loop'et, brug en `TextUI` metode til prompte brugeren for antal spillere.
- Anvend brugerens svar i `while`-loop'et, sådan at det kører indtil det ønskede antal spillere er nået.

Nu skal vi håndtere at brugeren måske taster et ulovligt antal spillere.

- Lige før `while`-loop'et. Tjek at brugeren har tastet et tal mellem 2 (minimum antal spillere) og 6 (maximum antal spillere).
- Kald metoden rekursivt hvis det **ikke** er tilfældet.

1. Randomiser listen med spillere

Hint:

Google hvordan man let kan gøre det med lister i Java

2. Tilføj en `currentPlayer` attribut til `Game` klassen

3. Tilføj en getter til `Player` klassen der returnerer en spillers navn.

I næste trin tester vi randomiseringen af spillerlisten.

På samme tid, kommer vi et skridt nærmere koden til et turbaseret spil.

1. Tilføj metode med signaturen, `void runGameLoop()` til `Game` klassen.

2. I `runGameLoop()`, tildel første element i `players` til `currentPlayer` attributten.

3. Print navnet på `currentPlayer`. Gerne i en sætning der fortæller hvis tur det er.

4. Kald `runGameLoop()` fra `main`.

Forventet output:

Det er Benny's tur

Kør koden nogle gange for at sikre at listen blevet randomiseret.