

# Projekt 4 - The Ludo Game

## User Stories:

- Som användare, ska jag ha möjlighet till att spara befintligt spel och sedan återuppta spelet vid ett senare tillfälle.
- Som användare, ska jag kunna flytta min pjäs och vid situationer där jag har fler pjäser som går att flytta ska jag kunna välja vilken pjäs jag vill flytta.
- Som användare, vid situationer där jag har en pjäs ute på planen och slår en 1:a eller 6:a ska jag ha möjlighet till att välja mellan att flytta på pjäsen ute på spelplanen eller ta ut en ny från boet.
- Som användare, ska jag ha möjlighet att kolla utgången på tidigare avslutade spel.
- Som användare, ska jag ha möjligheten att slå ut andra spelares pjäser när jag går på samma ruta.

## Struktur för programmet:

### Klasser:

- Game
  - GameID- int
  - PlayerTurn - int
  - GameFinished - bool
  - V one-to-many koppling
- Player
  - GameID - int
  - TeamID - int
  - playerName- string
  - playerColor - string
  - Lista med tokens
  - V one-to-many koppling
- Token
  - PlayerID- int
  - TokenID - int
  - PositionOnGameBoard - int (räknar steg)
  - PositionOnGoalLine - int(räkna steg 1-5)
  - InGoalLine - bool
  - InGoal - bool

## Game engine

### Klassernas ansvarsområden och metoder

#### Engine-Metoder

ThrowDie(Player) = kasta tärningen och kolla vilka pjäser som går att flytta, returnera resultat till spelappen.

MoveTokenAction(Token valdToken, int Die)

- Ändra valdToken.InNest = false;
- Flytta tokens steg med resultatet från die
- Kolla om token har gått i mål

UpdateGame(Game, Player)

- Kolla om spelaren har vunnit
- Uppdatera vems tur det är

SaveGame()

LoadGame()

#### Game:

Ansvar:

- Vems tur det är
- Hålla koll om någon har vunnit
- Skapa lag och spelare

Metoder:

[Vid start av nytt spel:](#)

CreatePlayers() = Ange hur många spelare som ska vara med i matchen, 2-4 spelare

ChooseColor() = Välj vilken färg man ska vara, röd, grön, blå, gul

ChooseName() = Ange namn på lagen/spelarna

[Vi laddning av spelet](#)

LoadGame() = Hämtar hem data från ett tidigare spel

In game

CheckTurn() = Kolla vems tur det är, Random i början av matchen och sedan följer det ett mönster

CheckWinner() = Kolla om någon har vunnit

AutoSave() = Spara spelet efter varje runda

LogWinner() = spara avslutat spel och presentera vinnaren.

## Grupp 5

Utökningar

CreateBoard() = skapar ett visuellt bräde

### Player:

Ansvar:

- Slå tärningen
- Välj vilken pjäs som ska flyttas
- Kolla ifall spelaren vill avsluta och spara spelet.

In game:

- ThrowDie()
- ChooseToken() = Om tärningen är 1 eller 6, kolla om det finns pjäser i boet. Kollar sedan om det finns pjäser på planen. Skriv ut vilka pjäser som spelaren kan flytta på och spelaren väljer en utav de möjliga pjäserna.
- SaveGame()

### Token:

Ansvar:

- Flytta på tokens
- Kolla om det har gått i mål

Metoder:

Ingame:

- MoveToken() = Flyttar den valda pjäsen efter utgångsvärdet från ThrowDice()
- HasFinished() = Returnerar true eller false ifall pjäsen är i mål eller inte

Utökning:

- Lägga in "VictoryLap" som är slut spurten för alla pjäser
- Lägga in en "KnockOut" funktion som ska kunna slå tillbaka andra spelare pjäser.

Flowchart:

## Grupp 5

