

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:

Regler:

- För att spelaren ska ha möjlighet till att flytta ut en pjäs från boet, behöver hen slå 1:a eller 6:a med tärningen
- Om spelaren har fler än en pjäs på brädet får spelaren själv välja vilken pjäs hen vill flytta under sin tur.
- För att en pjäs ska gå i mål behöver spelaren slå jämt, t ex om spelarens pjäs står på en ruta från mål och slår en trea behöver spelaren gå in i mål och sedan backa resterande steg.
- Om en pjäs har gått i mål tas den ut ur spel.
- Det får enbart finnas stå en pjäs på en spelruta, om en pjäs skulle hamna på en "ockuperad" spelruta knuffar den nya pjäsen ut den tidigare pjäsen och den nya pjäsen "ockuperar" spelrutan medan den tidigare pjäsen får börja om från start.
- Spelets vinnare är den spelare som har lyckats flytta alla sina pjäser i mål först.

Klasser:

- Game
 - GameID- int
 - GameName - string
 - LastSaved - DateTime
 - PlayerTurn - int
 - Round
 - GameFinished - bool
 - Players - List<Player>

- Player
 - GameID - int
 - TeamID - int
 - playerName- string
 - playerColor - string
 - Winner - bool
 - Tokens - List<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

Klassernas metoder

GameEngine

ThrowDie() = Simulerar ett tärningskast

TokensToMove() = Returnerar en lista med pjäser som spelaren kan flytta på, den valda token ska i sin tur skickas in i RunMovmentAction metoden.

RunMovmentAction(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 och uppdaterar vilken runda det är

SaveGame()

LoadGame()

Game:

Metoder:

CreatePlayer() = Skapar spelare och fyra tokens till spelaren

CheckTurn() = Kolla vems tur det är

CheckWinner() = Kolla om någon har vunnit

Todo-list

- Metod som sköter "KnockOutToken" fasen
- Fler tester, tester, tester

Flowchart:

