

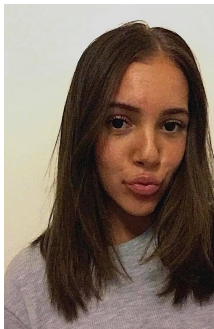


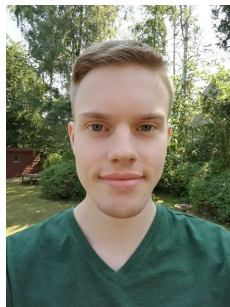




62531
Gruppe 44
Udviklingsmetoder til it-systemer

3-ugers

Mike Helt Jensen, s205865			Simon Schnack Jørgensen, s195822
Oline Melinda Andersen s201010			Steven Gordon Pedersen s205456
Katrine Amalie Harris Holst, s196509			Christian Proudfoot s205465

DTU Diplom

Softwareteknologi

[https://github.com/KatsCoding/44_final]

Abstract:	3
Timeregnskab	3
Kapitel 1- Indledning	4
Kapitel 2 - Krav	5
Kravliste:	5
Spillet:	5
Spilleregler:	5
Krav til hardware/software hos brugeren:	5
Prioriteret kravliste:	6
Kapitel 3 - Analyse	6
Use-cases:	6
Use case diagram:	7
Use case beskrivelser:	8
Use Case: Land on field	9
Use Case: Play turn	10
Domænemodel:	11
System Sekvensdiagram:	12
Kapitel 4 - Design	14
Klassediagram:	14
Sekvensdiagram:	15
Køb Hus	15
Kapitel 5 - Implementering	16
Kapitel 6 - Test	20
Dokumentation for gennemført test	20
Vurdering af systemets status/kvalitetsmæssigt	21
Kapitel 7 - Delkonklusion	22
Features:	22
Kapitel 8 - Konklusion - Forbedringer til fremtiden	23
Auktion	24
9. Litteraturliste / Bilag:	25
bilag 1: Spillepladen	25
Bilag 2.1 : Ejendomskort	26
Bilag 2.2 : Ejendomskort	26
Bilag 2.3 : Ejendomskort	27
Bilag 3.1 : Regelsæt	28
Bilag 3.2 : Regelsæt	29
Bilag 3.3 : Regelsæt	30
Bilag 3.4 : Regelsæt	31
Bilag 4.1 : Lykkekort	32
Bilag 4.2 : Lykkekort	33

Abstract:

We were given the assignment to make a monopoly game as our final exam project. We have been given some specifications for the game, as well as a description of how the client wants the finished product to function. The game's purpose is to allow 2-6 players to face each other in a monopoly game, which contains the functions of buying, selling and executing specific actions. To win the game, you must be the last player left when all other players go bankrupt. The customer has set some terms of functionality, and let it be up to us as developers to prioritize the work so the key functions will be a part of the finished product. We have simplified the process through analysis followed by coming up with a design on how we are going to complete the game, so the customer easily can understand what we are doing. We will also implement, debug and test our code, which we will accompany with completed documentation. Throughout our work we came to the conclusion of a successful monopoly game, that meets the key requirements set by the customer and project manager.¹

Timeregnskab

Rapport tid er lig med: hvor lang tid man bruger på selve rapporten, hjælper andre eller laver dialog med andre fra gruppen angående rapporten.

Kode tid er lig med: hvor lang tid man har brugt på at skrive koden, undersøge, hjælpe andre og lave dialog med andre fra gruppen angående koden.

Dialog tid: er hvor vi har lavet dialog i forhold til hvad og hvordan projektet skal laves, altså planlægning af projektet, med møder osv.

* Katrine har ikke haft talt dialogtid overhovedet, og har bare gættet.

Student	Katrine s196509	Mike s205865	Oline s201010	Simon s666666	Steven Gordon	Steven s205465
Tid på rapport en	10,5 timer	20ca. timer	20 timer ca.	10 timer.	15 timer	16 timer
Tid på koden	156 timer	140 timer	45 timer ca.	56 timer.	44. timer	12 timer
Dialog tid	17*	35-40ca. timer	20 timer ca.	22	20 timer	20 timer
Tid i alt	183,5	195-200ca.	85 timer ca.	88	79 timer	48 timer

¹ Dele af abstract taget fra CDIO 3, med ændringer så den passer denne proces

Kapitel 1- Indledning

I faget indledende programmering har vi som udviklere fået en eksamensopgave om at udvikle et system der udfører et matador spil. Vi skal igennem vores arbejde ende med et monopoly spil som kan spilles af 2-6 spillere, hvor der igennem terningekast landes på nogle felter på en plade som har forskellige konsekvenser. Udover dette skal spillerne have mulighed for at købe og sælge ejendomme. Spillet vindes ved at være den sidste spiller tilbage når alle andre medspillere er gået fallit. I denne opgave er der stillet nogle krav til spillet, som vi som udviklere skal tilfredsstille for at opnå et succesfuldt slutprodukt, samtidig med at tilstræbe kundens yderst tilfredshed. Vi vil udføre denne opgave ved først at opstille scenariet, og kigge på hvordan spillet skal forløbe og opstilles vha. use cases, domænemodel, systemsekvensdiagram mm. Alt dette vil vi gøre med kundes vision som drivkraft. Herefter vil vi ud fra vores analyse forsøge at programmere spillet vha. nyttige metoder, datastrukturer, klasser, nedrivning mm. Til sidst vil vi debugge og teste koden, så vi med sikkerhed ved at spillet fungerer som det skal og lever op til de krav stillet i starten af projektet. Formålet med dette projekt er at vi som udviklere får mulighed for at bruge den viden vi har lært gennem semestret, og gå igennem de mange forskellige trin der er inden for at udvikle et færdigt system, såsom det ville foregå i den virkelige verden. Denne opgave er en forlængelse af tidligere projekter og det er derfor ligeledes formålet at kunne arbejde videre, optimere, bygge ovenpå med ny viden og ændre fremgangsmåder baseret på konstruktiv kritik. Udover dette tester, styrker og træner det vores evne til at arbejde sammen med andre udviklere, som vi er afhængige af at kunne evne ude på arbejdsmarkedet. Resultatet af vores projekt skal bruges til læring og udvikling som software ingeniører.²

² Dele af indledning taget fra CDIO 3, med ændringer så den passer denne proces

Kapitel 2 - Krav

Kravliste:

Spillet:

- to terninger
- et brætspil
- 6 biler
- huse og hoteller
- skøder
- prøv lykken kort
- penge

Spilleregler:

- Alle spillere starter på start
- Hvis man får to ens øjne på terningen må man slå igen, hvis man i samme tur slår tre gange ens øjne flytter man sig ikke på tredje slag men ryger direkte i fængsel.
- Alle spillere modtager 30.000 kr
- Det øjnene viser det antal skal spilleren rykke
- Skal kunne købe huse og hoteller men uden at kun ligge alle husene eller hotellet på et skøde, det skal være ligeligt fordelt.
- Modtage 4000 kr for passering af start.
- Hvis en spiller ikke vælger at købe et skøde skal skødet aktioneres
- Hvis en spiller går fallit altså skylder mere end han ejer skal alt afleveres til banken og spilleren er ude.
- Hvis man ejer alle skøder af samme slags, er lejen dobbelt på de grunde og man kan begynde at bygge huse og derefter hoteller.
- Hvis man ryger i fængsel, kan man komme ude hvis man slår to ens øjen værdier på terningen, betale 1000 kr, eller at der går tre omgange.
- Man modtager 4000 kr hvis man passere start medmindre man ryger i fængsel eller prøvlykken kortet siger du ikke får dem.
- Skal kunne bytte laver handler.
- Når folk lander på andres felt, skal der betales leje til ejeren af skødet.

Krav til hardware/software hos brugeren:

- Brugeren skal have en computer med en Java 9 eller nyere version. Dette har Windows, Mac, og Linux hvis den nyeste version af Java er installeret.

Prioriteret kravliste:

SKAL udføres:

- To terninger
- Et brætspil
- 40 felter
- 6 biler
- Alle spillere starter på start
- Spilleren flytter så mange felter øjnene på terningen viser
- Starter ud med at have 30000kr når spillet start
- Modtager 4000kr når man passere start(undtagen når man ryger i fængsel)
- Prøv lykken kort
- Indkomstskat
- Ekstra kast
- Køb og leje
- Huse og hoteller
- Fængsel
- Skøder
- Indtast spillernavne
- Random spiller starter eller den første der indtaster navn starter

Udføres **HVIS** tid bagefter alt det andet:

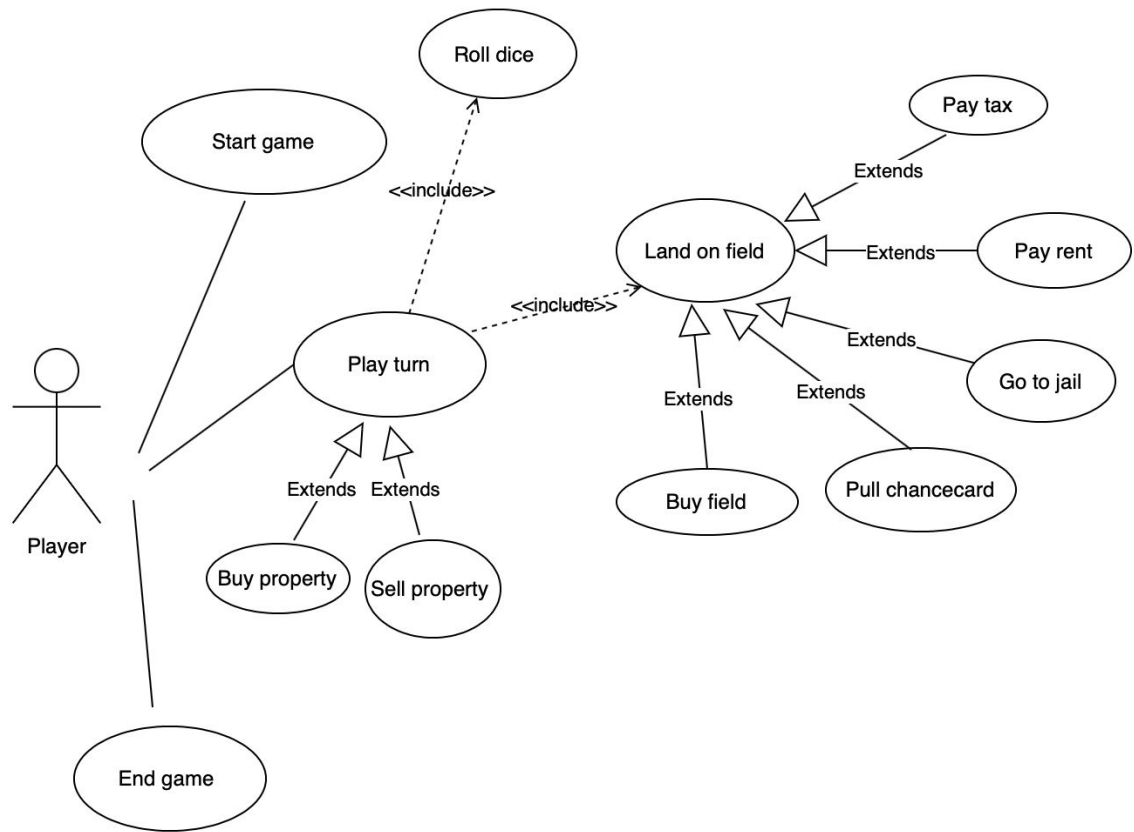
- Aktion
- Pantsætning
- Hurtigt spil
- "Glemmer" reglen
- "Fallit" reglen

Kapitel 3 - Analyse

Use-cases:

- Play turn
- Buy field
- Buy properties
- Go to jail
- Roll dice
- Change properties
- Pull chance card
- End game
- Start game
- Sell properties
- Pay tax
- Pay rent
- Pay income tax

Use case diagram:



Use case beskrivelser:

Breif beskrivelser:

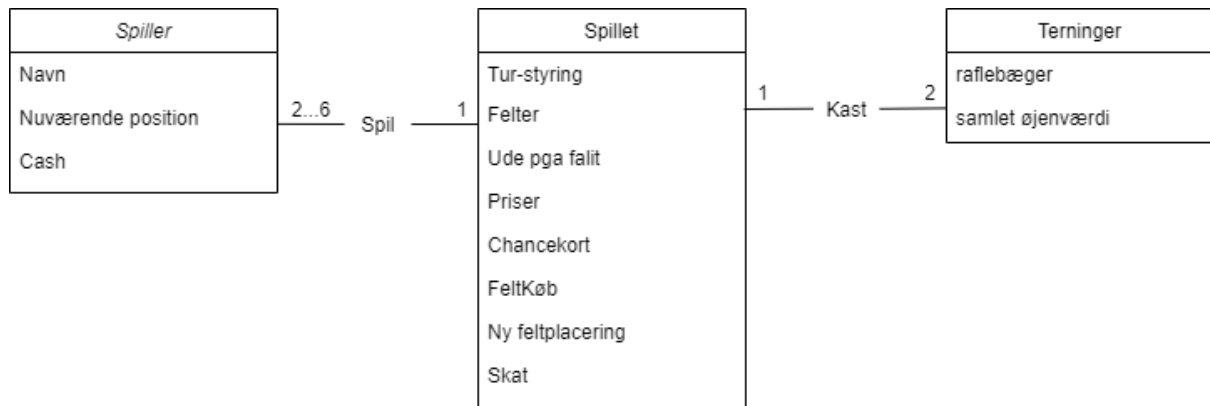
Use cases	Brief beskrivelser
Start game	<i>Spillerne starter spillet ved at angive navn</i>
Land on field	<i>Spilleren lander på et felt efter terningekast, hvorefter en handling udføres</i>
Play turn	<i>Spilleren skal uanset hvad kaste terningen, men kan udover dette vælge at købe eller sælge ejendomme</i>
Buy field	<i>Spilleren lander på et felt der er ledigt, og køber feltet</i>
Buy property	<i>Spilleren ejer en grund, og kan enten købe et hus eller hotel på denne grund {Hotel kan købes HVIS 5 huse ejes på grunden}</i>
Go to jail	<i>Spilleren lander på feltet "De fængsles", bliver rykket til fængsel feltet og modtager ikke 4000 kr ved at passere start.</i>
Roll dice	<i>Spillerne kaster på skift to terninger og rykker så mange felter frem som øjnene viser.</i>
Pull chance card	<i>Spilleren lander på feltet "Prøv lykken" og trækker et kort, hvor en konsekvens udføres</i>
End game	<i>Spilleren trykker afslut og forlader spillet</i>
Sell property	<i>Spilleren har tur og sælger en af sine ejede ejendomme</i>
Pay tax	<i>Spilleren lander på feltet "Betal indkomstskat" og skal betale 4000 kr eller 10 % af sine værdier eller spilleren lander på et ejet felt uden grunde, og skal betale skat til ejeren.</i>
Pay rent	<i>Spilleren lander på et ejet felt der har grunde, og betaler leje til ejeren</i>

Fully dressed beskrivelser:

Use Case: Land on field
ID: 1
Brief description: Spilleren lander på et felt efter terningekast, hvorefter en handling udføres.
Aktører: <ul style="list-style-type: none">• Spiller• System
Pre conditions: <ul style="list-style-type: none">• Spillet er startet• En spiller har kastet terningen
Main flow: <ol style="list-style-type: none">1. Use casen starter når spilleren har kastet terningen2. Spilleren lander på et bestemt felt3. Der startes et loop til en af de extended use cases {Pay tax, Pay rent, Go to jail, Pull chancecard, Buy property}, som udføres efter terningekast, og derefter bliver bragt tilbage til play turn, evt ved en ny spiller.
Post conditions: <ul style="list-style-type: none">• Use casen starter forfra ved ny spiller
Alternative flow: <ol style="list-style-type: none">3a. Tiden for spillet udløber, loopet stopper og en spiller vinder.3b. Alle spillere undtagen 1 er gået fallit, loopet stopper og en spiller vinder.

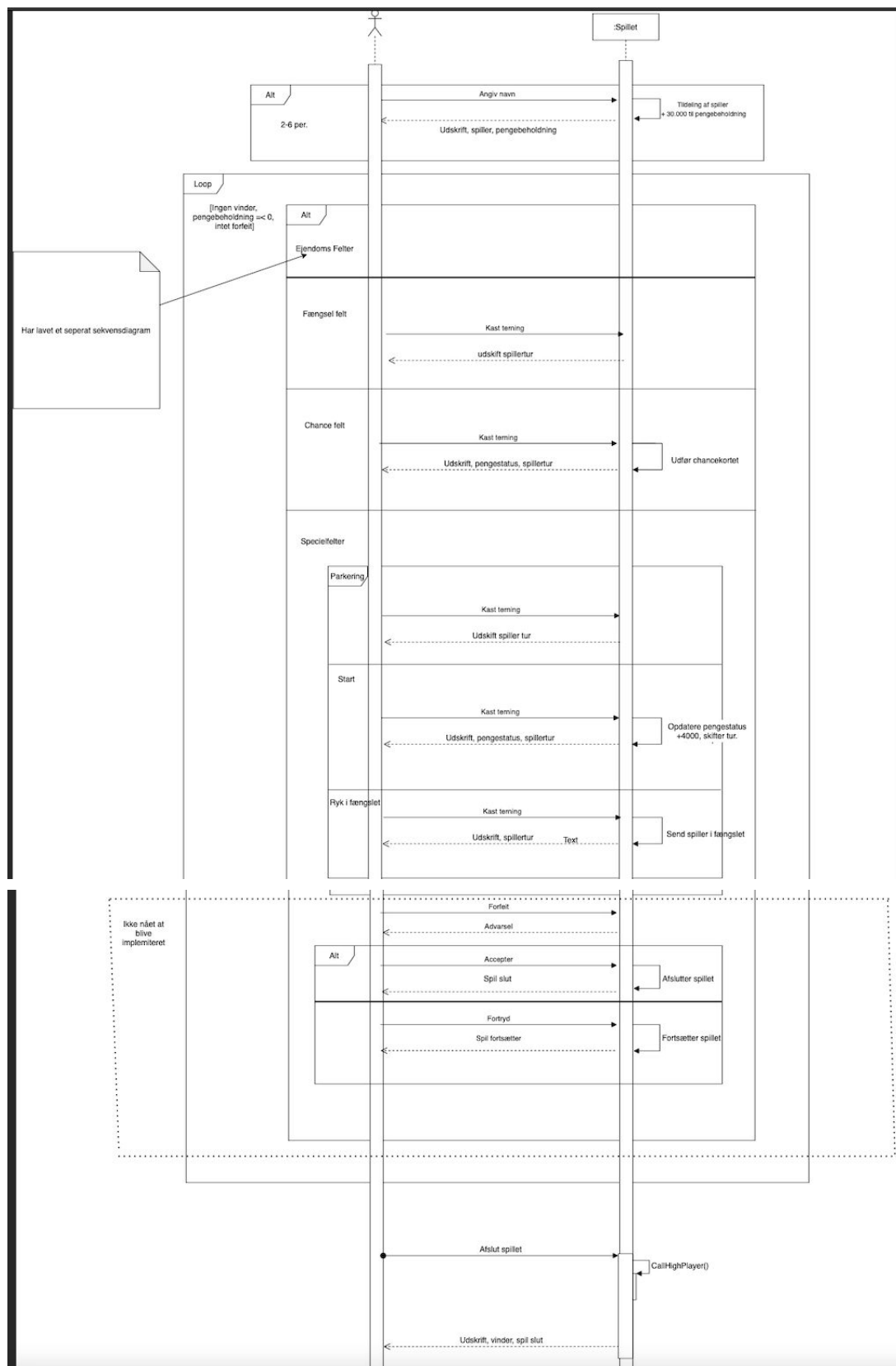
Use Case: Play turn
ID: 2
Brief description: Spilleren SKAL kaste terningen, men kan inden det vælge at sælge og købe ejendomme og grunde
Aktører: <ul style="list-style-type: none">• Spiller• System
Pre conditions: <ul style="list-style-type: none">• Spillet er startet• Det er spillerens tur• At spilleren kan komme ud hvis i fængsel {enten ved "get out of jail card " eller 1000 kr}
Main flow: <ol style="list-style-type: none">1. Use casen starter når det er spillerens tur2. Spilleren kaster terningen3. Der startes et loop til EN af de extended use cases{Sell property, Buy property} og ALLE de inkluderede use cases{Roll dice, Play turn}, som udføres efter terningekast, og derefter bliver bragt tilbage til play turn, evt ved en ny spiller.
Post conditions: <ul style="list-style-type: none">• Spilleren ejer nu købte grunde eller felter• Spilleren ejer ikke længere solgte grunde eller felter• Use casen starter forfra ved ny spiller
Alternative flow: <ol style="list-style-type: none">2a. Spilleren vælger at sælge eller købe en ejendom eller grund FØR spilleren kaster terningen.3a. Tiden for spillet udløber, loopet stopper og en spiller vinder.3b. Alle spillere undtagen 1 er gået fallit, loopet stopper og en spiller vinder.

Domænemodel:

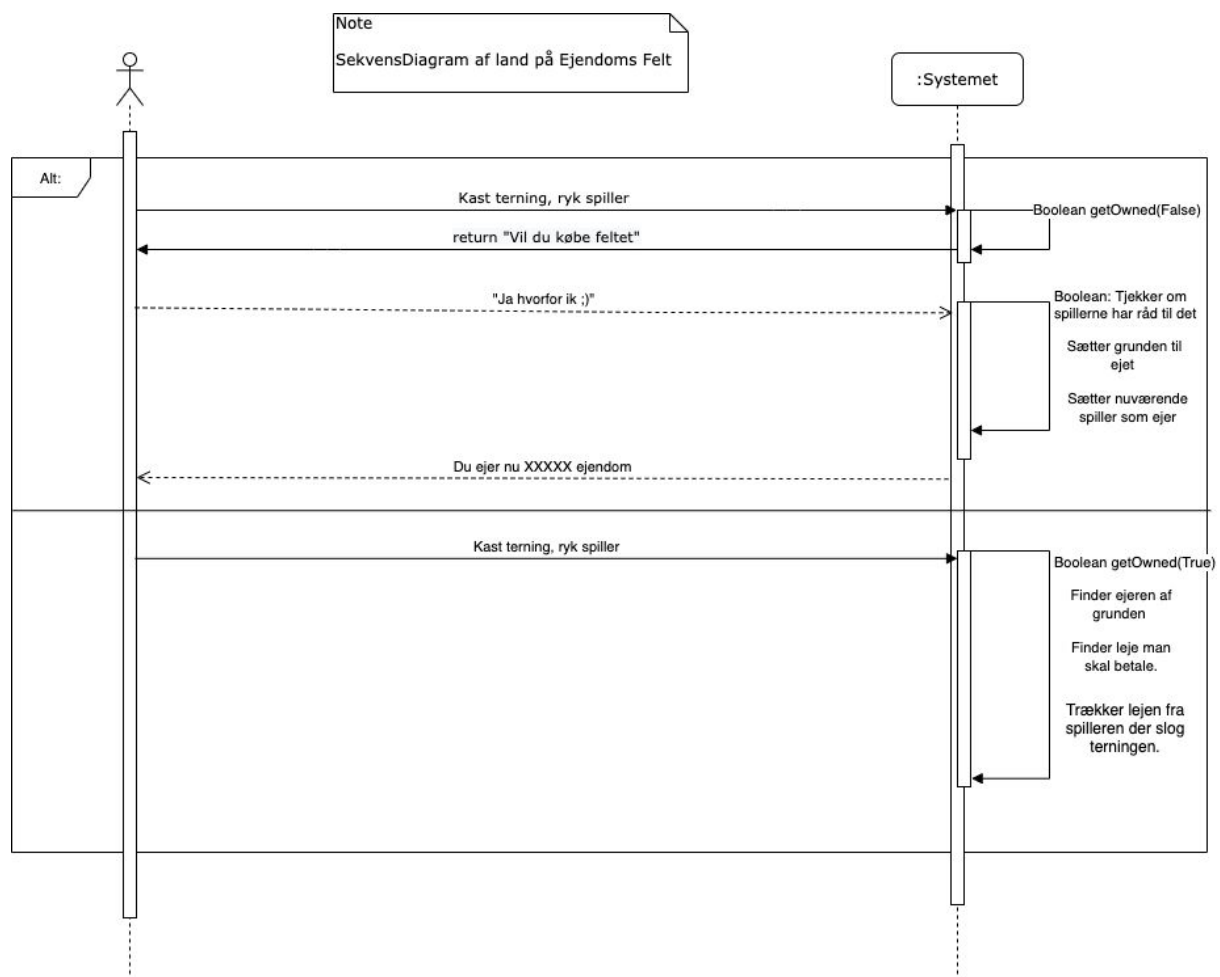


- Vi er obs på at vores terningers path ikke er korrekte, men grundet den måde hvorpå vi har implementeret dem, har vi valgt at skrive det op på følgende måde.

System Sekvensdiagram:

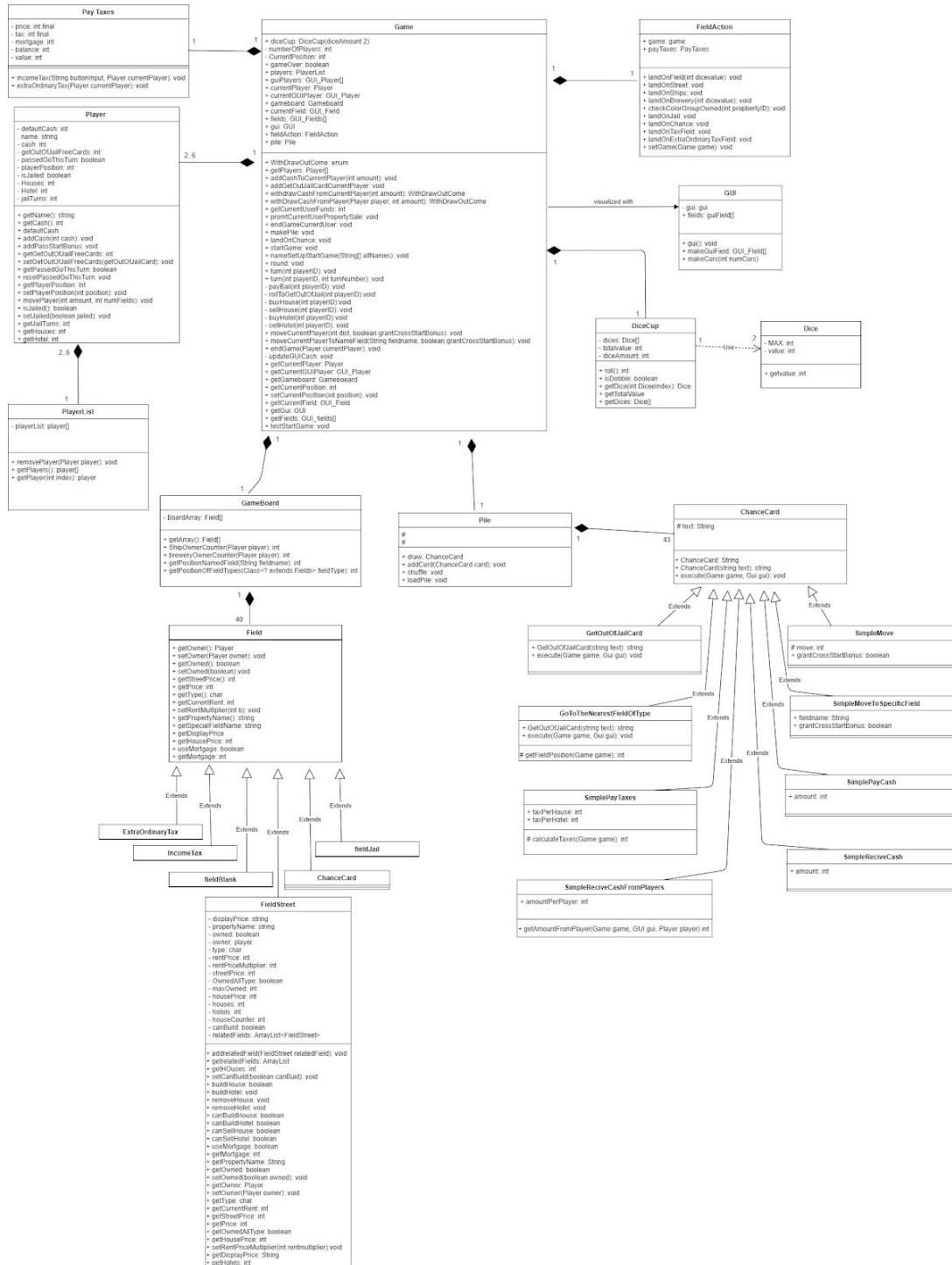


Følgende sekvensdiagram er lavet på det manglende scenarie for ejendoms felter.



Kapitel 4 - Design

Klassediagram:

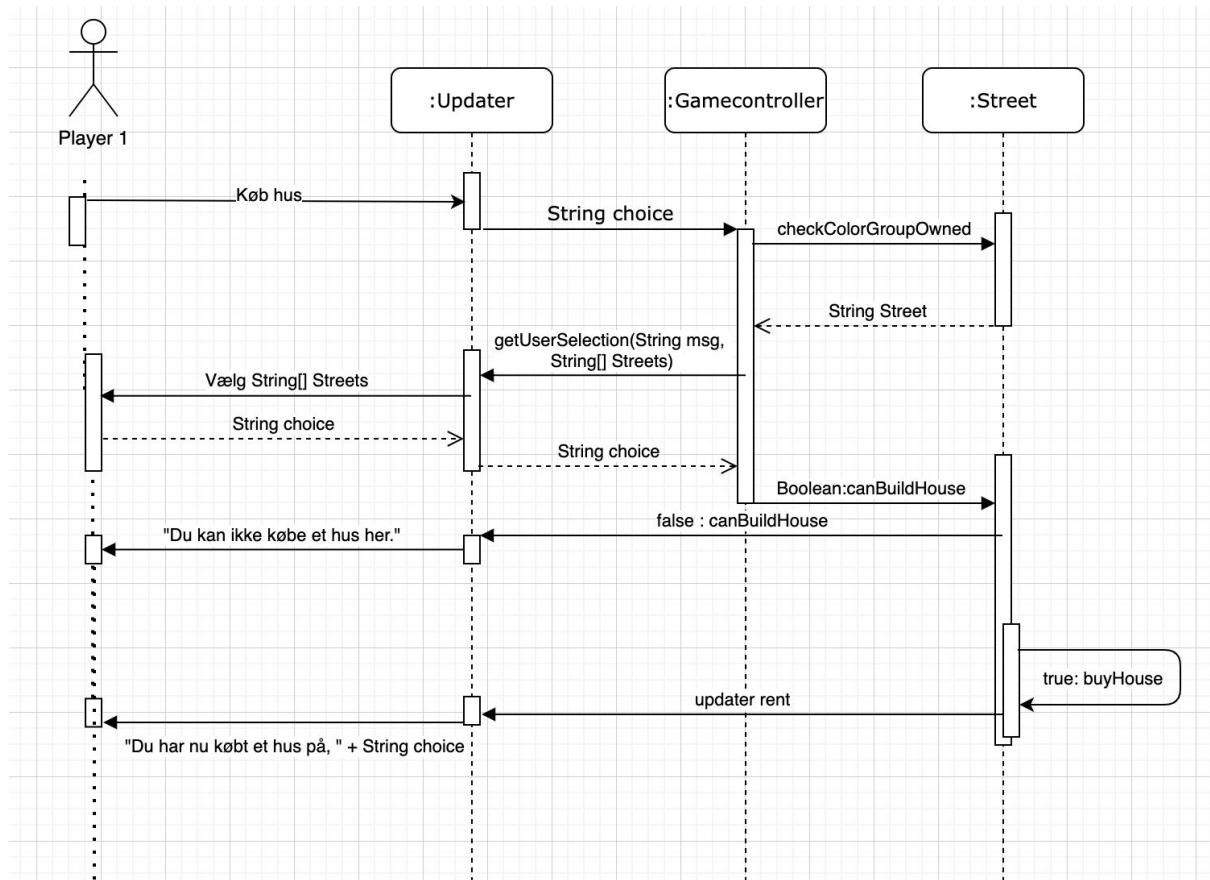


3

³ Link til at hente diagrammet ned fra extern drive mappe med både png og diagram-fil:
<https://drive.google.com/drive/folders/1koF635fDU8rx8WKkkMkME91HJRLYTJIn?usp=sharing>

Sekvensdiagram:

Køb Hus



Kapitel 5 - Implementering

Kodedel fra Gameboard klassen:

```
public int[] getPositionsOfFieldTypes(Class<? extends Fields> fieldType){
    List<Integer> positions = new ArrayList<>();
    for (int i = 0; i < boardArray.length; i++) {
        if (boardArray[i].getClass() == fieldType) {
            positions.add(Integer.valueOf(i));
        }
    }
    int[] retVal = new int[positions.size()];
    for (int i=0;i<positions.size();i++) {
        retVal[i] = positions.get(i).intValue();
    }
    return retVal;
}
```

getPositionOfFieldTypes tager en generisk klasse som parameter ***Class<? extends Fields>***. Vi definerer med denne syntaks at klassen kan være hvilken som helst klasse, dog SKAL den være nedarvet fra *Fields* klassen. Vi laver nu en *ArrayList* med en for loop metode der søger igennem board arrayet og finder positionen i arrayet af alle *Fields* som er af den specificerede *Fields* subklasse. Den specificerede *Fields* subklasse, er specificeret i vores pile i *Pile* klassen

```
addCard( new GotoNearestFieldOfType( text: "Tag med den nærmeste færg", FieldShips.class ));
```

Metoden benytter sig af at alle Java object har en generisk *getClass* metode, som returnerer den klasse som objektet er en instans af. Så i gennemløbet af board array tilføjer den alle de positioner hvor objektet er en instans af *FieldShips* til *ArrayList*en.

I metoden benyttes en mutable liste *List<Integer>* som en mellem variable, da vi leder efter alle felterne af denne type og ikke behøver at specificere hvor mange der er, herefter konverteres det til et almindeligt immutable integer array, før der returneres, da vi nu præcis ved hvor mange elementer der er i arrayet.

Kodedel fra Pile klassen:


```
public class Pile {
    protected Queue<ChanceCard> cards = new LinkedList<>();
    protected Queue<ChanceCard> currentPile = new LinkedList<>();

    public ChanceCard draw(){
        ChanceCard card = currentPile.poll();
        if(card != null) currentPile.add(card);
        return card;
    }

    public void addCard(ChanceCard card) { cards.add(card); }

    public void shuffle(){
        ArrayList<ChanceCard> tmp = new ArrayList<>(cards);
        Collections.shuffle(tmp);
        currentPile = new LinkedList<ChanceCard>(tmp);
    }
}
```

Pile klassen implementerer en stak af chancekort. Metoden benytter sig af Java generics til at implementere en dynamisk stak. Vi bruger en LinkedList til at implementere en Queue i stedet for et ArrayList da LinkedList benytter sig af dobbelt linked list implementation, som er mere effektiv mht. fjernelse af elementer.

Draw metoden tager brug af poll() metoden som returnerer det første element i listen(currentPile) som er toppen af stakken og bruger derefter add() metoden som bliver brugt til at lægge elementer bagerst i køen. I vores tilfælde laver vi en tilbagelægning af kortet som lige er trukket til bunden af stakken. Stakken bliver dermed aldrig tom.

Pile implementerer en shuffle metode til at blande stakken. I shuffle anvendes en midlertidig variabel, da LinkedList ikke implementerer Random Access interface. Vi kopierer derfor listen til en ArrayList for at benytte metoden Java Collections shuffle, og derefter ændre den tilbage til en LinkedList.

Kode brugt i forskellige klasser:

```
public enum WithdrawOutcome {
    OK,
    INSUFFICIENT_CASH
}
```

WithdrawOutcome er en Java enumeration, som definerer mulige udfald af at trække penge fra en spillers konto. Enum værdierne defineres som:

OK: Spilleren havde penge nok og beløbet er trukket

INSUFFICIENT_CASH: Spilleren havde ikke penge nok på kontoen og beløbet blev ikke trukket.

```
public WithdrawOutcome withdrawCashFromPlayer(Player player, int amount) {
    if (player.getCash() >= amount) {
        player.addCash(-amount);
        return WithdrawOutcome.OK;
    }
    else return WithdrawOutcome.INSUFFICIENT_CASH;
}
```

I dette stykke kode er det specificeret hvad der skal være muligt at ske for at case værdien er OK.

Metoden validerer at spilleren har tilstrækkeligt med penge på sin konto. Hvis spilleren har penge nok, trækkes det specificeret beløb fra kontoen og et OK svar returneres. Hvis spilleren ikke har penge nok returneres fejlsvaret INSUFFICIENT_CASH.

```
public void execute(Game game, GUI gui){
    gui.displayChanceCard(this.text);
    switch (game.withdrawCashFromCurrentPlayer(this.amount)) {
        case OK:
            break;
        case INSUFFICIENT_CASH: {
            int missing = this.amount - game.getCurrentUserFunds();
            gui.showMessage(String.format("Du har ikke penge nok. Der mangler %d \n Sælg ejendomme for at få penge nok", missing));
            // allow the user to sell properties (not implemented - could be for further work)
            game.promptCurrentUserPropertySale();
            // Then check whether the user now has enough cash otherwise end game for user
            switch (game.withdrawCashFromCurrentPlayer(this.amount)) {
                case OK:
                    break;
                case INSUFFICIENT_CASH:
                    game.endGameCurrentUser();
                    break;
            }
        }
        break;
    }
}
```

Execute metoden i samtlige klasser implementerer et flow, hvor der forsøges at trække det specificerede beløb fra spillerens konto. Metoden kalder ud til Game, som enten vil returnere OK hvis pengene blev trukket fra spillerens konto eller fejlsvar INSUFFICIENT_CASH, hvis spilleren ikke har penge nok. I dette stykke kode kan det ses at hvis casen har værdien INSUFFICIENT_CASH kalder metoden endGameCurrentUser, som slutter spillet for spilleren. Det kan samtidigt ses her at der er lagt op til at spilleren bliver promptet til at sælge ejendomme for at få penge nok, hvorefter der igen forsøges at trække penge fra spillerens konto. Metoden for salg gør ikke noget da den ikke er implementeret, men er her oprettet, hvis det i senere udvikling bliver muligt for en spiller at sælge for at få flere penge.

Kode fra Game klassen:

```
private void buyHouse(int playerID) {
    int totalOwnedFields = 0;
    FieldStreet[] ownedFields = new FieldStreet[40]; // Der er ikke 40 fieldStreets på brættet, men det går nok
    for (int i = 0; i < this.gameboard.getArray().length; i++) {
        if (!(this.gameboard.getArray()[i] instanceof FieldStreet)) {
            continue;
        }
        // Pt tjekker vi alle felter i en gruppe alle de gange hvor feltet er der (men it works!)
        // (Så hvis en gruppe felter har 5 felter så tjekkes det hele 5 gange i stedet for kun 1 gang)
        fieldAction.checkColorGroupOwned(i);
        FieldStreet field_i = (FieldStreet) this.gameboard.getArray()[i];
        if (field_i.getOwner() == this.currentPlayer && field_i.canBuildHouse()) {
            ownedFields[totalOwnedFields] = field_i;
            totalOwnedFields++;
        }
    }
    if (totalOwnedFields == 0) {
        gui.showMessage( msg: "Det ser ikke ud til at du kan købe nogen huse");
        turn(playerID);
        return;
    }
    String[] streetSelection = new String[totalOwnedFields];
    for (int i = 0; i < totalOwnedFields; i++) {
        streetSelection[i] = ownedFields[i].getPropertyName();
    }
    String streetChoice = gui.getUserSelection( msg: "Hvor vil du købe et hus?", streetSelection);
    int streetChoiceInteger = 0;
    for (int i = 0; i < this.gameboard.getArray().length; i++) {
        if (this.gameboard.getArray()[i].getPropertyName() == streetChoice) {
            streetChoiceInteger = i;
            break;
        }
    }

    ((FieldStreet) this.gameboard.getArray()[streetChoiceInteger]).buildHouse();
    ((GUI_Street) this.fields[streetChoiceInteger]).setHouses(((FieldStreet) this.gameboard.getArray()[streetChoiceInteger]).getHouses());
    updateGUICash();
    gui.showMessage( msg: players.getPlayer(playerID).getName() + " har købt et hus på " + streetChoice + ".");
    turn(playerID);
}
```

buyHouse metoden går ud på at bruge nogle if statements og for loops for at først sikre sig at en spiller er i stand til at købe et hus, og derefter bygge det der hvor spilleren gerne vil have den. Først bliver der lavet en liste (ownedFields) som indeholder alle de felter som spilleren ejer. Derefter vil den fortælle spilleren at de ikke kan købe et hus hvis de ejer 0 felter, og ellers vil den give dem en liste over felterne de kan købe huse på. Spilleren kan så vælge et af dem hvor der derefter bygges et hus.

Kapitel 6 - Test

Dokumentation for gennemført test

DiceTest:

```
Number of eyes from 2 to 12 starting lowest to highest: [25, 48, 76, 128, 142, 180, 125, 110, 83, 59, 24]
Number of dice rolls with the same number of eyes on each die: 155

Expected probabilities for number of eyes: [2.78, 5.56, 8.33, 11.11, 13.89, 16.67, 13.89, 11.11, 8.33, 5.56, 2.78]
Actual probabilities for number of eyes: [2.5, 4.8, 7.6, 12.8, 14.2, 18.0, 12.5, 11.0, 8.3, 5.8999999999999995, 2.4]

Expected probability for same number of eyes on each die: 16.667
Actual probability for same number of eyes on each die: 15.5
roll value is the same as dice.getEyeValue
Checking if roll value is the same as a second call to dice.getEyeValue
Checking that dice.getEyeValue is the same as dice.getEyeValue. It is.
Rolling new die
Old roll value is 3. New roll value is 2.
Old roll value is not the same as dice.getEyeValue
roll value is the same as dice.getEyeValue
Checking if roll value is the same as a second call to dice.getEyeValue
Checking that dice.getEyeValue is the same as dice.getEyeValue. It is.
```

I vores DiceTest har vi brugt de samme tests som i CDIO_del3, men med få ændringer. Vi tester sandsynligheden for diverse slag, samt tjekker op på at de reelt opfører sig tilfældigt. På billedet ovenover er et test-run, som viser alle de ting, med tekst, som vi har testet, samt resultatet på dette.

StreetFieldTest

✓	Test Results	35 ms
✓	StreetFieldTest	35 ms
✓	testCannotBuildWithoutOwner()	23 ms
✓	testBuyShipAndCola()	5 ms
✓	testCanLinkFields()	2 ms
✓	testCanPlayerBuyHouseWithoutOwn	3 ms
✓	testStreetCreation()	1 ms
✓	testBuildMaxFourHouses()	1 ms

I vores StreetFieldTest tester vi for flere ting:

- At man ikke kan bygge huse uden en ejer
- At man kan købe skibe og cola felter
- At man kan linke (referere) til felterne andre steder fra.
- At man ikke kan købe et hus uden at eje alle felter.
- At man kan lave nye Streets
- At en spiller max kan bygge 4 huse på en grund

Som man kan se på billedet, løber alle testene korrekt igennem, uden problemer. (Det lille grønne check mark foran metode navnene.)

SimpleReciveCashFromPlayersTest:

✓ Test Results	33 ms
✓ StreetFieldTest	33 ms
✓ testCannotBuildWithoutOwner()	23 ms
✓ testBuyShipAndCola()	3 ms
✓ testCanLinkFields()	3 ms
✓ testCanPlayerBuyHouseWithoutOwi	2 ms
✓ testStreetCreation()	1 ms
✓ testBuildMaxFourHouses()	1 ms

```
void receiveCashFromPlayersChanceCard() {
    Game game = new Game();
    game.testStartGame();

    for (Player p : game.getPlayers()) {
        assertEquals( expected: 30000, p.getCash());
    }

    SimpleReceiveCashFromPlayers cashFromPlayer = new SimpleReceiveCashFromPlayers( text: "du modtager 200 fra de andre bruh", amountPerPlayer: 200);
    cashFromPlayer.execute(game, game.getGui());

    assertEquals( expected: 30400, game.getPlayers()[0].getCash());
    assertEquals( expected: 29800, game.getPlayers()[1].getCash());
    assertEquals( expected: 29800, game.getPlayers()[2].getCash());
}
```

I denne SimpleReciveCashFromPlayersTest tester vi at det fungerer rigtigt når man lander på et chancekort, og en spiller skal modtage penge fra de andre spillere. Det første billede viser at alle test kører korrekt.

Det andet billede viser hvordan vi har opsat det enkelt, så vi hurtigt kan teste hvordan dette fungerer. Vi starter med lige at få fat i et game, og bruge en metode der hedder testStartGame, som er en simplen verison af startGame, men uden GUI-delen. Den kommer med 3 default spillere. Derefter benytter vi metoden SimpleReciveCashFromPlayers til at modtage 200 fra de 2 andre spillere.

Så kører vi metoden execute som der udfører udvekslingen.

Derefter bruger vi assertEquals til at teste om det fungerer som vi forventer. (At en spiller modtager 200 fra hver spiller, 400 i alt, og at de to spillere reelt mister de 200 som de giver.)

Vurdering af systemets status/kvalitetsmæssigt

Vores vurdering af vores projekt er, at vi har fået et spil der har alle de features vi havde håbet på, som vi vurderede som de vigtigste, samt at de fleste af de features som vi gerne ville have udviklet på, har vi fået udviklet, og/eller påbegyndt, så vi kan videreudvikle i fremtiden. Dog er vores vurdering også, at der stadig er stor plads til forbedringer, både ift selve den kode som vi har skrevet, men også mht de implementeringer og de bugs som vi gerne ville have kunne arbejde videre på, givet de rette betingelser og tid. Vi synes vores kvalitet på det produkt som vi har afleveret har været svingende, men fungerende. Det vi har afleveret har en okay kvalitet, men med massere af plads til at udvikle videre, samt at implementere nogle ting, som vi har lagt op til.

Kapitel 7 - Delkonklusion

Vi har et Matador-spil, hvor de mest basale funktioner og features virker. Der er nogle features som vi havde håbet at have færdiggjort, som mangler. Nogle er under udvikling, men da man skal overholde sin tidsgrænse for aflevering, så må de desværre falde under kategorien, "Klar til videreudvikling".

Features:

Skulle være færdigt

- To terninger ✓
- Et brætspil ✓
- 40 felter ✓
- 6 biler ✓
- Alle spillere starter på start ✓
- Spilleren flytter så mange felter øjnene på terningen viser ✓
- Starter ud med at have 30000kr når spillet start ✓
- Modtager 4000kr når man passere start, med undtagelsen når man ryger i fængsel ✓
- Prøv lykken kort (✓) 2 kort ikke lavet
- Ekstra kast ✓
- Køb og leje ✓
- Fængsel ✓
- Skøder ✓
- Indtast spillernavne ✓
- den første der indtaster navn starter ✓
- synlig konto ✓
- vælge antal spillere ✓
- Huse og hoteller ✓
- Indkomstskat ✓

Burde være halvt færdigt

- Pantsætning
- Aktion
- "Fallit" reglen (Vi har ingen auktion så dette er ikke en mulighed)

Ikke færdigt

- Random spiller starter
- Hurtigt spil

Kapitel 8 - Konklussion - Forbedringere til fremtiden

Til videreudvikling er vi klar til:

Skulle have være færdigt:

- **Prøv lykken kort (✓) mangler 2 kort**
 -
- **“Fallit” reglen (Vi har ingen auktion så dette er ikke en mulighed)**
 - En spiller kan godt gå fallit, men der bliver ikke lavet en auktion bagefter. Har vi desværre ikke kunne gå i gang med

Burde være halvt færdigt:

- **Pantsætning**
 - Vi har basiske metoderne klar, men de er aldrig implementeret i selve spillet, da vi løb tør for tid. Men det er klar til at blive lavet færdigt, hvis mere tid blev tilrettelagt til os.
- **Aktion**
 - Vi har forslag til metoderne klar, men de har ikke virket i test, så om de er korrekte er op til debat. Men det er klar til at blive lavet færdigt, hvis mere tid blev tilrettelagt os.

Ikke færdigt

- **Tilfældig spiller starter**
 - Ikke påbegyndt.

Tilfældig spiller starter, er noget der nemmere kan implementeres, da metode bare skal køre når spillet starter.

- **Hurtigt spil**
 - Ikke færdigt

Hurtigt spil, skal man give to tilfældige skøder til hvert så det kunne man implementere at den vælger to tilfældige skøder fra streets og giver til hver spiller og at man sætter antal tid på selve spillet før det slutter og til sidst bruger en metode for at tælle spillerens værdier op.

Andre notater:

- Til næste projekt vil vi være mere fokuserede på at få en korrekt .gitignore, da dette gav store problemer, efter vi fik forkerte anvisninger, og dermed forsinkede hele projektet en dag mere end forventet, som derfor betød at vi ikke kunne nå lige så mange features som vi havde planlagt.
- Vi er opmærksomme på en bug, som gør at spillere kan gå i minus når de forsøger at købe huse og hoteller, eller hvis de ikke kan betale leje. Så vi mangler et område hvorpå at vi kalder game over for en spiller, hvis de går i minus. Dette kunne desværre ikke blive nået at blive fikset.

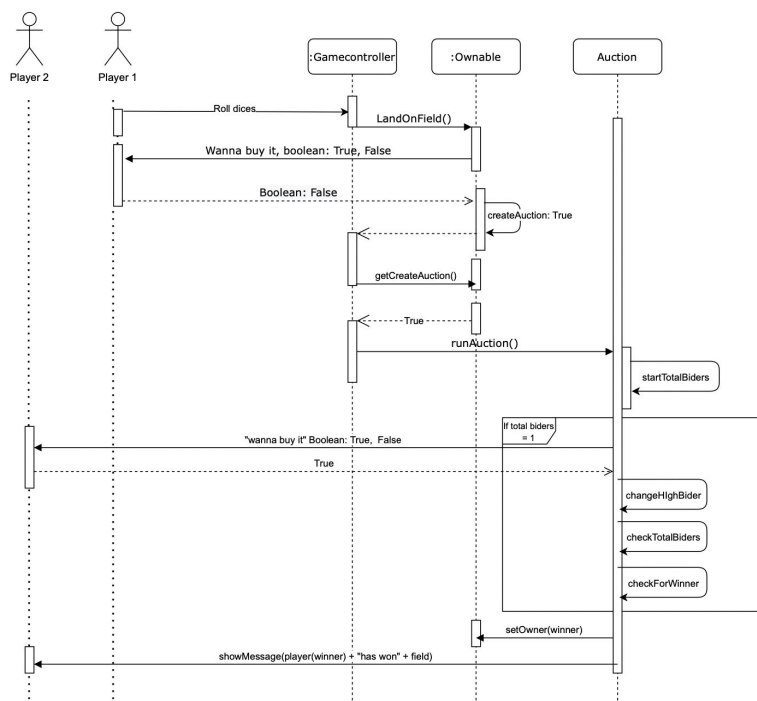
Mortgage property er en use case, som vi havde håbet at få gennemført, men som vi erkender at vi ikke havde nok tid til at få implementeret.

Mortgage property

Spilleren pantsætter sine grunde, i bytte for penge

Auktion

Dette er var vores tankegang for en udvikling af scenariet vedrørende en auktion



9. Litteraturliste / Bilag:

bilag 1: Spillepladen



Katrine Amalie Harris Holst s196509, Simon Schnack Jørgensen s195822, Steven Christian Proudfoot s205465, Steven Gordon Pedersen s205456, Oline Melinda Andersen s201010, Mike Helt Jensen s205865

Bilag 2.1 : Ejendomskort

Scandlines

GEDSER - ROSTOCK

Leje	kr.	500
Hvis 2 rederier ejes	»	1.000
Hvis 3 rederier ejes	»	2.000
Hvis 4 rederier ejes	»	4.000
Pantsætningsværdi	kr.	2.000

Scandlines

HELSINGØR-HELSINGBORG

Leje	kr.	500
Hvis 2 rederier ejes	»	1.000
Hvis 3 rederier ejes	»	2.000
Hvis 4 rederier ejes	»	4.000
Pantsætningsværdi	kr.	2.000

Hvis 1 virksomhed ejes, betales 100 gange så meget, som øjnene viser.

Hvis både Coca-Cola og TUBORG SQUASH ejes, betales 200 gange så meget, som øjnene viser.

Pantsætningsværdi kr. 1.500

Allégade

Leje af grund	kr.	150
m/ 1 hus	»	800
» 2 huse	»	2.000
» 3 huse	»	6.000
» 4 huse	»	9.000
» hotel	»	12.000
Hvis en spiller ejer alle grundene i samme farvegruppe, fordobles lejen af de ubebyggede grunde i den pågældende gruppe.		
Hvert hus koster	kr.	1.000
Et hotel koster	»	1.000
+ 4 huse		
Pantsætningsværdi	kr.	1.200

Rødovrevej

Leje af grund	kr.	50
m/ 1 hus	»	250
» 2 huse	»	750
» 3 huse	»	2.250
» 4 huse	»	4.000
» hotel	»	6.000
Hvis en spiller ejer alle grundene i samme farvegruppe, fordobles lejen af de ubebyggede grunde i den pågældende gruppe.		
Hvert hus koster	kr.	1.000
Et hotel koster	»	1.000
+ 4 huse		
Pantsætningsværdi	kr.	600

Mols-Linien

En genvej i Danmark

Leje	kr.	500
Hvis 2 rederier ejes	»	1.000
Hvis 3 rederier ejes	»	2.000
Hvis 4 rederier ejes	»	4.000
Pantsætningsværdi	kr.	2.000

Scandlines

KØBBØ - PUTTGÅRDEN

Leje	kr.	500
Hvis 2 rederier ejes	»	1.000
Hvis 3 rederier ejes	»	2.000
Hvis 4 rederier ejes	»	4.000
Pantsætningsværdi	kr.	2.000

Hvis 1 virksomhed ejes, betales 100 gange så meget, som øjnene viser.

Hvis både TUBORG SQUASH og Coca-Cola ejes, betales 200 gange så meget, som øjnene viser.

Pantsætningsværdi kr. 1.500

Roskildevej

Leje af grund	kr.	100
m/ 1 hus	»	600
» 2 huse	»	1.800
» 3 huse	»	5.400
» 4 huse	»	8.000
» hotel	»	11.000
Hvis en spiller ejer alle grundene i samme farvegruppe, fordobles lejen af de ubebyggede grunde i den pågældende gruppe.		
Hvert hus koster	kr.	1.000
Et hotel koster	»	1.000
+ 4 huse		
Pantsætningsværdi	kr.	1.000

Hvidovrevej

Leje af grund	kr.	50
m/ 1 hus	»	250
» 2 huse	»	750
» 3 huse	»	2.250
» 4 huse	»	4.000
» hotel	»	6.000
Hvis en spiller ejer alle grundene i samme farvegruppe, fordobles lejen af de ubebyggede grunde i den pågældende gruppe.		
Hvert hus koster	kr.	1.000
Et hotel koster	»	1.000
+ 4 huse		
Pantsætningsværdi	kr.	600

Bilag 2.2 : Ejendomskort

Østergade

Leje af grund	kr.	500
m/ 1 hus	»	2.400
» 2 huse	»	7.200
» 3 huse	»	17.000
» 4 huse	»	20.500
» hotel	»	24.000

Hvis en spiller ejer alle grundene i samme farvegruppe, fordobles lejen af de ubebyggede grunde i den pågældende gruppe.

Hvert hus koster	kr.	3.000
Et hotel koster	»	3.000
+ 4 huse		
Pantsætningsværdi	kr.	2.800

Hellerupvej

Leje af grund	kr.	300
m/ 1 hus	»	1.400
» 2 huse	»	4.000
» 3 huse	»	11.000
» 4 huse	»	15.000
» hotel	»	19.000

Hvis en spiller ejer alle grundene i samme farvegruppe, fordobles lejen af de ubebyggede grunde i den pågældende gruppe.

Hvert hus koster	kr.	2.000
Et hotel koster	»	2.000
+ 4 huse		
Pantsætningsværdi	kr.	1.800

Østerbrogade

Leje af grund	kr.	350
m/ 1 hus	»	1.800
» 2 huse	»	5.000
» 3 huse	»	14.000
» 4 huse	»	17.500
» hotel	»	21.000

Hvis en spiller ejer alle grundene i samme farvegruppe, fordobles lejen af de ubebyggede grunde i den pågældende gruppe.

Hvert hus koster	kr.	3.000
Et hotel koster	»	3.000
+ 4 huse		
Pantsætningsværdi	kr.	2.200

Amagertorv

Leje af grund	kr.	550
m/ 1 hus	»	2.600
» 2 huse	»	7.800
» 3 huse	»	18.000
» 4 huse	»	22.000
» hotel	»	25.000

Hvis en spiller ejer alle grundene i samme farvegruppe, fordobles lejen af de ubebyggede grunde i den pågældende gruppe.

Hvert hus koster	kr.	4.000
Et hotel koster	»	4.000
+ 4 huse		
Pantsætningsværdi	kr.	3.000

Gl. Kongevej

Leje af grund	kr.	250
m/ 1 hus	»	1.250
» 2 huse	»	3.750
» 3 huse	»	10.000
» 4 huse	»	14.000
» hotel	»	18.000

Hvis en spiller ejer alle grundene i samme farvegruppe, fordobles lejen af de ubebyggede grunde i den pågældende gruppe.

Hvert hus koster	kr.	2.000
Et hotel koster	»	2.000
+ 4 huse		
Pantsætningsværdi	kr.	1.600

Kgs. Nytorv

Leje af grund	kr.	450
m/ 1 hus	»	2.200
» 2 huse	»	6.600
» 3 huse	»	16.000
» 4 huse	»	19.500
» hotel	»	23.000

Hvis en spiller ejer alle grundene i samme farvegruppe, fordobles lejen af de ubebyggede grunde i den pågældende gruppe.

Hvert hus koster	kr.	3.000
Et hotel koster	»	3.000
+ 4 huse		
Pantsætningsværdi	kr.	2.600

Strandvejen

Leje af grund	kr.	350
m/ 1 hus	»	1.600
» 2 huse	»	4.400
» 3 huse	»	12.000
» 4 huse	»	16.000
» hotel	»	20.000

Hvis en spiller ejer alle grundene i samme farvegruppe, fordobles lejen af de ubebyggede grunde i den pågældende gruppe.

Hvert hus koster	kr.	2.000
Et hotel koster	»	2.000
+ 4 huse		
Pantsætningsværdi	kr.	2.000

Triangeln

Leje af grund	kr.	350
m/ 1 hus	»	1.800
» 2 huse	»	5.000
» 3 huse	»	14.000
» 4 huse	»	17.500
» hotel	»	21.000

Hvis en spiller ejer alle grundene i samme farvegruppe, fordobles lejen af de ubebyggede grunde i den pågældende gruppe.

Hvert hus koster	kr.	3.000
Et hotel koster	»	3.000
+ 4 huse		
Pantsætningsværdi	kr.	2.200

Nygade

Leje af grund	kr.	600
m/ 1 hus	»	3.000
» 2 huse	»	9.000
» 3 huse	»	20.000
» 4 huse	»	24.000
» hotel	»	28.000

Hvis en spiller ejer alle grundene i samme farvegruppe, fordobles lejen af de ubebyggede grunde i den pågældende gruppe.

Hvert hus koster	kr.	4.000
Et hotel koster	»	4.000
+ 4 huse		
Pantsætningsværdi	kr.	3.200

Bülowsvej

Leje af grund	kr.	200
m/ 1 hus	»	1.000
» 2 huse	»	3.000
» 3 huse	»	9.000
» 4 huse	»	12.500
» hotel	»	15.000

Hvis en spiller ejer alle grundene i samme farvegruppe, fordobles lejen af de ubebyggede grunde i den pågældende gruppe.

Hvert hus koster	kr.	2.000
Et hotel koster	»	2.000
+ 4 huse		
Pantsætningsværdi	kr.	1.400

Bredgade

Leje af grund	kr.	450
m/ 1 hus	»	2.200
» 2 huse	»	6.600
» 3 huse	»	16.000
» 4 huse	»	19.500
» hotel	»	23.000

Hvis en spiller ejer alle grundene i samme farvegruppe, fordobles lejen af de ubebyggede grunde i den pågældende gruppe.

Hvert hus koster	kr.	3.000
Et hotel koster	»	3.000
+ 4 huse		
Pantsætningsværdi	kr.	2.600

Bernstorffsvej

Leje af grund	kr.	300
m/ 1 hus	»	1.400
» 2 huse	»	4.000
» 3 huse	»	11.000
» 4 huse	»	15.000
» hotel	»	19.000

Hvis en spiller ejer alle grundene i samme farvegruppe, fordobles lejen af de ubebyggede grunde i den pågældende gruppe.

Hvert hus koster	kr.	2.000
Et hotel koster	»	2.000
+ 4 huse		
Pantsætningsværdi	kr.	1.800

Grønningen

Leje af grund	kr.	400
m/ 1 hus	»	2.000
» 2 huse	»	6.000
» 3 huse	»	15.000
» 4 huse	»	18.500
» hotel	»	22.000

Hvis en spiller ejer alle grundene i samme farvegruppe, fordobles lejen af de ubebyggede grunde i den pågældende gruppe.

Hvert hus koster	kr.	3.000
Et hotel koster	»	3.000
+ 4 huse		
Pantsætningsværdi	kr.	2.400

Vimmelskæft

Leje af grund	kr.	550
m/ 1 hus	»	2.600
» 2 huse	»	7.800
» 3 huse	»	18.000
» 4 huse	»	22.000
» hotel	»	25.000

Hvis en spiller ejer alle grundene i samme farvegruppe, fordobles lejen af de ubebyggede grunde i den pågældende gruppe.

Hvert hus koster	kr.	4.000
Et hotel koster	»	4.000
+ 4 huse		
Pantsætningsværdi	kr.	3.000

Frederiksberg Allé

Leje af grund	kr.	200
m/ 1 hus	»	1.000
» 2 huse	»	3.000
» 3 huse	»	9.000
» 4 huse	»	12.500
» hotel	»	15.000

Hvis en spiller ejer alle grundene i samme farvegruppe, fordobles lejen af de ubebyggede grunde i den pågældende gruppe.

Hvert hus koster	kr.	2.000
Et hotel koster	»	2.000
+ 4 huse		
Pantsætningsværdi	kr.	1.400

Bilag 2.3 : Ejendomskort

Valby Langgade		Rådhuspladsen		Frederiksberggade	
Leje af grund	kr. 100	Leje af grund	kr. 1.000	Leje af grund	kr. 700
m/ 1 hus	» 600	m/ 1 hus	» 4.000	m/ 1 hus	» 3.500
» 2 huse	» 1.800	» 2 huse	» 12.000	» 2 huse	» 10.000
» 3 huse	» 5.400	» 3 huse	» 28.000	» 3 huse	» 22.000
» 4 huse	» 8.000	» 4 huse	» 34.000	» 4 huse	» 26.000
» hotel	» 11.000	» hotel	» 40.000	» hotel	» 30.000
Hvis en spiller ejer alle grundene i samme farvegruppe, fordobles lejen af de ubebyggede grunde i den pågældende gruppe.		Hvis en spiller ejer alle grundene i samme farvegruppe, fordobles lejen af de ubebyggede grunde i den pågældende gruppe.		Hvis en spiller ejer alle grundene i samme farvegruppe, fordobles lejen af de ubebyggede grunde i den pågældende gruppe.	
Hvert hus koster	kr. 1.000	Hvert hus koster	kr. 4.000	Hvert hus koster	kr. 4.000
Et hotel koster	» 1.000	Et hotel koster	» 4.000	Et hotel koster	» 4.000
	+ 4 huse		+ 4 huse		+ 4 huse
Pantsætningsværdi	kr. 1.000	Pantsætningsværdi	kr. 4.000	Pantsætningsværdi	kr. 3.500

Bilag 3.1 : Regelsæt

Spillets formål

- Formålet med spillet er at købe, udleje eller sælge ejendomme så fordelagtigt, at man bliver den rigeste spiller og dermed spillets eneste matador.
- Man begynder ved "START" og flytter brikkerne venstre om ifølge terningkast. Når en spillers brik lander på et felt, der ikke allerede ejes af nogen anden deltager, kan spilleren købe det af banken og indkassere leje af modspillerne, når de lander på det pågældende felt. Ønsker spilleren ikke at købe grunden, sætter banken det straks på auktion.
- Lejesummen forøges betydeligt ved opførelse af huse og hoteller.
- For at skaffe flere penge kan man pantsætte grunde til banken.
- Felterne "Prøv lykken" giver ret til at trække et kort, hvis ordre derefter skal følges.
- Somme tider kommer en spiller i fængsel.
- Spillet er fuld af spekulation og spænding, og auktionsholderen kan ofte bidrage til at forøge denne.

Indhold

Spilleplade
6 biler
2 terninger
Skøder
Prøv lykken-kort
Huse og hoteller
Pengesedler



Forberedelser

En af deltagerne vælges til at være bankør. Bankøren giver hver deltager 30.000 kr. fordelt således:

2 stk. 5.000 kr., 5 stk. 2.000 kr., 7 stk. 1.000 kr.,
5 stk. 500 kr., 4 stk. 100 kr. og 2 stk. 50 kr.

Banken beholder resten af pengene samt skøderne, de grønne huse og de røde hoteller. Gennem banken foregår alle spillets ud- og indbetalinger undtagen leje, der betales til ejeren, samt handel med skøder og løsladelseskort, der foregår blandt spillerne indbyrdes.

Prøv lykken-kortene lægges i en bunke midt på spillepladen med bagsiden opad.

Bilag 3.2 : Regelsæt

Selve spillet

Deltagere stiller deres bil på feltet "START" og bliver enige om, hvem der begynder. Spillet fortsætter derefter i urets retning.


Første spiller kaster begge terninger og flytter sin bil så mange felter frem, som øjnene viser. Når spilleren har benyttet retten eller opfyldt pligten, som feltet angiver, går turen videre til næste spiller. Hver gang man passerer "START", modtager man 4.000 kr. fra banken.

Lander man på et felt med "**Prøv lykken**", skal man tage det øverste kort i bunken med Prøv lykken-kort og rette sig efter ordlyden på det. Når et kort er benyttet, lægges det tilbage nederst i bunken.

Lander man efter et terningkast eller ifølge ordren på et af Prøv lykken-kortene på en grund eller virksomhed, der ikke ejes af nogen anden deltager, kan man købe denne af banken for den pris, der står på feltet, og man får så udleveret skødet, der lægges med forsiden opad foran spilleren. Efter de takster der står på skødet, kan man nu opkræve leje af de spillere, der lander på ens grund. Køber man ikke skødet, sætter banken det straks på auktion, og denne har alle lov til at deltage i.

Lander man på feltet "De fængsles", skal man gå direkte i fængsel og modtager ikke de 4.000 kr. for at passere "START". Lander man derimod på feltet "I fængsel", er man blot på besøg og fortsætter næste gang uden straf.

Indkomstskatten har man lov til at betale med 4.000 kr. Men man kan også betale 10% af sine værdier: Kontanter, bygninger og den trykte pris for grunde og virksomheder (også pantsatte). Spilleren skal vælge betalingsmåden, inden han tæller sine værdier sammen.

Man får et **ekstrakast**, hvis man kaster 2 af samme slags (f.eks. 2 femmere), og man skal rette sig både efter forskrifterne for det felt, man lander på efter første kast og efter ekstrakastet. Kaster man 3 gange i træk 2 af samme slags, må man ikke flytte tredje gang, men skal gå direkte i fængsel. 

Feltet med **Parkering** er et fristed, indtil man skal kaste igen.

Bilag 3.3 : Regelsæt

Man kommer ud af fængslet på en af følgende måder:

- 1) Ved at betale en bøde på 1.000 kr., inden man kaster terningerne.
- 2) Ved at benytte et af løsladelseskortene fra bunken med Prøv lykken-kort.
- 3) Ved at kaste 2 af samme slags. Man flytter så straks det antal felter frem, som øjnene viser, og har alligevel ekstrakast.

Man kan ikke blive i fængslet mere end tre omgange. Får man ikke to af samme slags, når man kaster tredje gang, må man betale bøden på 1.000 kr. og flytte, som øjnene viser. Er man i fængsel, har man stadig ret til at købe grunde (ved auktion eller handel spillerne imellem), men man kan ikke opkræve leje af de andre spillere.

Huse og hoteller

Ejer man alle grundene i samme farve, får man dobbelt leje af de ubebyggede grunde og har ret til når som helst at bygge huse, der købes hos banken til den pris, der står på skøderne.

Der skal bygges jævnt, dvs. at man kan opføre det første hus på den grund i gruppen, man ønsker, men inden hus nr. 2 opføres på en grund, skal der være bygget ét hus på hver af de andre grunde i gruppen osv.

Inden man opfører et hotel, skal der være fire huse på hver grund i gruppen.

Der må kun bygges ét hotel på hver grund. Når man køber et hotel, afleverer man de fire huse til banken.

Banken skal, når som helst man ønsker det, tage bygningerne tilbage til halv pris. Prisen for et hotel er fem gange prisen for et hus.

Har banken ingen bygninger, når man vil købe, må man vente, til der kommer nogle tilbage. Er der flere, der vil købe, og har banken ikke nok til alle, sætter banken de huse, der er, på auktion.

Indbyrdes handel med ubebyggede grunde etc. er spillerne tilladt til den pris, de kan blive enige om.

NB! Har man bygget, skal man sælge bygningerne tilbage til banken, inden man kan afhænde nogen grund i den pågældende gruppe.

Bilag 3.4 : Regelsæt

Pantsætning

- Man kan kun pantsætte sine ubebyggede grunde etc. til banken for det beløb, der står trykt på skøderne. Har man bygninger på grundene, skal man først sælge disse til banken. Spilleren beholder skødekortene, men vender bagsiden opad. Renten er 10% (der rundes op til nærmeste 100 kr.), og renten betales sammen med lånet, når pantsætningen hæves.
- Hvis en pantsat ejendom sælges, og køberen ikke straks hæver pantsætningen, skal han alligevel betale 10%, hvis han senere hæver pantsætningen.
- Af pantsat ejendom kan der ikke opkræves leje.
- Banken giver kun lån mod pantsætningssikkerhed.
- Pantsætning af grunde samt handel med bygninger sker kun gennem banken.
- Spillerne må ikke låne indbyrdes.

Glemmer man at opkræve leje af en medspiller, har man tabt sin ret, når spiller nr. 2 efter vedkommende har kastet terningerne.

Fallit. Skylder en spiller mere, end han ejer, skal han overdrage alt til sin kreditor efter at have solgt eventuelle bygninger tilbage til banken, og han må derefter udgå af spillet.

Er det banken, der er kreditor, sælger bankøren straks modtagne grunde på auktion.

Hurtigt spil

Bankøren blander skødekortene og giver hver spiller to skøder, for hvilke han modtager den trykte pris.

Der bestemmes en spilletid, og når tiden er gået, har den spiller vundet, som har størst formue.

God fornøjelse!

Bilag 4.1 : Lykkekort

Priser stiger

1. Oliepriserne er steget, og De skal betale kr 500 pr hus og kr 2000 pr hotel
2. Ejendomsskatten er steget. Ekstraudgifterne er: 800 kr pr hus, 2300 kr pr hotel.

Betal penge til banken

3. De har kørt frem for "fuldt stop", Betal 1000 kroner i bøde
4. Betal for vognvask og smøring kr 300
5. Betal kr 200 for levering af 2 kasser øl
6. Betal 3000 for reparation af deres vogn (antal kort: 2)
7. De har købt 4 nye dæk til Deres vogn, betal kr 1000
8. De har fået en parkeringsbøde, betal kr 200 i bøde
9. Betal deres bilforsikring, kr 1000
10. De har været udenlands og købt for mange smøger, betal kr 200 i told.
11. Tandlægeregning, betal kr 2000.

Modtag penge

12. De har vundet i klasselotteriet. Modtag 500 kr. (Antal kort: 2)
13. De modtager Deres aktieudbytte. Modtag kr 1000 af banken (antal: 3)
14. Kommunen har eftergivet et kvartals skat. Hæv i banken 3000 kr.
15. De have en række med elleve rigtige i tipning, modtag kl 1000
16. Grundet dyrtiden har De fået gageforhøjelse, modtag kr 1000.
17. Deres præmieobligation er udtrykket. De modtager 1000 kr af banken. (antal 2)
18. De har solgt nogle gamle møbler på auktion. Modtag 1000 kr af banken.
19. Værdien af egen avl fra nyttehaven udgør 200 som de modtager af banken
20. De modtager "Matador-legatet" på kr 40.000, men kun hvis værdier ikke overstiger 15.000 kr

Få penge fra spillere

21. Det er deres fødselsdag. Modtag af hver medspiller 200 kr.
22. De har lagt penge ud til et sammenskudsgilde. Mærkværdigvis betaler alle straks. Modtag fra hver medspiller 500 kr.
23. De skal holde familiefest og får et tilskud fra hver medspiller på 500 kr.

Ryk felt

24. Ryk frem til START (antal 2)
25. Ryk tre felter frem
26. Ryk tre felter tilbage (Antal kort: 2)
27. Ryk frem til Frederiksberg Allé. Hvis De passere START, indkasser da 4000 kr.
28. Ryk frem til det nærmeste rederi og betal ejeren to gange den leje han ellers er berettiget til, hvis selskabet ikke ejes af nogen kan De købe det af banken.(Antal 2)
29. Tag med Mols-Linien, flyt brikken frem og hvis De passerer START indkassér da kr 4000.
30. Ryk frem til Grønningen, hvis De passerer start indkasser da kr 4000
31. Ryk frem til Vimmelskæftet, hvis de passerer start indkasser da kr 4000
32. Tag med den nærmeste færge, hvis de passerer start indkasser da kr 4000
33. Ryk frem til Strandvejen. Hvis De passere START, indkasser da 4000 kr.

Bilag 4.2 : Lykkekort

34. Tag til Rådhuspladsen

Noget med fængsel

35. I anledning af kongens fødselsdag benådes De herved for fængsel. Dette kort kan opbevares indtil De får brug for det, eller De kan sælge det. (antal kort: 2)

36. Gå i fængsel, De indkasserer ikke 4000 kr for at passere start. (antal kort: 2)