

## Bjarnes notes for Matching game chapter 3

### 1. Card movieclip with different stats

linkage properties => Export for actionscript ( Card class )

### 2. Putting the cards to the stage

( see the external as file )

#### a. Loops

```
0 => boardWidth = 2 //horizontal
```

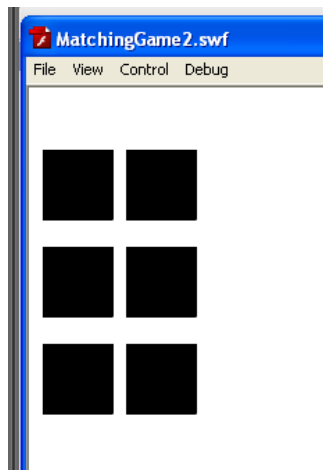
```
0=>boardHeight = 3 // vertical
```

```
private static const boardWidth:uint = 2;
```

```
private static const boardHeight:uint = 3;
```

Using the debugger to display the value of the variables. !

Stepping through the loops



```
var thisCard:Card = new Card(); // Here we construct a new instance of the Card class
thisCard.stop(); //show backside of card
thisCard.x = x*cardHorizontalSpacing+boardOffsetX; //placement
thisCard.y = y*cardVerticalSpacing+boardOffsetY;
addChild(thisCard); // Add the Card to the displaylist = stage
```

### 3. Raffle the cards a place on the board

#### Using an Array

```
// make a list of card numbers
```

```
var cardlist:Array = new Array();
```

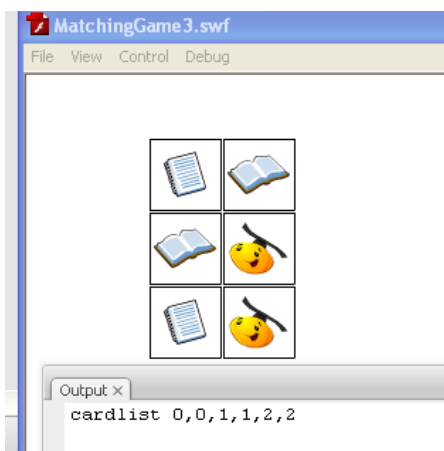
```
cardlist.push(i);
```

```
var r:uint = Math.floor(Math.random()*cardlist.length); // get a random face
```

```
c.cardface = cardlist[r]; // assign face to card
```

```
cardlist.splice(r,1); // remove face from list
```

```
c.gotoAndStop(c.cardface+2);
```



#### 4. Simple Interactivity

```
// player clicked on a card
public function clickCard(event:MouseEvent) {
    var thisCard:Card = (event.currentTarget as Card); // what card?
    trace(thisCard.cardface);
}
```

#### 5. Is there a card match ?

```
                private var firstCard:Card;
                private var secondCard:Card;

if /else tests

exercise change
// compare two cards
if (firstCard.cardface == secondCard.cardface) {
// remove a match
//removeChild(firstCard);
//removeChild(secondCard);
firstCard.alpha = .2;
secondCard.alpha = .2;
```

#### 6. Game Over

```
if (cardsLeft == 0) {
    gotoAndStop("gameover");
}
```

#### 7. Counting Score , textField

```
                public function showGameScore() {
                    gameScoreField.text = "Score: "+String(gameScore);
                }
```

#### 8. Timer, enterframe event makes the loop.

```
                public function showTime(event:Event) {
                    gameTime = getTimer()-gameStartTime;
                    gameTimeField.text = "Time: "+clockTime(gameTime);
                }
```

## 9. Adding sound and flip animation done with actionscript.

```
// set up sounds
var theFirstCardSound:FirstCardSound = new FirstCardSound();
var theMissSound:MissSound = new MissSound();
var theMatchSound:MatchSound = new MatchSound();
```

Class diagram

<i>Class name</i>
<b>Card10</b>
<i>Properties (instans variables )</i>
private var flipStep:uint; private var isFlipping:Boolean = false; private var flipToFrame:uint;
<i>Methodes</i>
public function startFlip(flipToWhichFrame:uint) public function flip(event:Event)

## Exercises:

1. Change the constant of boardWidth a and boardHeight to 2 and 3 in the matching games examples and test the result.

```
private static const boardWidth:uint = 2;
```

```
private static const boardHeight:uint = 3;
```

Try to explain what is going on in the examples to a fellow student.

Try to change some of the constants and see what happens.

2. **Convert game into a math game for children with 3 relations (e.g. 1,I,2,II,3,III )**
3. Feedback on case 1 requirement specifications.