

# Sem1\_programming\_exam

This project is a simple breakout game. It's made for an exam for introduction to programming.

The program is made using Processing.

It was created in November 2018

In the game there is only one level, though throughout the game the ball will keep gainig speed when it hits a block, which makes it difficult to reach the max. of points. At least I haven't reach the max. score while testing it

## the game

in the game the ball, curser and points, are curently fill with color by using noise, which means the color is switch while playing.

to move the curser you have hover the game and just move the mouse.

## How to use (code)

the game is made with 4 classes (the score(class\_score), the ball(class\_ball), the curser(class\_bottom\_rect) and the blocks(class\_top\_rect)).

The color scale used in the program is HSB

All the functions is initialized on the first page in setup where the classes are called and in draw where the functions from each class is used.

The blocks in the top is made with a 2-dimensional array to make both row and column and then made with two for loops one for row and one for column:

```
```processing
for(int i=0; i<block.length; i++){
for(int i0=0; i0<block[i].length;i0++){
block[i][i0]=new blocks(blockX+i*(blockW+5),blockY+i0*blockH,blockW,blockH);
}
}
```
```

To detect when the ball is colliding with either one of the blocks or the curser there is used a couple of if statements. To make the the blocks in the top to dissapear when the ball hits there is used a boolean which is set to false when it hits the block:

```
```processing
```

```

//hit the sides of the block
if((ball0.y > blockY) && (ball0.y < blockY + blockH) && (ball0.x - ball0.r/2 <= blockX +
blockW) &&
(ball0.x-ball0.r/2>blockX) && (G==true)){
    G=false;
    ball0.speedx=-ball0.speedx;
    score0.point+=1;
    ball0.speedy+=1;
    ball0.speedx+=1;
} else if ((ball0.y > blockY) && (ball0.y < blockY + blockH) &&
(ball0.x + ball0.r/2 < blockX + blockW)&&(ball0.x+ball0.r/2>=blockX)&&(G==true)) {
    ball0.speedx=-ball0.speedx;
    G=false;
    score0.point+=1;
    ball0.speedy+=1;
    ball0.speedx+=1;
}

```

```

When the either the points hit 50(since this is the number of blocks visible when starting the game) or the ball falls down below the height of the screen there will appear a Game Over screen, and when the mouse is pressed the game will start over and reset everything:

```

```processing
void GameOver() {
    if (score0.point==50 || ball0.y>height) {
        textAlign(CENTER, CENTER); // makes the text aligne in the center of the text
        text("Game Over", width/2, height/2);
        text("click to start over", width/2, height/2+40);
        text(score0.point, width/2, height/2-40);
        score0.playing=false; //sets the score to false so it won't show when Game Over
    }
}

```

```

ball0.valid=false; //sets the ball to false so it won't show when Game Over
curser0.moving=false; //sets the curser to false so it won't show when Game Over
} else if (mousePressed) { // if the mouse is pressed while Game Over the following will happen
score0.playing=true; //the score will be shown again
G=true; //all the blocks will be shown again
ball0.valid=true; //the ball will be shown again
curser0.moving=true; //the curser will be shown again
ball0.y=height/2; //the ball will be set to start in the middle
score0.point=0; //the point will reset to zero
ball0.speedx=ball0.startSpeedx; // the speed of the ball will be set back to the start speed
ball0.speedy=ball0.startSpeedy; // the speed of the ball will be set back to the start speed
}
}

```

...

## ## problems

### ##### Current issues

There is a bug when the ball hits the corners of the curser(block in the bottom). When the ball hits the corners it will be stock here until you again move the curser, and the ball will mostly fall down.

### ##### Solved issues

throughout writing the code I accoutered a problem when I had to make the row and columns for the blocks in the top, this took some time to solve. It was solved with making a 2-dimensional array  
`blocks [][] block= new blocks[10][5];`.

There was also a smaller dificulty when making the collision between the ball and the blocks, this was solved with help from another student.

## ## run it

To run this game you need to have processing.

When open in processing press run to play the game.

To run it you will also need the mp3 file: effect0, which is inside the data folder. The other files in the data folder is not currently used.

You will also have to download the sound library in processing to run.