

Aflevering 1

Lauritz Andersen, Helga Rykov Ibsen og Jonas Kramer

September 13, 2021

[Link to assignment 1](#)

[Link to assignment 2](#)

1 Assignment 1g0

In our first assignment, we were given the task to design a small, fun program using only 10 predetermined blocks. We were allowed to use any block as many times as we would like, and were even allowed to exclude some entirely. Our initial idea was to have a cat as a sprite, that, when clicked, would hide for a few seconds, reappear and jump close to the screen (our face) and say "Give me food, peasant!", because... well, it's a cat. Considering the simplicity of this task, we did not run into too many problems, if any at all. Ideally, we would have liked the "Give me food, peasant!"-bubble to appear right as the "meow"-sound would play, and not a second after, but since it was such a minor problem, we decided to ignore it and instead turn our attention to the second assignment.



Figure 1

2 Assignment 1g1

2.1 First brainstorm

In our second assignment, we were tasked with making a small game. The game had to include 2-5 sprites, have a typical gameplay of about a minute and at least 1 variable had to be included in our code. We started out by brainstorming a

very simple concept, where we would move our sprite, a cat, with the arrow keys and have to catch moving objects from above. If the object was not caught, it would touch the ground, which would end the game and show a "GAME OVER"-backdrop. We made a first draft on paper drawing how we imagined the game to play out. Then, we turned our attention to the actual coding. The initial code was fairly simple to write, although we ran into a problem, where an object wouldn't be collected upon collision, but instead we had to resort to pressing to space-key. We quickly figured out how to make the object immediately disappear when touching our sprite by simply putting our code into a "forever"-loop.



Figure 2

2.2 Variables

To make the game meaningful, we initially introduced 2 variables: Point and Timer. The former's purpose was to keep a count of the amount of collisions/catches, adding 1 point per catch up until a maximum of 15 in which case you would win the game. The latter variable was used to control the overall length of a single round. If a player went a full minute without the ball dropping to the ground, the timer would reach zero, and the game would be over. Later on, we added a third variable, speed, to control the speed of our objects falling from the sky. This process was fairly simple and did not cause us much trouble.

2.3 Final iteration

From here, we decided to add a bit more depth to the game by introducing two more sprites, a bat and a dog, both of which would also fall from the sky. However, instead of awarding a point when colliding, they would instead subtract points (the bat subtracts 5, the dog 2), thus forcing the player to have to avoid them while still catching the initial object. All of this was very simple to code, since we more or less copied the code from our first object, only changing certain conditions. Lastly, we changed our initial object to instead be a mouse, and we added one more condition: if the player caught enough

mice, and the timer reached 15, the player would win the game and a "YOU WIN"-backdrop would pop up. By doing so, we added a game with atleast a little bit of depth, where it was possible to both win and loose. In the end, the brilliant and revolutionary game, "It's Raining Bats and Dogs", was born.



Figure 3



Figure 4. Active state.

2.4 Final problems

While the overall code was very simply to make, we did run into one problem that we did not solve. Whenever the "GAME OVER"-backdrop appeared, we wanted the entire game to fully stop, but we could not figure out a way to prevent us from being able to move the cat while the game was not in play.