

## Laboratory: Game

In this laboratory, we are learning how to use tkinter in order to do simple animation.

For any game you develop, you should consider at least two different classes. One for the abstract game that among other things contains the Physics engine and another one for the Graphical User Interface.

We start out with a somewhat function game consisting of two classes: Ball for the physics engine and GUI for the GUI.

The physics engine models a ball subject to acceleration controlled by the keys on the left of your key-board. We assume friction which will eat up velocity and acceleration. The constructor will set random positions and velocities. Per time step, the position is changed by the velocity and the velocity is changed by the acceleration. To model friction, acceleration and velocity are also reduced by 5% each time step. There are also functions to increase the acceleration in horizontal and vertical directions.

The GUI contains two new elements. First, we use bind in order to bind keystrokes to the window. The bind function is connected to the key-method that processes the bound event, in this case the pressing of a key.

The animation follows a simple scheme. We first update the ball. We then delete everything from the canvas. (Better gaming interfaces have much more sophisticated versions of this.) We then paint the ball, update the canvas, and go to sleep of 1/10th of a second.

Your tasks:

1. The Ball class limits the acceleration to a maximum in x and y direction. This is not good, since a ball can have a bigger acceleration in diagonal direction than in vertical or horizontal direction. The absolute size of the acceleration is  $\sqrt{accx^2 + accy^2}$  and we can insure by multiplying with the right scalar that the absolute size of the acceleration never exceeds 2. Make this change in the Physics engine.
2. Currently, the roles of the z-key and the w-key are reversed. Unreverse them.
3. Create randomly a green rectangle to which the player has to steer the ball. If the player succeeds, then the rectangle moves somewhere else in the canvas.
4. Add labels that display the current absolute size of the velocity and acceleration of the ball.