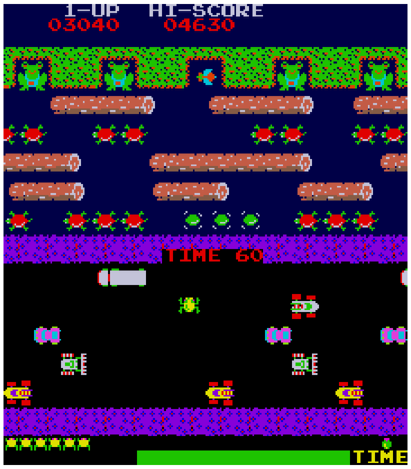
Programming for Artists and Designers
Wintersemester 2023/2024

Final Project - Frogger



Tutor: Aaron Kammer, aaron.jordan.kammer@uni-weimar.de

Submission

The assignment is due on **01.03.2024 at 23:59**. Compress the **directory** of your solution to a **.zip** file and upload it on **Moodle**. Please start working on the project as soon as possible and ask questions via email, if you have any.

I will also offer consultations via Zoom to answer questions about the project. Exact dates will be announced on Moodle.

Description

In this project you will implement the game Frogger using the **StdDraw** library. The player plays as a frog controlled with the mouse that can jump up, down, left and right. The frog must cross a multi-lane road without being run over by cars. If the frog reaches the other side of the screen, the player wins the game.

Try to break down complex tasks into smaller functions to make your code more readable and easier to debug. Stick to the core mechanics of the game but feel free to get creative and add your own flair to the game.

1 The Road (35%)

On the screen you will have a road with multiple lanes. In each lane there will be cars driving from left to right or from right to left. The cars in one lane all have the same size, speed and space to the next car, but these properties are randomized for each lane.

a) Create a class **Car** with that can fulfil the following tasks:

   a1) Draw a car on the screen at a position with a rectangle and color of your choice.

   a2) Let the car move with a given speed on the x-axis.

b) Create a class **Lane** that can fulfil the following tasks:

   b1) Draw a lane on the screen with a position and a given width and height.

   b2) Create a list of cars on the lane. Make the lane automatically fill up with cars based on a car length, speed and spacing between the cars given as parameters in the constructor.

   b3) Create a function to move all cars on the lane simultaneously.

   b4) Teleport cars back to the other side of the screen if they reach the end of the lane (Hint: To make the teleport less noticeable, the lanes can be made longer than the screen width.)

c) Create a list of lanes arranged next to each other with random car sizes, speeds and spacing for each lane and

d) continuously draw them on the screen and move the cars.

2 The Frog (35%)

The frog is controlled with mouse clicks. It can move in the four directions up, down, left and right.

a) Create a class **Frog** that can draw a frog on the screen with a position and a geometric shape of your choice.

b) Write a function to detect mouse clicks and move the frog one step in the direction of the mouse position relative to the frogs position.

c) Extend the **Car** class with a function to check if collides with the position of the frog.

d) Create a function for your game loop to check for collisions between the frog and the cars in the road in each frame draw.

e) Display a number of lives of the frog on the screen, starting with 3 lives.

f) Teleport the frog back to its bottom of the screen if it collides with a car and remove one life.

g) End the game if the frog has no more lives left.

3 The Goal (15%)

The goal of the game is to get the frog to the other side of the screen.

a) Create an area on the top of the screen that represents the goal.

b) Check if the frog reaches the goal. If it does, let the player know that they won, and end the game.

4 Polishing (5%)

Replace the geometric shapes of the game objects with small images and display them with `StdDraw.picture(...)`.

5 Comments (10%)

Add comments to your code to explain what your code does. You don't have to comment every line of code, but you should add comments to explain the purpose of each class and function.