Introduction to Visual Computing

-

Project

Group AA
Jonathan Link - Romain Gallay - Céline Dupuis

12.06.2015

Game Calorie King

Eat tons of burgers to become the King!

Settings When you launch the programme, you can choose to play either a virtual game with the mouse or a tangible game with the webcam.

Mode Once you have chosen the way you will interact with the game (settings), you can select between two modes.

- Classic mode
  A classic tray meal will show up, with food set by default. You can add new burgers, drinks
  and french fries in the edit window (`shift` key). Each time you bounce back foods, you win
  calories. On the other hand, try not to touch the tray borders as it will make you lose weight.

- Eat All mode
  As its name suggests it, the Eat All mode consists of eating the foods instead of bouncing
  back from them. This mode was initially made for the tournament; the tray is thus bigger
  and all the foods have the same radius. We offer the possibility to add the foods before the
  game begins, which allowed the referee to add the obstacles at strategic places during the
 tournament.

Keyboard functionalites

Key	Action
M	back to Menu
R	Reset the position of the tray
S	Switch between *mouse* and *webcam*
shift	edit window to add obstacles

Blob Detection

If you choose the webcam to play, you can add burgers with blob detection. When pressing the
`shift` key, the edit window displays. You can either add foods with the mouse or do the following
steps:

1. Place your 3D-printed burger on the lego board (the detection works for brown object then)

2. Place the lego board orthogonally to the webcam

3. Press the 'Blob ON/OFF' button to run the blob detection

The following images show the edit window. On the upper left of the window, we can see the blob
detection in action. First we place ourself in front on the webcam and then the programme will
detect the burger (in blue), to finally add it on the tray (in orange).
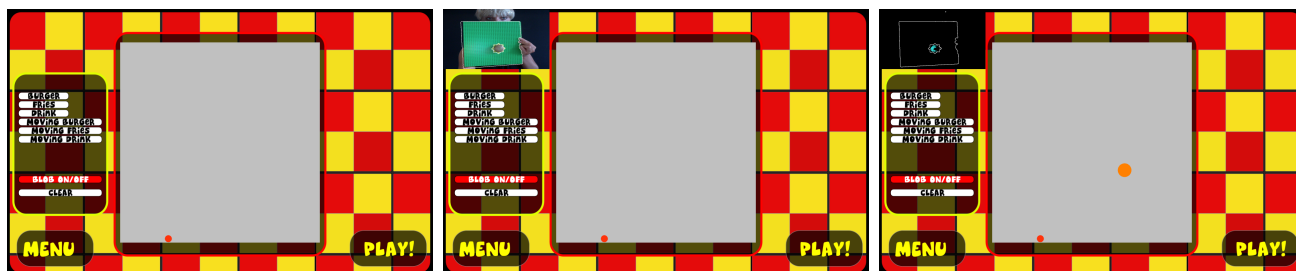
Note for Milestone 4

- In order to test the tangible game without launching the camera, we added a DEMO tag in our code which allows to load a video instead of the camera stream. Simply run the programme given on Github and test the game with the following setting/mode:

$$\text{Webcam} \rightarrow \text{Play} \rightarrow \text{Classic or Eat All}$$

  If you still want to try the interaction with the lego board, you have to set the DEMO tag to false in `app.controllers/PlateController.java` (line 32) and adapt the thresholds in `app.imageProcessing.solver/ImageProcessingSolver.java` (line 54). We added *TODO* such that you can easily find these two tasks in our code.

- Please note that we do not have any classe named *TangibleGame*. Indeed, our architecture being quite big, we didn't want to change our class names at that point. If you want to have a look, the main implementation of the tangible game is contained in `app.imageProcessing/` `Webcam.java`. You can find an UML diagram of our implementation on

$$https://documents.epfl.ch/users/l/li/link/public/UML.jpg$$

- If you play with the webcam setting, stay reasonable in terms of obstacles added. We optimized the image processing pipeline to gain some milliseconds in runtime and avoid lag, but the display of our game still uses a lot of ressources. If you want to test the features of our game and add a lot of foods on the tray, we suggest to rather use the Mouse setting. It will offer you a smoother user experience.

Have fun !

Gallery