Worms:

Event-Oriented Programming

The aim is to create a program that simulates the movement and jump of two Worms following the conditions detailed below:

1. Both Worms are initially positioned on the horizontal that we are going to take as the floor y = 616px. Its position in x must be such that 701px> X> 1212px. Within these parameters each group can choose the initial position and orientation of the two Worms as they prefer.
2. One of the Worms responds to the keys "LEFT arrow" to move left, "RIGHT arrow" to move right, and "UP arrow" to jump.
3. The other Worm responds to the keys "A" to move left, "D" to move right and "W" to jump.
4. The user can press any of these keys at any time.
5. If two different Worms are commanded, for example "W" and "LEFT arrow" are pressed at the same time, both Worms must respond at the same time (following the example one of the Worms jumps and the other moves toward the left).
6. If two keys are pressed on the same Worms, priority will be given to one of them, following some strategy, for example the one that the system sees first and the second will be ignored until the condition of two keys on the same worms pressed simultaneously.
7. The movement of each Worms is governed as follows:
    a. The Worms moves at a rate of 27 pixels per second while the player keeps any of the "LEFT arrow", "RIGHT arrow", "A" or "D" keys pressed.
    b. In order for the worm to start moving, the player must keep the key pressed for more than 100ms. If the key is pressed for less time, the machine understands that the user wants to turn the selected worm on itself if it were the reverse arrow and it is ignored in another case.
    c. The machine will take offsets multiples of 27 pixels. Therefore, after understanding that the user wants to make a movement, only at 900ms will check again if the arrow is pressed for 100ms to move the worm again. If it is not, the movement is terminated. If it were, it would continue.
    d. To get more info: http://worms2d.info/Phases_of_Worm_walking

8. We will only take horizontal displacements in the scenario provided by the chair. Worms can be moved at Y = 616px between 701px> X> 1212px.
9. The jump is calculated with an initial velocity equal to 4.5. It is considered that the worm jumps with an angle of 60 ° (therefore an initial velocity on the vertical axis ("Y") results from the jump of sin (60 °) * 4,5). We will only allow "forward jumps" so the worm jumps in the direction it is pointing.
10. While the Worms is jumping (the duration of the jump is determined from the oblique shot equations) the state of the key that commanded said jump is ignored. Once the jump is over, it is reconsidered.
11. Vale aclarar que tanto el salto como el movimiento tienen un preámbulo conocido como Start-up en donde no se realiza ningún movimiento que se va a discutir en clase. También cuenta dentro del tiempo de salto. Por el contrario, el Start-up está incluido en el tiempo de desplazamiento de 1s.
12. We will take g=0.24.
13. The angles are measured from the horizontal line representing the floor where the worm rests to the vertical in the direction of the worm. If the angle is negative, it is measured below the worm and if it is positive above it.
14. Speed is measured in pixels per frame, acceleration in pixels per frame per frame.
15. The refresh rate is fixed at 50FPS (frames per second).
16. To represent the Worms we will use the images provided by the chair.