Logic Design

Project Gate Check #3

Lunar Lander Solution

(25 pts)

main

- Call config
- REPEAT until landing is true or crash is true or ESC button pressed
  - Call erase_objects
  - Call move_objects
  - Call draw_objects
  - Call process_controls
  - Call detect_collisions
- Call game_over

config

- Setup graphics window dimension variables, graphics window title variables, and open window accordingly
- Initialize variables to include:
  - Lander's initial position/location variables, speeds, etc.
  - Other Required variables
  - Some have random components e.g. landing pad location
- Call generate_landscape

generate_landscape (procedure)

- Initialize variables for generating the landscape
  - step_max to 2.6
  - step_change to 0.9
  - height_max to window height * 3 / 8
  - height to random * height_max
  - slope to 0
  - x_location to 1
- Start with the left edge of the screen and REPEAT until right edge reached (x_location starts at 1 and goes to the width of the screen)
  - If you are drawing a line that is part of the landing pad do option 1 otherwise do option 2
    - Option 1 - Landing pad
      - Draw line at a fixed height for mountain under landing pad
      - Draw line at a fixed height for the landing pad
    - Option 2 - Mountain
      - Calculate a new slope using slope <- slope + (random * step_change * 2 – step_change)
      - Calculate a new height using height <- height + slope
      - If slope is greater than step_max set it to step_max
      - If slope is less than -step_max set it to -step_max
      - If height is > height_max set it to height_max and change sign of slope
      - If height is < 1 set it to 1 and change sign of slope
      - Draw a line from 1 to generated height at current x_location
    - Assign mountains array at x_location the generated height

erase_objects

- Call erase_lander
- Call erase_thrusters

erase_lander

- Draw an object to erase the lander's current position

erase_thrusters

- Draw objects to erase any thrust at the current position

draw_objects

- Call draw_lander
- Call draw_thrust

draw_lander (procedure)

- Draw the bitmap loaded for the lander

draw_thrusters

- Complete following steps using lander's current position as reference:
  - If variable <up_thrust> is true draw graphics (lines, shapes, images) to represent up thrust
  - If variable <right_thrust> is true draw graphics (lines, shapes, images) to represent right thrust
  - If variable <left_thrust> is true draw graphics (lines, shapes, images) used to represent left thrust

process_controls

- If "Up" key down, set variable <up_thrust> to true; otherwise set it to false
- If "Right" key down, set variable <left_thrust> to true; otherwise set it to false
- If "Left" key down, set variable <right_thrust> to true; otherwise set it to false
- If "Esc" key down, set variable <quit_game> to true; otherwise keep it set to false

move_objects

- If variable <up_thrust> is true, modify deltaY to move the lander up using formula <deltaY <- deltaY + 0.02> otherwise use "deltaY <- deltaY - 0.01" (this is gravity's impact on the lander)
- If variable <right_thrust> is true, modify deltaX to move the lander left using formula "deltaX <- deltaX - 0.01"
- If variable <left_thrust> is true, modify deltaX to move the lander right using formula "deltaX <- deltaX + 0.01"
- Update position of lander using formulas "landX <- landX + deltaX" and "landY <- landY + deltaY"
- Call wrap_lander

wrap_lander

- If the new position of the lander is off the left or right side of the screen, reset its position to the left or right side of the screen so that it appears that the lander goes off the screen wraps

detect_collisions

- Initialize variables
  - Set landing to false
  - Set crash to false
  - Set x_position to landX
- REPEATEDLY check the mountains array underneath the lander to see if it has collided with the mountain or landing pad (x_position starts at landX and goes to landX + width of the lander)
  - If the lander is above the mountains do nothing (no collision) otherwise (collision) do the following
    - If the lander is not completely on the landing pad set crash to true and stop checking for a collision
    - If the lander is completely on the landing pad going too fast set crash to true and stop checking for a collision
    - If the lander is completely on the landing pad and not going too fast set land to true

game_over

- If landing is true display a successful landing message
- If crash is true display a crashed message
- Display a message telling the user to click the left mouse button to exit
- If the left mouse button is clicked exit the game