**COMPUTER GRAPHICS AND ANIMATION**

**PROGRAMMING ASSIGNMENT 17-CS**

**Sprite Animation**

Create an application that simulates Crystal Snail in action. The program must be able to do the following:

* Provide a background.
* Start with Crystal Snail doing his intro animation.
* Allow the user to control Crystal Snail. Crystal Snail can do the following actions:
  + Jump up. This can only be done if Crystal is on the ground.
    - Crystal Snail will jump to a certain height.
    - When it reaches said height, Crystal Snail hides in his shell and starts spinning in mid-air.
    - Crystal Snail will spin indefinitely until the user issues another command.
  + Fly in shell. This can only be done if Crystal Snail is spinning mid-air.
    - The user clicks at a target point.
    - Crystal Snail will stop spinning (he is still hiding in his shell) and fly towards the tartget point.
    - Make the shell face the proper direction when flying.
    - Upon reaching the target point, Crystal Snail starts spinning in mid-air again.
    - Crystal Snail will spin indefinitely until the user issues another command.
  + Slow down time. This can only be done if Crystal Snail is spinning mid-air.
    - Crystal Snail stops spinning, comes out of hiding, and charges his antennae, causing time to slow down.
    - The screen warps when Crystal snail is slowing down time.
    - While the time is slowed down, the speed of Megaman dummy’s movements (which will be explained later) are cut by half.
    - This effect lasts 10-15 seconds.
    - After charging his antennae, Crystal Snail retreats back into his shell and starts spinning in mid-air again.
    - Crystal Snail will spin indefinitely until the user issues another command.
  + Stop spinning. This can only be done if Crystal Snail is spinning mid-air.
    - Crystal Snail stops spinning, comes out of hiding, and descends to the ground.
  + Change facing direction. This can only be done if Crystal Snail is on the ground.
  + Shoot Crystal Hunter. This can only be done if Crystal Snail is on the ground.
    - Crystal Snail shoots a crystal projectile with a 45 degree elevation. The projectile travels in a parabolic path.
    - Crystal Snail shoots two extra projectiles: one projectile with a 60 degree elevation traveling in a parabolic path, another projectile travling horizontally.
    - The Crystal projectiles disappear when they hit the wall, the ground, or the Megaman dummy.
* Place a dummy, doll, or puppet of Megaman (or anything else that you like) as a target for Crystal Snail.
  + The dummy starts at a fixed location on the ground.
  + The dummy always walks back and forth from one edge to another.
  + The dummy can occasionally shoot projectiles. The projectiles move horizontally.
    - If Crystal Snail gets hit by a projectile shot by the dummy, he either staggers back or gets knocked out of his shell.
    - If Crystal Snail is knocked out of his shell, he cannot do anything other than jump or run towards his shell to reclaim it.
  + If the dummy collides with Crystal Snail when Crystal Snail is not hiding in his shell, Crystal Snail will stagger backwards.
  + If the dummy collides with Crystal Snail when Crystal Snail is hiding in his shell, the dummy will disappear.
    - After a few seconds, it will respawn at a random location on the ground.
    - The dummy must not collide with Crystal Snail when it respawns.
    - Display an animation of the dummy getting destroyed.
  + If the dummy gets hit by a crystal projectile, it will become frozen for a few seconds.
* Provide the proper animations for each action.
* Bonus points for user friendliness. Negative points for extreme user unfriendliness.

You must display Crystal Snail (and other sprites) and the background using the same picturebox.

You **may not** display Crystal Snail by putting it on a picturebox and overlaying it with the background on another picturebox. Doing this will result in rejection of your project.

You must use non-transparent image files for the sprites. You mat not use transparent PNG files or something similar to that.

You must create your own procedure/function to handle collision detection. You **may not** use the built in collision detection procedure/function provided by the programming language you use (if there is any).

What to submit:

* An executable file (EXE)
* The source code (+ modules or libraries if necessary)
* A report.

The report should contain explanations of at least the following:

1. Introduction.

* What is the program about?
* In what language is the program implemented?

1. Basic theory.

* Explain sprite animation.
* Explain how to put a sprite on a background.
* Explain character state diagram and state trasition table.
* Explain collision detection, especially the method of collision detection used in the program (but do not write something like “*this program* uses this method” or something similiar in this section).

1. Implementation

* Explain the main interface of the program.
* Explain every feature in the program and how to use them.

1. Design

* Explain the main data structures (if any) used in the program.
  + Explain how the background and sprite data are represented in the program.
* Explain the main/global variables used in the program.
* Explain the character state diagram and the state trasition table for Crystal Snail.
  + Explain each state in the state diagram.
* Explain the sprites used for the animation of each action (include the duration of each frame).
* Explain how the collision detection works in the program.
* Explain the pseudocode for each time tick.
* Explain how the bonuses (if done) are implemented.

1. Evaluation

* Evaluate the following test cases:
  + Starting the program (the intro animation).
  + Crystal Snail jumping up and spinning in mid-air.
  + Crystal Snail flying in his shell.
  + Crystal Snail slowing down time.
  + Crystal Snail shooting crystal projectiles.
  + The dummy walking back and forth.
  + The dummy getting hit by Crystal snail’s attacks.
  + The dummy getting frozen by a crystal projectile.
* Also perform a test case for all the bonus features you implemented.
* Include screenshots of each test case.
* Explain whether each case is successful.

1. Work log.

* Record the date and time of every moment you work on this assignment and job description of each member at each session. The work log should be a table with the following columns:
  + Date
  + Activity / progress
  + Personnel involved
* Write a summary of the implementation of each requirement given in the first page. For each requirement, explain whether that requirement is fully implemented, partially implemented, or not implemented at all. Give explanations if necessary.

1. Conclusion and remarks.

* Does the program work as expected?
* If some parts of the program do not work as expected, explain why.
* What are your comments about this assignment?

Submit the assignment no later than midnight, 3 May 2020, to [x60880@yahoo.com](mailto:x60880@yahoo.com).