Assignment 3 - Centipede (JS/HTML)

Start Assignment

Due Tuesday by 11:59pm **Points** 99 **Submitting** a file upload **File Types** zip
Available Feb 16 at 12am - Mar 22 at 11:59pm about 1 month

Introduction

This is second of two projects to help develop your game programming skills and knowledge in preparation for the final project. The purpose of this project is to expand the scope of techniques into a more sophisticated game, in preparation for the much larger scope for the final project.

You are going to create an HTML5 web-based version of a game that I used to play in my childhood in the arcades, Centipede. While the game difficulty would ramp up fairly quickly (for my skills), I enjoyed it a lot. One of the interesting aspects of the game was the use of a trackball to control player movement. We'll have to use keyboard (or controller) for that, but that works out okay.

The following link takes you to an AARP (yes, I think this is hilarious too) website that has a playable version of the original: **https://games.aarp.org/games/atari-centipede** (https://games.aarp.org/games/atari-centipede)

Gameplay, Visual, & Control Requirements

Use the following link for a very good description for how the game elements should work (it can also be used for art assets): **https://www.retrogamedeconstructionzone.com/2020/08/the-characters-of-centipede.html** (https://www.retrogamedeconstructionzone.com/2020/08/the-characters-of-centipede.html)

Must implement the Centipede game play, with the following key gameplay elements

- The Centipede
  - When a segment is hit, it turns into a mushroom
  - Specific head segment, then body segments for the rest of the centipede
  - When a centipede is split because of a bullet hit, in the original game it would create a new head segment for the split part. You do not need to do this, unless you want to.
- The mushroom play field
- Player movement, including limited vertical movement
- Spiders
- Fleas
- Scorpions and poisoned mushrooms and the effect they have on centipede segments

In addition to the visual elements required for gameplay, provide the additional visual elements

- Show remaining player lives left by drawing them at the top and left'ish part of the display area
- Show current score by showing it at the top and center of the display area
- When the game ends, display a "game over" message with the final score. Once acknowledged the game returns to the main menu

Menuing system that allows the player to...

- Start a new game
- View high scores
- Customize controls
- View credits (most important part of any game!)

Technical Requirements

- High scores must persist to the browser's local storage; keep at least the top 5 scores
- Must use sound effects; these sound effects should probably be very short
    - Effect upon player weapon fire
    - Effect upon player death
    - Effect upon centipede segment hit
    - Effect upon mushroom hit
    - Effect upon spider, flea, scorpion hit
- Animated sprites for...
    - Centipede
    - Spider
    - Flea
    - Scorpion
- Multiple destruction states for the mushrooms
- User ability to configure the controls. You may also implement the use of a controller, but only a warm fuzzy feeling inside for doing so, no extra credit.
    - Required Defaults
        - Arrow left - Move Left
        - Arrow right - Move Right
        - Arrow up - Move Up
        - Arrow down - Move Down
        - Space - Fire
    - The interface for this must present a screen where the name of the game control is displayed (move left, move right, move up, move down, fire) and to the right of it, the key used for the control. The user can select the action (by selecting the action with the mouse), then some visual will change to indicate it is possible to now select a new key, then the user presses the new key (doesn't have to be a combination, could be a single key) and that immediately becomes the new keyboard control for that game function.

  - The game only needs to support a single key for an action, does not have to allow for key combinations to perform an action.

Graphical Assets

You are free to theme the game if you like. I was able to find graphical assets at the following links; I'm sure there are other resources too:

- **https://www.spriters-resource.com/arcade/centipede/sheet/50437/** (https://www.spriters-resource.com/arcade/centipede/sheet/50437/)
- **https://www.pngkit.com/view/u2w7r5u2e6u2a9r5_general-sprites-centipede-arcade-game-sprites/** (https://www.pngkit.com/view/u2w7r5u2e6u2a9r5_general-sprites-centipede-arcade-game-sprites/)
- **https://www.retrogamedeconstructionzone.com/2020/08/the-characters-of-centipede.html** (https://www.retrogamedeconstructionzone.com/2020/08/the-characters-of-centipede.html)

Other General Comments

- When ESC is pressed during gameplay, the game must pause and give me the option to quit or continue. You are not required to provide this capability, but if you do, it should work like this.
- When pressing ESC in menu navigation, it should go back one level per press, not all the way back to the main menu. The probably doesn't apply for this game, menus are likely only one level deep.
- Be sure to scale the gameplay area based on the window size; If I change the window size (in code), your game should still render correctly.
  - You don't have to allow the window size to change during gameplay, just that it handles whatever size is set in code. You can expect reasonable window sizes to be tested, not unusual sizes.
- Allow the mouse to be used for menu navigation, keyboard is optionally nice, but mouse navigation is required.

Possible Development Strategy

The following is something to consider for a development strategy for this project; doing things in the order defined below. I will note that I haven't taken the time to do any architectural design for the application. A more thorough up-front consideration of the application design might suggest a different strategy.

- Develop an architectural design of the major code components, using UML. You should have learned this in a software engineering course, go ahead and use what you have learned :) You will more easily find success by doing some design work up-front, I want to very strongly emphasize the importance of some design work before starting on the code. The following are what I consider the major components (in no particular order)...
  - Menuing system
    - Biggest component, user ability to reconfigure input

  - Player
  - Centipede segments (remembering what I said in class about how to do this)
  - Mushroom
  - Flea
  - Spider
  - Scorpion
  - Gameplay arena, including the mushroom initialization
  - Bullets, or whatever it is the weapon fires
  - Animated Sprite support
  - Collision detection
- Get the menuing system and the player control customization done ***first***. Don't wait to do the player customization until later or toward the end, because you'll probably have to rewrite a lot of code, and it will be messy to do so. Get this done first and get it done right to set the tone for its use throughout. You will be tempted to work on the "game" first, don't fall prey to that temptation, trust me.
- Then develop the gameplay arena, along with the mushrooms and their placement
- Work on the player controlled character, getting the movement worked out, including collision detection with the mushrooms
  - At this time update the gameplay arena with the location of the remaining player lives and the current score
- Develop the weapon firing, including the damage states for the mushrooms
- Develop the flea and its dropping of mushrooms
- Develop the centipede and its movement
  - At this time you will also add in collision with the player's weapon bullets
- Develop the annoying spider
- Develop the scorpion and the poisoned mushrooms, including their effect on the centipede movement

The following are other items that need to be completed, in no particular order, could be afterwards or mixed in during the above:

- Persistence of the user control configuration
- Scoring system
- Persistence of the high scores

Assignment 3 - Centipede (JS/HTML)

Criteria	Ratings		Pts
Works on both Firefox and Chrome. 0 points if it only works on 1 of the browsers. Ideally it should work on all of MS Edge, Safari, Firefox, and Chrome.	**5 pts** **Full Marks**	**0 pts** **No Marks**	5 pts
Menuing New Game, High Scores, Customize Controls, Credits Use of mouse to select menu items	**5 pts** **Full Marks**	**0 pts** **No Marks**	5 pts
Technical - User control configuration Configurable controls, persisted to the browser	**15 pts** **Full Marks**	**0 pts** **No Marks**	15 pts
Technical - Sound Effects	**5 pts** **Full Marks**	**0 pts** **No Marks**	5 pts
Technical - High Scores persisted to browser	**5 pts** **Full Marks**	**0 pts** **No Marks**	5 pts
Technical - Animated Sprites Centipede, spider, flea, scorpion	**10 pts** **Full Marks**	**0 pts** **No Marks**	10 pts
Gameplay - Player control	**8 pts** **Full Marks**	**0 pts** **No Marks**	8 pts
Gameplay - Lives remaining	**5 pts** **Full Marks**	**0 pts** **No Marks**	5 pts
Gameplay - Current score	**3 pts** **Full Marks**	**0 pts** **No Marks**	3 pts
Gameplay - End of game message	**5 pts** **Full Marks**	**0 pts** **No Marks**	5 pts

Criteria	Ratings		Pts
Gameplay - Centipede Movement, splits correctly, etc.	**8 pts** **Full Marks**	**0 pts** **No Marks**	8 pts
Gameplay - Mushrooms Initial placement, damage states, etc.	**5 pts** **Full Marks**	**0 pts** **No Marks**	5 pts
Gameplay - Flea Initial placement	**5 pts** **Full Marks**	**0 pts** **No Marks**	5 pts
Gameplay - Spider	**5 pts** **Full Marks**	**0 pts** **No Marks**	5 pts
Gameplay - Scorpion	**5 pts** **Full Marks**	**0 pts** **No Marks**	5 pts
Gameplay - Poisoned Mushrooms Along with how they affect the centipede	**5 pts** **Full Marks**	**0 pts** **No Marks**	5 pts
			Total Points: 99