Josh Malmquist

1. We didn’t implement as many powerups as we first had imagined, and we chose to implement a leaderboard system for the current instance of the game to fulfill the key daemon requirement. We had ideas for more rock types, such as rocks that split in two, but also didn’t use those.
2. Media Requirements:

* Background music and sound effects
* Consistent style kept by using in-house images
* We used a font called Permanent Marker as our embedded font
* Multiple images are drawn on the canvas
* Spritesheet animation was used to animate the Z’s coming off the sloth
* A particle system is used when the rocks are destroyed
* Paths, arcs, and gradients are used to draw parts of the slingshot and background, as well as several main menu buttons

Interaction Requirements:

* The user can control the slingshot with the mouse and keyboard controls are used for leaderboard input. Players must hold shift and enter simultaneously to confirm the high score submission

Usability Requirements:

* The game pauses and unpauses when the window is not in focus
* Instructions are provided to the player to help them understand the game
* A score and life system are in place for the user to have an idea of how well they are doing
* As the game progresses, powerups are introduced. This is offset by the fact that more rocks spawn when a round (not shown to the player) is cleared, and higher rounds have a chance to spawn faster rocks in addition to the normal rocks
* We have 6 screens: main menu, instructions, credits, game, pause, and game over
* We put our names on this, because we have no shame

Game Design Requirements:

* We made something resembling our original game idea
* The game is playable by one person at a time, there is a lose condition in place, but the game is designed to run infinitely. The user can potentially improve by playing the game a lot. The game is engaging to those who play it

Coding Requirements:

* We used victor.js, and made our own versions of emitter.js and sounds.js, which were provided in class.
* Images are preloaded, object literals are used, the module pattern is followed, and two function constructors were used, one for particles and one for players in the high score system
* General code conventions were followed

1. Everything went pretty smoothly, the only notable thing that went wrong at one point was an overwrite of ctx.stroke as a string.
2. CITATIONS:
   * We used “Permeant Marker” font on google fonts
   * Background music - Artist: Lobo Loco, Song: Twisting Kangurus Album: Candy Nights, Source: https://freemusicarchive.org/music/download/cbe4393d2204fcfdf239f40e526707dd55547407
   * Slingshot fire - Artist: renne100 song:Slingshot Source:https://www.freesound.org/people/renne100/sounds/353033/
   * rockBoom - Artist: greenvwbeetle song: Boom 2 source: <https://www.freesound.org/people/greenvwbeetle/sounds/343452/>
3. Contributions:
   * KEVIN:
     1. Slingshot
     2. Particle effect
     3. Pause menu
     4. Css
     5. Image swap
     6. Rescaling for different resolutions
     7. Balancing/difficulty
     8. rocktypes
   * Josh
     1. Game state setup
     2. Rocks falling
     3. Collisions
     4. Audio
     5. made sloth and rock images
     6. leaderboard
     7. spritesheet animation
     8. menu hook ups
   * Both:
     1. Powerups
     2. Live system
4. Grade
   * KEVIN: 100%
   * Josh: 90%