Project Proposal

Problem I'm trying to solve:

1. I am creating a game with three main modes and multiple levels. User gets to design their own sand dunes, given some quantity of sand to place on the ground. The falling sand forms sand dunes, on which they will later drive their (different types of cars) over. One mode includes the help screen, instructions to play the game, and the choice of different types of cars. The second mode is the sand mode, where the user will click on some point of the screen, and from there sand will drop to the ground, and pile up at the bottom of the screen. The third mode involves driving the car (with different ability to climb inclines, speed and etc.) across the screen, in a side scroller-like fashion. Another feature is to have a drunk man walking on the sand dunes, who is unable to balance himself, and the user has to use left and right arrows the balance forward and backwards. (The center of gravity will be place on the head, so he may fall forward or backwards; game terminates when the man falls.) Same with the car mode; if the car runs out of fuel to climb hills or encounter a climb that cannot be overcome, the game terminates.
2. Audio and more complicated images will be used to perfect the game at a later stage.
3. I will be using NumPy and Pygame to make my game. To minimize the number of items stored in the sand mode, I stored the max points at each pixel within the object of Wall, and kept a dictionary of Walls as I keep updating it. As I progress, I may also use PyMunk to create the physics model for the car.

Competitive Analysis

- The idea of piling sand is inspired by **ThisIsSand** website, which simulates the falling of sand onto the screen to make landscape. I am trying to create the same mechanism of modeling the falling of sand, using the same mechanism as explained in my storyboard. (Triangular shape is overall more stable than other shapes, so I randomly select one direction to shift the sand.)
- However, one thing I don't really like about ThisIsSand is that it does not really model what sand looks like in nature under other forces. When factors such as wind are taken into account, the top of the sand dunes are often in the shape of an almost-triangle, except the top is smooth. So, to approximate the shape of sand dunes, I used NumPy to approximate using a sixth degree polynomial to find a continuous and smooth curve that describes the shape of the sand dunes.
- In addition to using high degree polynomials, I am also planning on adding the "wind" feature at the end to flatten out any cusps at the end of dumping sand.

- The car driving part is inspired by **Shark Bike**, an online game that allows the user to control the motion of the bike's movement on slopes. I like the physics involved in the motion of the Shark Bike game, but necessarily the same physics can be applied to a car, because the weight is more evenly distributed throughout the moving body and the center of gravity is closer to the ground, whereas the Shark is more heavy on top and less stable. However, I think I will use the same techniques to display my game.
- To generate a relatively smooth curve, I think I will use NumPy to generate best fit line, using the current position, displayed at the center of the screen, as the midpoint of evaluating best fit line, and use the left and right bound as the bounds.
- Similar to the Shark Bike game, I will be using up, down, left and right to control the movement of the vehicle. However, there will be multiple different models users can choose from, and the amount of fuel each has and each vehicle's ability to climb hills will be different. Basically, if you choose a faster car, your ability to climb high hills and the fuel (basically moving time) will be compromised, and vice versa.
- The end goal is to get to the end of the course. If the fuel runs out, then game ends. So basically if you're stuck at a spot or at a climb that you can't run over, your fuel will keep decreasing until you're completely out.
- There is also a mode that features a drunken man who is trying to walk on desert, inspired by game Die Wagenschenke. Similar to Die Wagenschenke, the person walking will wobble (randomly, left or right), and the user must use left or right key to prevent him from falling down. He will be automatically walking forward, in contrary to the driving mode. The end goal for this mode is for the drunken man to make it across the screen without falling.
- In my game, this feature is upgraded, because the character must balance forward and backwards, at the mean time, the sand is moving underneath due to the force exerted. Also the drunk guy is climbing a slope, which would require additional physics.
- Different from any of these games aforementioned, for any object moving on sand, the sand will moving accordingly and sand dunes will change shape as one steps on the sand piles.

1. https://thisissand.com (Sand piling / landscape making game)
2. http://www.duckgame.net/play_game/shark_bike.html (Shark bike)
3. http://www.wagenschenke.ch/index_site.htm (Die Wagenschenke)

Storyboard

Storyboard is included in the zip file.

Code Artifacts
Code artifacts are included in the zip file.

Timesheet
Inclucded in the zip file.