The goals that we identified for the midpoint check in are having the general classes for the hero and monster along with basic capabilities. We have accomplished these goals, but have also realized the loftiness of many of our stretch goals and are adjusting our planning accordingly.

We based much of our design on the “Brick Breaker” framework that Ben used as an example in class. The classes that we have created are: brick, hero, and arrow. The code for our classes can be reviewed here: <https://github.com/sd19spring/InteractiveVisualization-Mellie_Anthony/blob/master/model.py>

Brick will be used to track the health of our characters and in it has a height, width, x position, and y position. We need to add a capability for changing the level of health to the brick so that the health of our hero and monster can be visualized.

Hero will be used as the characters in our game and has the attributes name, health, height, width, x position, y position, and velocity. Hero and monster are types of the hero class. When a hero collides with a projectile (the arrow class), they take damage based upon the lower\_health function. Monsters can also be befriended by getting hit by cookies and therefore have a raise\_health function. Monsters can be defeated by decreasing their health with arrows or increasing their health (“befriending” them) to a certain level with cookies. Monsters move back and forth across the screen according to a constant velocity. Heroes move based upon mouse position or keyboard buttons, we haven’t decided. We definitely want to include images to take the place of our heroes, but our stretch goal of having the characters’ legs move as they move is definitely a long shot.

Arrow will be used as a projectile that shoots from a character towards the opposing character and has the attributes damage, height, width, x position, and y position. Arrows, fireballs, and cookies are types of the arrow class. Heroes will shoot arrows to do damage to monsters and cookies to befriend monsters. Filling a monsters “befriend” brick by hitting the monster with enough cookies is one way to win the game. The other way is to kill the monster with arrows. Fireballs are shot by the monster and damage the hero.

We have fulfilled our basic capabilities midpoint goal by having the monster move back and forth at a constant velocity and having the hero track our mouse. The basic framework of establishing a window with our objects inside of it is done and the code for it can be found here: <https://github.com/sd19spring/InteractiveVisualization-Mellie_Anthony/blob/master/view.py>

The next big step is figuring out projectiles that originate at the position of the hero and move at a constant velocity across the screen vertically. We then have to figure out how to make health adjust based upon the collision of a hero and an arrow. The final step that we definitely want to accomplish is making our objects into images instead of basic shapes. Both Anthony and Mellie will work on all of these steps together.

Some stretch goals that are still within our reach are establishing some sounds during the game play such as when an arrow is shot or when a collision occurs as well as a start screen and a screen that pops up if you defeated the monster. Some stretch goals that might be out of the picture now are having multiple rounds with different health for our heroes or damage for our projectiles.