**STATUS REPORT #4**

**Scott & Nicole**

List of scratches

AH Bitmap font: Created text for the gameover screen. The text is also used in rules screen and has made it much easier to show something on screen without it being in image. It also allows us to quickly change the text displayed, as we are still working on the different features of the game.

AH Fixed animation of hamster: We started by trying to change the image all together but we found problems in that the animation was having troubles with a non square image, since I had cut out the white space. Nicole then went back in and got a bounding rectangle from the sprite. From that rectangle, a new rectangle is created, with the Y and height cut off. Then this rectangle is used in the isHit function.

AH Pellets: Created a function for pellet to make it easier for it to be on any screen needed and easier for Pellet Maker (randomly generated food)

AH Poison: Created a function for poison for the exact same reason I made a function for pellet

AH Rules screen: We had an unused screen and as a result I filled it with an incomplete thing for now… just makes things feel more complete. It also helps, as all of our features are now able to be viewed in one place/

AH Unique hit detection in scrGame: One of the main features of Level 2. If the two hamsters hit head on, the bigger hamster wins, but if they hit any other way, the faster one wins.

AH Choice of animal: Choose colour of hamster (only 2 so far because only need 2 right now)

AH Level 2: Made the 2nd level of the game. Different background, food type, with the addition of poison which slows down the player while leaving the players size alone, as well as unique hit detection and size and speed bars in the making.

AH Randomly generated food: Randomly places food around the screen and times it so that it will disappear if in the same spot for too long.

AH Size and speed bar in scrGame: Using pixmaps, draws two bars for each hamster, one for size and one for speed. More information on this scratch in the challenges below, as this scratch is still a work in progress.

Major challenges/setbacks

A challenge continues to be with GitHub. GitHub sometimes thinks that files have been changed, even though they were not altered at all. Abdullah helped us download GitKraken, which has helped a lot.

The randomly generated food scratch proved to be an interesting challenge. We have to functions, one called Pellet and the other called PelletMaker. The Pellet function loads the texture, the size, and the flip. The PelletMaker function draws the pellets randomly (using Math.random) and removes it. This means that everything that is happening across the different screens is centralized and allows us better channeling ability when changing what is on the screen. Glad with how it all worked out and now only dealing with interesting hit detection problems where if a pellet spawns in a wall and how the pellet is told to spawn somewhere else be at the same time adds another pellet into the mix… So if 2 originally were the only pellets in, if one hits a wall and it doubles, now all of a sudden there are 3, which shouldn’t happen but does.

The size and speed bar scratch has probably been the most challenging and time-consuming scratch these past weeks. The idea is that each player has two bars: one for speed, and one for size. When a hamster runs over a strawberry, both size and speed are increased. When he runs over a poison, only speed is decreased. In scrGame2, a pixmap is created. Then, the colour is set. Then pixmap.setRectangle is called, using a variable called nShapeW. 15 is either added or subtracted to nShapeW when either a strawberry is eaten or a poison is eaten. To be able to draw the bars, a texture is created from the pixmap, and then a sprite from the texture. However, there is a bug in this code, which we do not understand yet. The rectangles are increasing fine when a strawberry is eaten. However, when a poison is eaten, it does nothing, until the hamster runs over a strawberry again; then, 15 is just not added to that rectangle. Much thought and work has already gone into this scratch, but sadly it is not working well for this status report as of yet. The idea is also that we will be put labels on the bars, so that the players know which bars are for which player and which feature.

Source any web site/book that helped you with that concept

Information on a null pointer exception error related to Bitmap Font:

<https://stackoverflow.com/questions/43959998/libgdx-bitmapfont-nullpointerexception>

More information on Bitmap Font rotation:

<https://stackoverflow.com/questions/8508749/draw-a-bitmapfont-rotated-in-libgdx>

Explanation of the math.random feature, which was used in the randomly generated pellet scratch:

<https://stackoverflow.com/questions/7961788/math-random-explained>

Information on how to draw the bars in the size and speed bar scratch:

<https://stackoverflow.com/questions/15397074/libgdx-how-to-draw-filled-rectangle-in-the-right-place-in-scene2d>

General information:

ICS3UI website

Abdullah with the continued github and other stuff help

Dennis for being a great help with random questions that might come up

Lessons learned from the last two weeks:

1. Rectangles are hard and for some reason dislike changing one way from another. We will research more about this to see if there is a more efficient way to do this.
2. Creating randomly generated things is quite fun and can lead to quite a challenge of where they get placed.
3. Putting everything we have done into a new version when so much has been worked on is really cool to see where the project has gone.
4. We have gotten a lot of practice with functions, which had always been a more confusing topic.
5. Sitting by people or talking to people who worked / are working on something similar to what you are doing is great to generate ideas / figure out what you need to work on.
6. We have also gotten a lot of practice with static public variables, and when and where to use them correctly.