**15-112: Term Project**

HERD RUN by Angela Liu (amliu)

**Project purpose:** Create an engaging, fun yet strategic game for all

**Problem:** High demand of free games

People spend significant amounts of time online. The younger population utilizes the web as a means of finding entertaining distracters, while the older balance between work and relaxation. As such, from both types of users, there is an ever growing demand for free games. My game aims to fulfill that demand. It’s simple point mechanic and playful sounds will attract younger users. The transitional upgrades you can apply to your character, plus the attacks and how they may be combined for better use, add a strategic element to the game. This will help keep the interest of the older users.

**Method:**

I have used python 2.7.2 and Pygame to create the game. I split components of the code into three parts – herdGame, object\_creator, and util. Each contains different, integral parts of code that come together in herdGame. However, util and object\_creator also call each other’s functions as well.

herdGame.py is the main function of the game. It initializes values for each game and maintains them until game over, then resets them when new games are made. To initialize those values, it creates objects using classes in module object\_creator, and then modifies them as the game progresses. It mainly gets player moves and draws them. A few managing functions (i.e. moveAnimals) call functions found in module util that take care of things behind the scenes.

Object\_creator creates objects for values to be stored in. There are three types of moving objects: *player*, *animal*, and *monster* objects. These store images, current x/y positions (which are changed during the game), movement information, and more. Movement works by switching on and off move\_left (etc) values to true or false, allowing diagonal movement. The moving objects work similarly, although each has special values. Player objects must store status levels and more. In addition to those, the *powerups* class stores powerup images and values. *Score* stores time, score, transition, tip images, level, and other related information. Finally, *weapons* stores images related to attacks and the x/y coordinates of effect, as well as other values that influence how each attack distinctly works.

util contains all the functions that modify, reset, check, and sense object values. Classes each have general names which relate to what types of functions they contain. *Tossing* contains functions involved in picking up/letting go of animals, powerups, and more. *Moving* contains functions involved in moving the player/animal/monster, as well as detecting collisions. Moving is by far the largest class as it also contains the animal AI for wandering/running away from monster once it is close, and the monster AI for targeting close animals/decoys. It also prevents walking into UI areas as functions sense collisions between objects*. Managing* is mostly concerned in spawning animals and boosting difficulty for higher levels. *Using* draws attacks and sense the boundaries of activity created when successfully used. *Sprite* manages current image switching to allow simple animation on top of moving images. Finally, *points* keeps track of upgrades, point values, and more. Many of the functions call other functions located in different classes in util to prevent crashing and other obvious reasons.

**User Interface:**

I integrated what I learned in other classes to carefully place the most important bits of information where the user can easily locate them. Things in the top left corner are processed first, as per habit. Since time is the only factor that can cause a game over and the monster, if it eats cows, depletes time, I decided to put it at top where users (even if they are not focused on it) can see at all times. If that is not enough, I made it to change colors from green (healthy amount of time) to red (bad!) as time gets closer to reaching 0.

The user’s score and next score goal is also integral in making decisions – should I risk going for an animal, or freeze the monster now? That is placed in the top left, a space that people are naturally inclined to look towards due to convention.

Other components of UI are all in black, as to easily separate it from the game board. Less important information are displayed lower on the right and do not need to be looked at as much.

**Other games:**

The main game I took inspiration from is called “Extreme Herder.” At its core, Herd Run is the same, as both involve picking up animals and dropping them off to safe zones while preventing them from being eaten by some monster force. However, I changed the design significantly. In “Extreme Herder,” the pen is located in the middle and game over occurs when the monster has eaten three animals (or you). I felt that the difficulty of the game spiked tremendously as life never heals and the monster gets faster, so I changed mine to be time based. I also added a layer of depth to my game by allowing attacks and stored usage of powerups. In “Extreme Herder,” there really is only one way to play. But mine allows customization and testing. Perhaps upgrading just the zap attack isn’t the way to go to achieve the highest score? Next time, a user may try a different strategy. Using the food attack in conjunction with the fire wall or zap attack is also an unintended combination of effects that turn out to be more effective than using attacks individually. The feel of the game is much more active and hectic than “Extreme Herder.” Herd Run adds a layer of complexity that I believe improves on the game I based it off.

Being able to upgrade skills is a feature present in many games, though I got the idea to incorporate it after playing “Zombie Hero.”

**User Studies:**

4/25/2013 – Two other users played a version of Herd Run. Their concerns involved confusion regarding controls and objectives. In response, I have added a guiding arrow to lead users towards the pen when carrying animal and changing tips that act as a sort of tutorial.

I also changed having to manually drop animals with “A” in the pen in response to its unintuitive nature. Now, animals automatically drop when you touch the pen.

**Future Improvements:**

Features that were planned to be implemented in the future:

* In the bottom left corner, it displays information about the animal you are carrying. I would like multiple kinds of animals, each with different effects (i.e. +2 speed, freeze monster, -2 speed) when held and with different point values. This would add another layer of strategy. A chicken may be worth 20 points. Is it worth going after if it decreases your time?
* More types of powerups:
  + Refill stamina
  + Freeze time
  + Magnet -> animals go towards you for X seconds
* Upgrading attacks also boost their power
  + Fire Wall -> stronger slowing effect, larger radius
  + Zap -> longer freeze effect
* Multiple monsters in later levels that corner animals/work together
* Bonus bar -> If user scores animals in rapid succession, score bonus points and time
* High Score
* Multiplayer -> Person controls herder, other controls monster
* Interactive tutorial involving fulfilling goals based on user interaction
* Customizing controls
* Online play