Project Proposal Report

Team Infinity

Environmental Pet Simulator

Cara Fisher c086f573@ku.edu

Paul Dreger pdreger@ku.edu

Hans Brown h458b873@ku.edu - ***Contact***

Jackson Schilmoeller j427s666@ku.edu

Daniel Murga d060m098@ku.edu

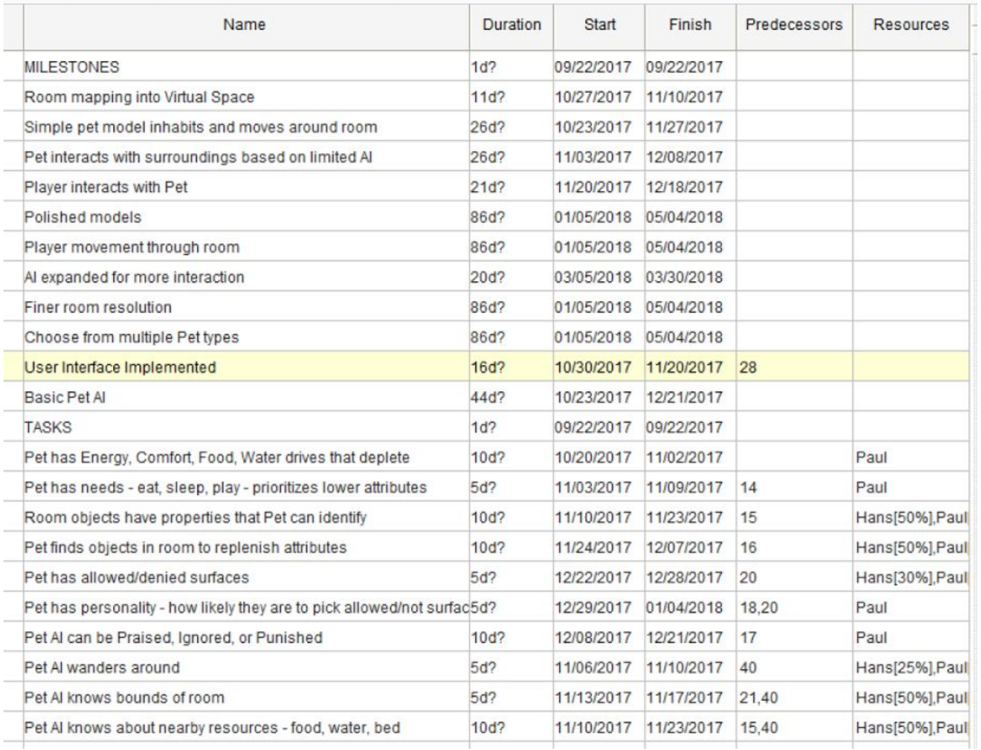
Project Description

- Why is the project being undertaken?
    - We wanted to do a VR project. Some of us wanted to do virtual environments, and others wanted to make a game. Given the scope of recreating the real world in virtual space, we decided on a pet game that would have the possibility of real-virtual world interactions.
- Describe an opportunity or problem that the project is to address:
    - There has been substantial research showing that owning a pet improves health (see works cited). If one can not have a pet due to limiting physical capability, allergies, housing contracts, or any other reasons, then a real life pet is infeasible, and those health benefits are lost. Therefore, we decided to make a VR pet simulator to allow for those affected to hopefully gain some of the health benefits. By reading in the user's environment, and translating the physical environment to an (ideally) one-to-one virtual environment, the feeling of disconnect between the physical world and virtual world will be lessened.
- What will be the end result of the project?
    - An interactive virtual pet living inside a virtual user-customizable model of the the user's room.

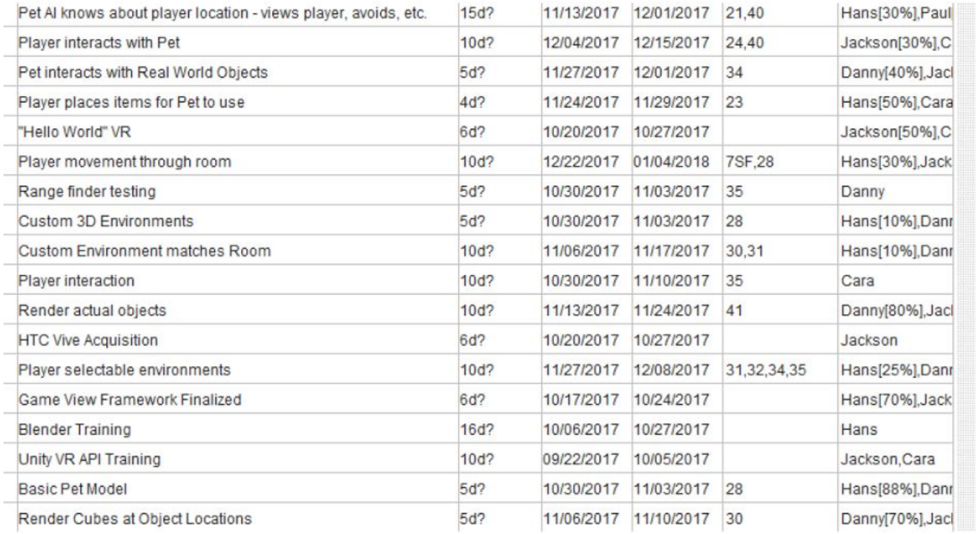
Project Milestones

- First semester
    - Room is mapped into virtual space
        - October 27, 2017
    - Simple pet model inhabits and moves around room
        - November 10, 2017
    - Pet interacts with surroundings based on some limited AI (neural network or fuzzy logic finite state machines)
        - December 8th, 2017
    - Player interaction with pet
        - December 8th, 2017
- Second semester
    - Polished models
        - May 4th, 2018
    - Player movement through room
        - May 4th, 2018
    - AI is expanded for more interaction
        - March 26th, 2018
    - Finer room resolution
        - May 4th, 2018
    - More choices of pets
        - May 4th, 2018

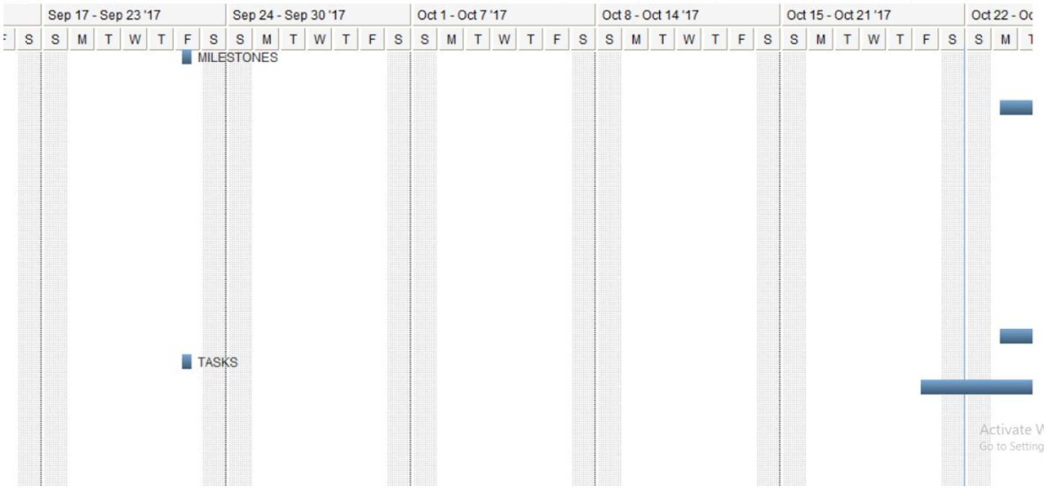
- Gantt Chart:

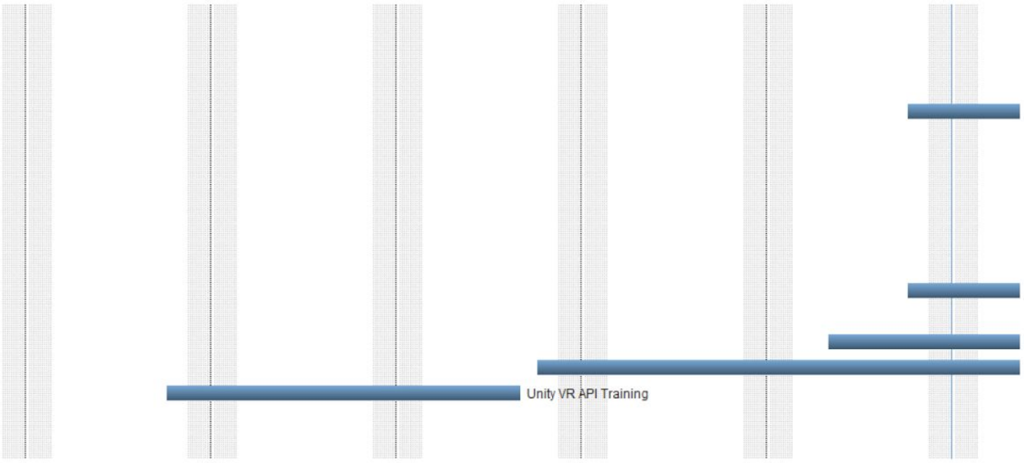


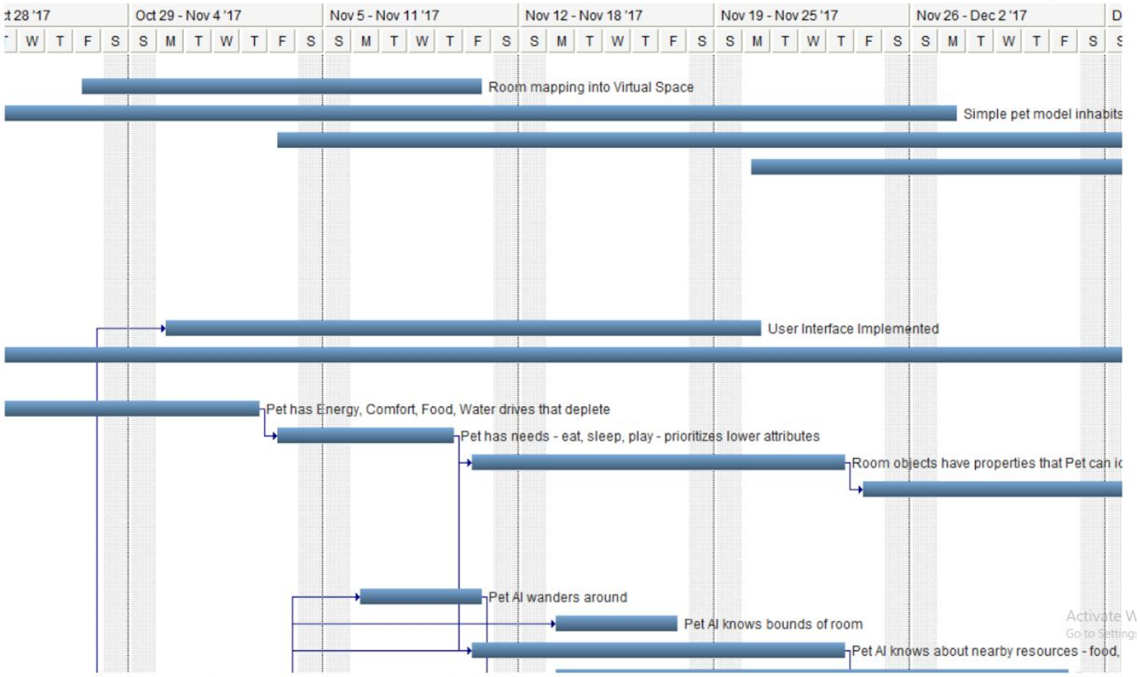
○

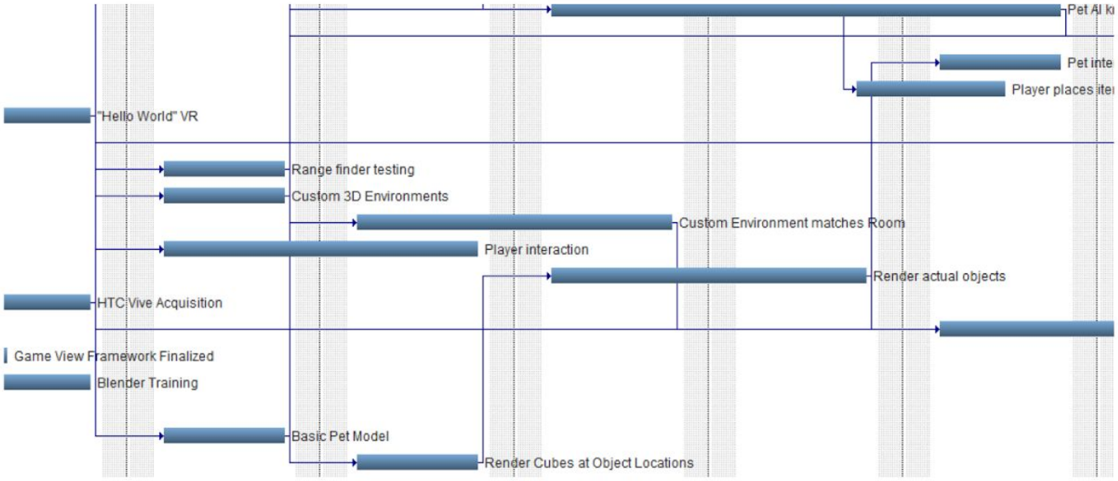


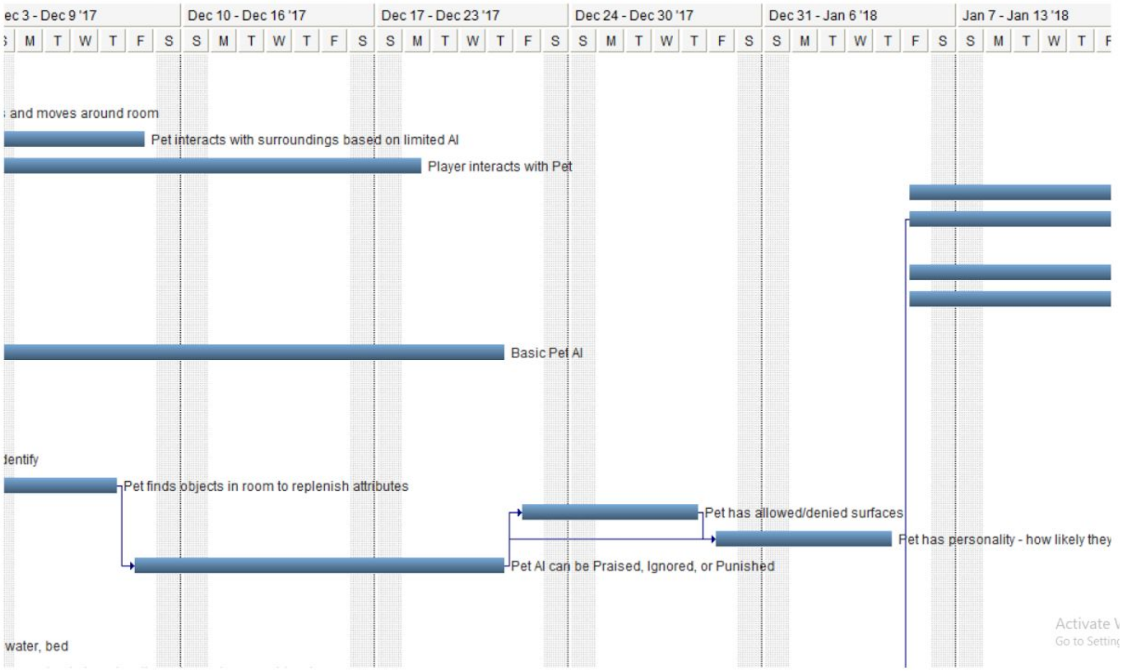
Name	Duration	Start	Finish	Predecessors	Resources
MILESTONES	1d?	09/22/2017	09/22/2017		
Room mapping into Virtual Space	11d?	10/27/2017	11/10/2017		
Simple pet model inhabits and moves around room	26d?	10/23/2017	11/27/2017		
Pet interacts with surroundings based on limited AI	26d?	11/03/2017	12/08/2017		
Player interacts with Pet	21d?	11/20/2017	12/18/2017		
Polished models	86d?	01/05/2018	05/04/2018		
Player movement through room	86d?	01/05/2018	05/04/2018		
AI expanded for more interaction	20d?	03/05/2018	03/30/2018		
Finer room resolution	86d?	01/05/2018	05/04/2018		
Choose from multiple Pet types	86d?	01/05/2018	05/04/2018		
User Interface Implemented	16d?	10/30/2017	11/20/2017	28	
Basic Pet AI	44d?	10/23/2017	12/21/2017		
TASKS	1d?	09/22/2017	09/22/2017		
Pet has Energy, Comfort, Food, Water drives that deplete	10d?	10/20/2017	11/02/2017		Paul
Pet has needs - eat, sleep, play - prioritizes lower attributes	5d?	11/03/2017	11/09/2017	14	Paul
Room objects have properties that Pet can identify	10d?	11/10/2017	11/23/2017	15	Hans[50%],Paul
Pet finds objects in room to replenish attributes	10d?	11/24/2017	12/07/2017	16	Hans[50%],Paul
Pet has allowed/denied surfaces	5d?	12/22/2017	12/28/2017	20	Hans[30%],Paul
Pet has personality - how likely they are to pick allowed/not surfac	5d?	12/29/2017	01/04/2018	18,20	Paul
Pet AI can be Praised, Ignored, or Punished	10d?	12/08/2017	12/21/2017	17	Paul
Pet AI wanders around	5d?	11/06/2017	11/10/2017	40	Hans[25%],Paul
Pet AI knows bounds of room	5d?	11/13/2017	11/17/2017	21,40	Hans[50%],Paul
Pet AI knows about nearby resources - food, water, bed	10d?	11/10/2017	11/23/2017	15,40	Hans[50%],Paul
Pet AI knows about player location - views player, avoids, etc.	15d?	11/13/2017	12/01/2017	21,40	Hans[30%],Paul
Player interacts with Pet	10d?	12/04/2017	12/15/2017	24,40	Jackson[30%],C
Pet interacts with Real World Objects	5d?	11/27/2017	12/01/2017	34	Danny[40%],Jack
Player places items for Pet to use	4d?	11/24/2017	11/29/2017	23	Hans[50%],Cara
"Hello World" VR	6d?	10/20/2017	10/27/2017		Jackson[50%],C
Player movement through room	10d?	12/22/2017	01/04/2018	7SF,28	Hans[30%],Jack
Range finder testing	5d?	10/30/2017	11/03/2017	35	Danny
Custom 3D Environments	5d?	10/30/2017	11/03/2017	28	Hans[10%],Dann
Custom Environment matches Room	10d?	11/06/2017	11/17/2017	30,31	Hans[10%],Dann
Player interaction	10d?	10/30/2017	11/10/2017	35	Cara
Render actual objects	10d?	11/13/2017	11/24/2017	41	Danny[80%],Jack
HTC Vive Acquisition	6d?	10/20/2017	10/27/2017		Jackson
Player selectable environments	10d?	11/27/2017	12/08/2017	31,32,34,35	Hans[25%],Dann
Game View Framework Finalized	6d?	10/17/2017	10/24/2017		Hans[70%],Jack
Blender Training	16d?	10/06/2017	10/27/2017		Hans
Unity VR API Training	10d?	09/22/2017	10/05/2017		Jackson,Cara
Basic Pet Model	5d?	10/30/2017	11/03/2017	28	Hans[88%],Dann
Render Cubes at Object Locations	5d?	11/06/2017	11/10/2017	30	Danny[70%],Jack

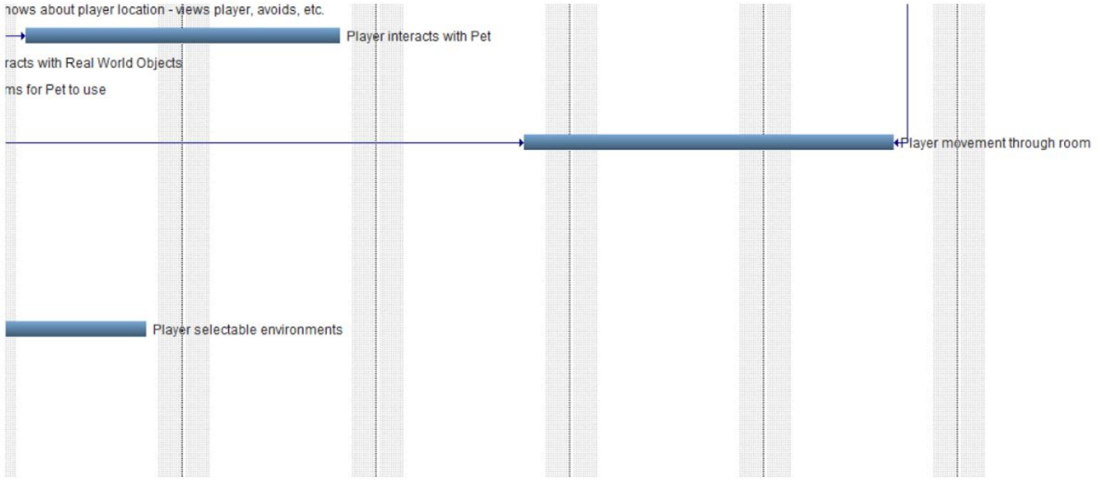

Sep 17 - Sep 23 '17
Sep 24 - Sep 30 '17
Oct 1 - Oct 7 '17
Oct 8 - Oct 14 '17
Oct 15 - Oct 21 '17
Oct 22 - Oc
MILESTONES
TASKS

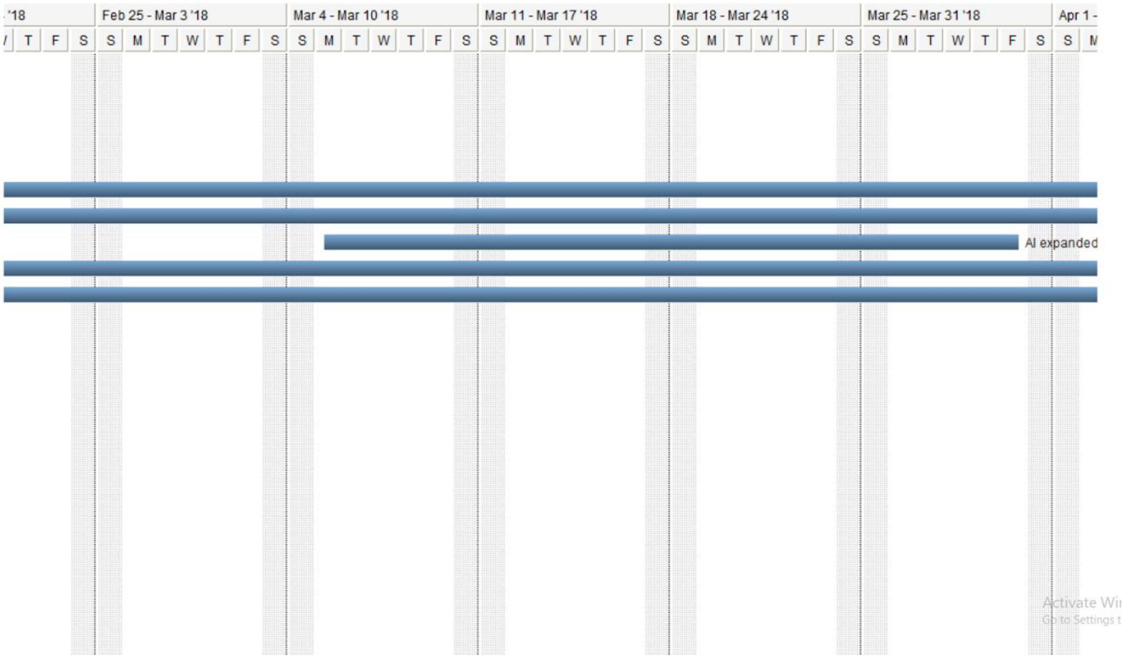

Unity VR API Training

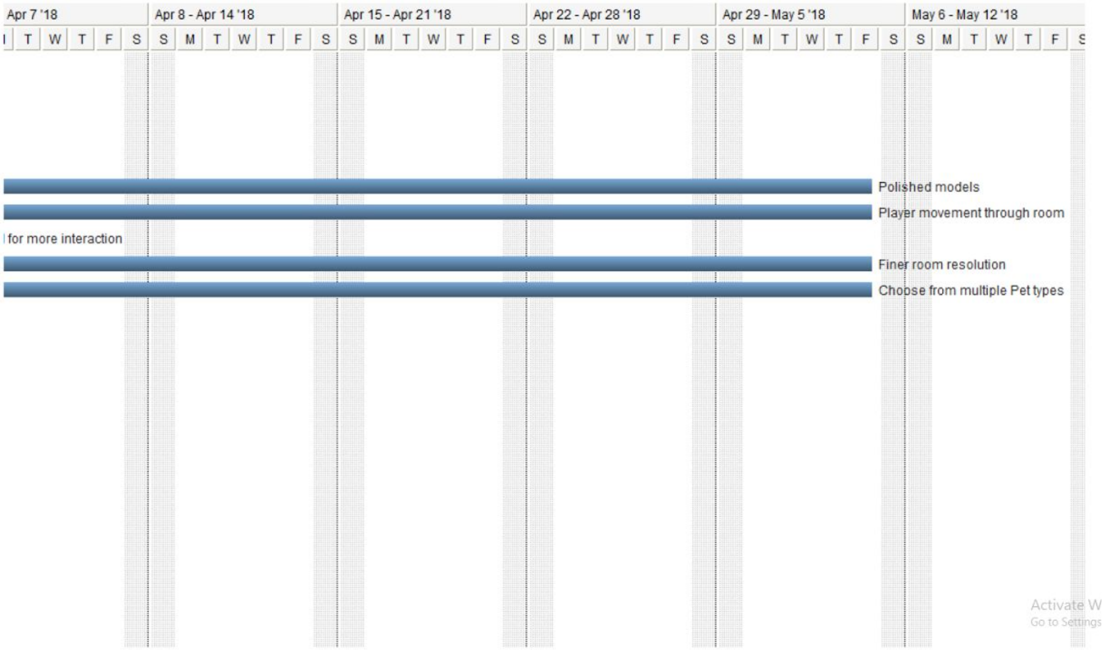

t 28 '17
Oct 29 - Nov 4 '17
Nov 5 - Nov 11 '17
Nov 12 - Nov 18 '17
Nov 19 - Nov 25 '17
Nov 26 - Dec 2 '17
D
Room mapping into Virtual Space
Simple pet model inhabits
User Interface Implemented
Pet has Energy, Comfort, Food, Water drives that deplete
Pet has needs - eat, sleep, play - prioritizes lower attributes
Room objects have properties that Pet can ic
Pet AI wanders around
Pet AI knows bounds of room
Pet AI knows about nearby resources - food,


Pet AI kı
Pet inte
Player places ite
"Hello World" VR
Range finder testing
Custom 3D Environments
Custom Environment matches Room
Player interaction
Render actual objects
HTC Vive Acquisition
Game View Framework Finalized
Blender Training
Basic Pet Model
Render Cubes at Object Locations


ec 3 - Dec 9 '17
Dec 10 - Dec 16 '17
Dec 17 - Dec 23 '17
Dec 24 - Dec 30 '17
Dec 31 - Jan 6 '18
Jan 7 - Jan 13 '18
S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F
and moves around room
Pet interacts with surroundings based on limited AI
Player interacts with Pet
Basic Pet AI
dentify
Pet finds objects in room to replenish attributes
Pet has allowed/denied surfaces
Pet has personality - how likely they
Pet AI can be Praised, Ignored, or Punished
water, bed
Activate \
Go to Settin


nows about player location - views player, avoids, etc.
Player interacts with Pet
racts with Real World Objects
ms for Pet to use
Player movement through room
Player selectable environments





- For PDF document, download using link: https://docs.google.com/document/d/118HIyEobgK--d8o48tiRq4er8ZdlbHpPOKRAksOfQVw/edit?usp=sharing
- To view original Gantt chart in Gantter, download using link: https://drive.google.com/file/d/0Bz17CpbcFn_bcFBobkFDQW5FQWc/view?usp=sharing, then open using the Gantter website (requires free account) or the Gantter for Google Chrome App.

Project Budget

- Hardware, software, and/or computing resources
    - Unity Game Engine Personal Edition (free)
    - HTC Vive (provided through the school)
- Estimated cost
    - Unity Game Engine Personal Edition: Free, project is non-profit
    - HTC Vive: $599
- Vendor
    - Steam for HTC Vive
- Special training (e.g., VR)
    - Unity Game Engine (free)
    - Unity VR API
        - https://unity3d.com/learn/tutorials/topics/virtual-reality/vr-overview?playlist=22946
- When they will be required?
    - VR Headgear is needed as soon as we can begin testing.
        - Testing begins after we finish researching VR implementation in the Unity Engine

Work Plan

- Paul will handle implementation of Pet Logic, the Pet-Player interaction system, and Pet Artificial Intelligence
- Cara will handle VR implementation and the Game Model
- Jackson will handle VR implementation, VR->Real World mapping
- Hans will handle layout design, graphics, and the Game Model
- Danny will handle Sensor Reading and VR->Real World mapping

Github/Unity links

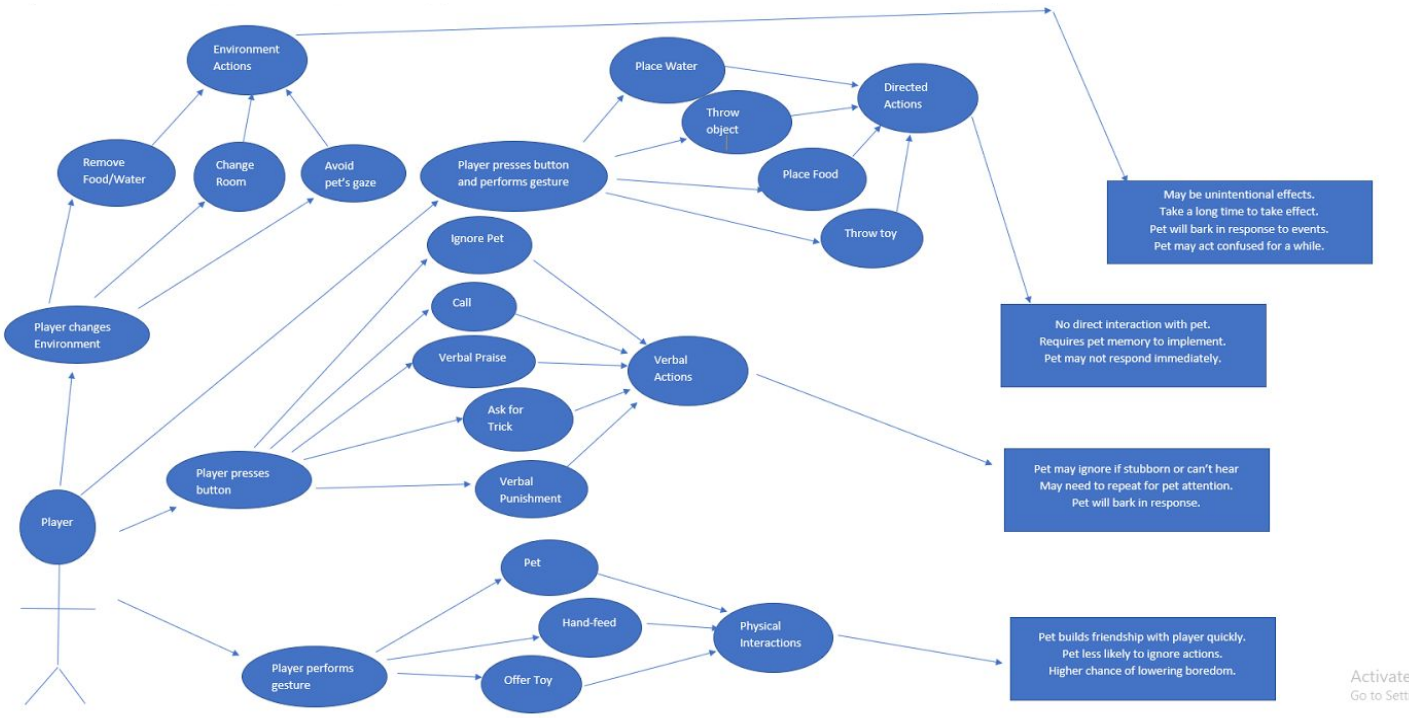
- https://github.com/Hansigzandrinovka/581-Ecosystem-Pet-Simulator
- Unity Cloud Project Name: EcosystemPetSimulator
- Unity Cloud Project ID: 4a251b7e-c15d-4ac6-9951-cdfa5f406abb

Preliminary Project Design

Our project will consist of four components. Pet Design will govern the modeling, AI logic, and general actions performed by the pet. VR implementation will govern the Player's view of, and interactions with, the virtual world through the HTC Vive headset and its peripherals. The Game Model abstracts away the details of the game engine and provides an

API that the player, pet, and other high-level components can use to interact. Finally, the Environment Reader will handle reading in the environment around the Vive headset and translating that into game coordinates.

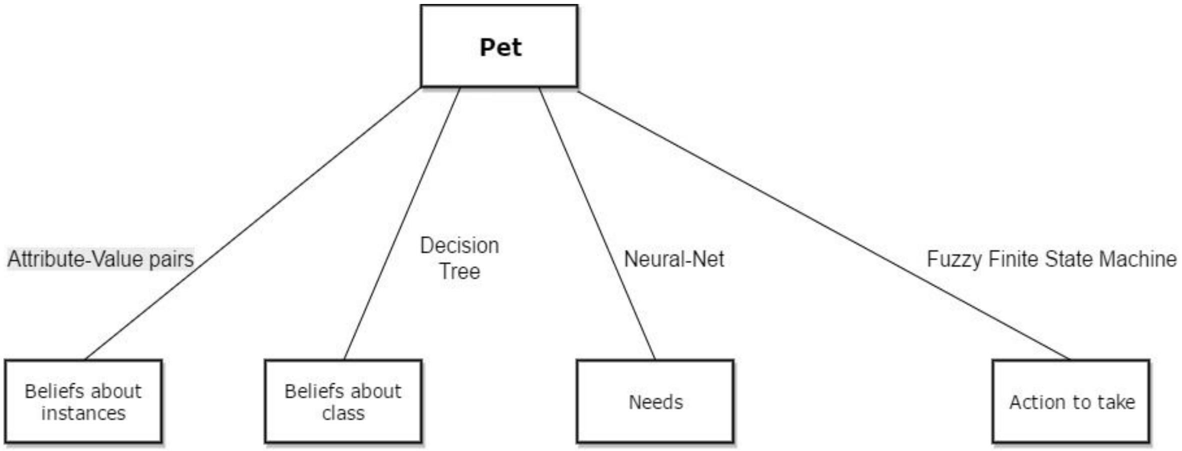
Pet Design


Environment Actions
Place Water
Directed Actions
Throw object
Remove Food/Water
Change Room
Avoid pet's gaze
Player presses button and performs gesture
Place Food
Throw toy
May be unintentional effects.
Take a long time to take effect.
Pet will bark in response to events.
Pet may act confused for a while.
Ignore Pet
Call
Player changes Environment
Verbal Praise
Verbal Actions
No direct interaction with pet.
Requires pet memory to implement.
Pet may not respond immediately.
Ask for Trick
Player presses button
Verbal Punishment
Pet may ignore if stubborn or can't hear
May need to repeat for pet attention.
Pet will bark in response.
Player
Pet
Hand-feed
Physical Interactions
Player performs gesture
Offer Toy
Pet builds friendship with player quickly.
Pet less likely to ignore actions.
Higher chance of lowering boredom.
Activate
Go to Sett

Player-to-Pet Use Case Diagram:
Download from
https://drive.google.com/file/d/0Bz17CpbcFn_bZ3RrR1INSmxzNmM/view?usp=sharing
for a better view.

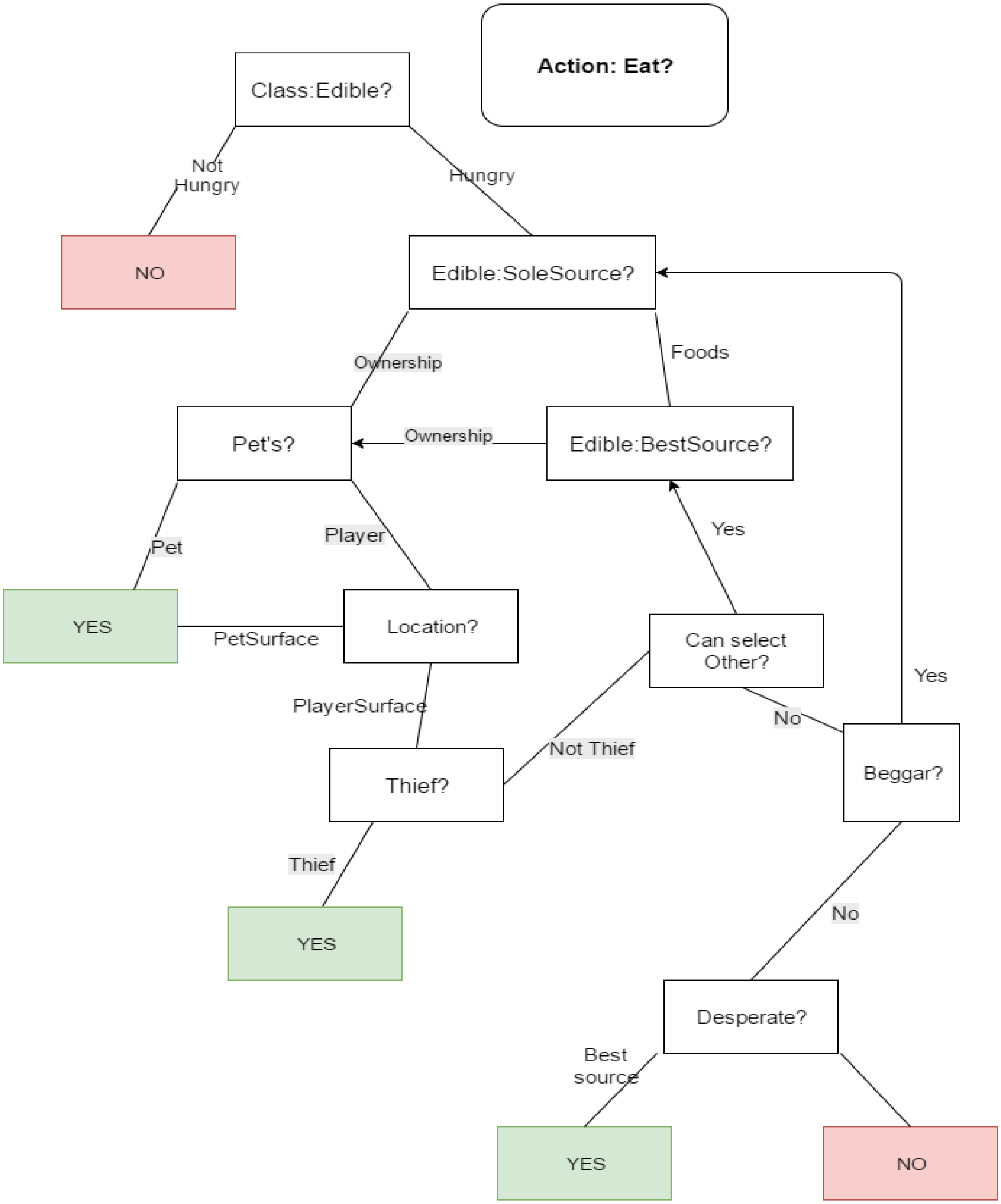
- Pet needs
  - Food - The pet will get hungry over time, more so when the pet is exerting itself, IE playing, running around. When the pet becomes very hungry, it will seek out any nearby food sources, or beg the player to provide one. After the player provides a food source, the pet will try to eat from it or ask for food. Once the pet has brought its hunger-meter to near-empty, it will stop eating and re-evaluate its needs.
  - Water - The pet will get thirsty over time, more so when the pet is exerting itself. After the player provides a water source, the pet will seek it out. If there is no water inside, the pet will beg the player to fill it up. Then the pet will re-evaluate its needs.
  - Entertainment - The pet will get bored over time if not properly stimulated. The player will have a variety of toys they can use to play with the pet to keep it entertained. Unlike other needs, the pet will be almost always open to playing

with the player if the player equips a toy. As boredom progresses in stages, the pet may ask the player to play with it, or find some other way to entertain itself.
    - Energy - Over time the pet will lose energy (at a variable rate depending on the action taken). Once it is depleted enough, the pet will look for surfaces/objects that are most beneficial to regain its energy tempered with the learned responses. For example, if the pet learns through feedback that a couch is not for sleeping, the pet should choose a safer (if lower value) object to sleep on to prevent the user's displeasure.
- Pet AI

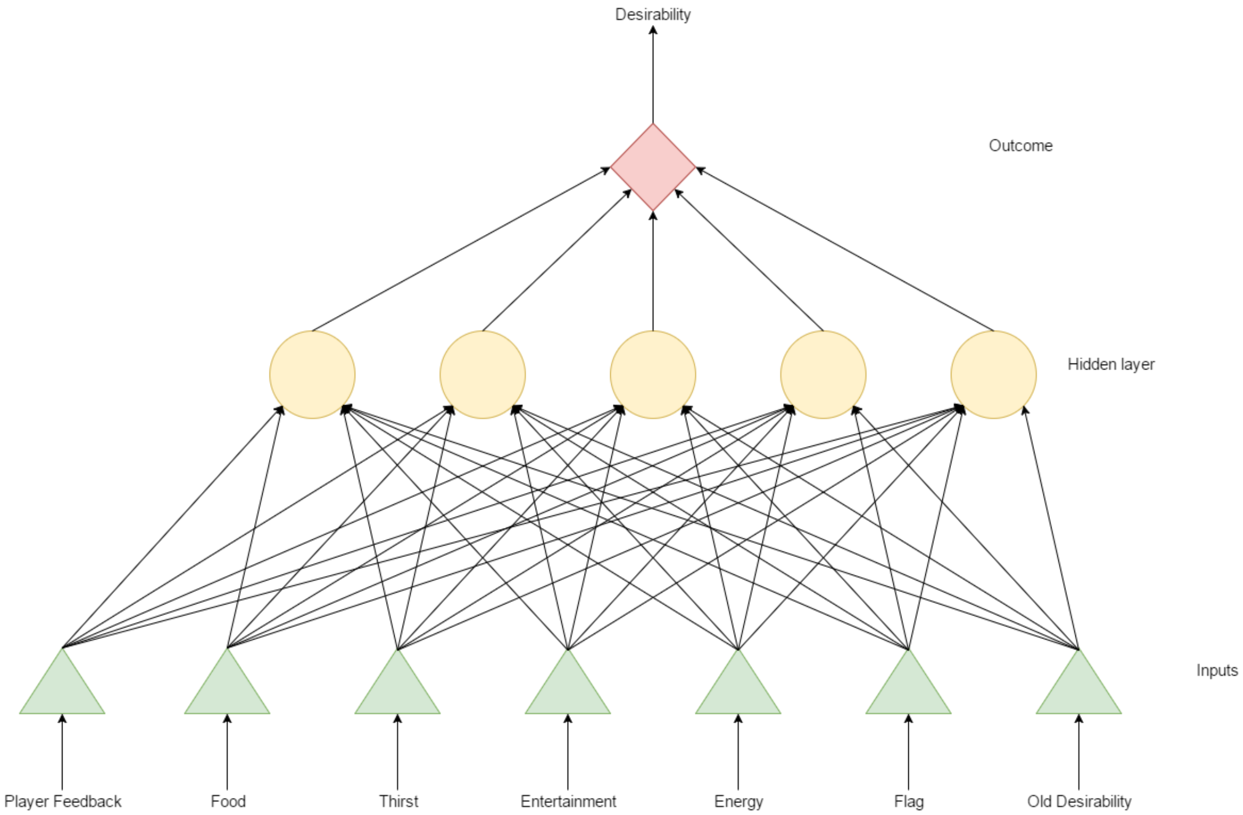


    - Here is the high-level view of the Pet AI. It operates by assigning values and flags to an instance of an object in the Attribute-Value stage. Next, a decision tree is built out of the related instances' feedback, so that the pet can decide what to do for that class of object. The neural network adjusts the weights of the attribute-values from player feedback (Praise, Punish, Ignore), making it so the the decision tree changes overtime. The pet decides the most pressing need it needs to fulfill, and selects the action to best fulfill it.
        - Attribute-Value Pairs: How well a specific instance of an object fulfills the driving need, as well as setting flags for the pet for clarification. For example, while PlayerFood may fulfill Hunger by 8 points, if the pet has learned that instances of PlayerFood NOT on the floor are for the player, the Desirability rating of PlayerFood without the tag of being on a PetSurface would be less than an allowed food instance (on the floor, PetFood, etc). The values change by the user Punishing/Praising/Ignoring the pet by use of the neural network.
        - Decision Tree: Uses the Attribute-Value Pair instances to generalize to make a decision about an action. For example, in the following figure, the pet currently has some sort of instance, and wants to decide whether or not it should eat it. First, the pet (referring back to the attribute pairs) queries if the instance is edible. If it is not edible, and the pet is not hungry, it does not eat the instance. Next the pet knows that the instance is edible, and the pet is hungry. It now looks around for options (Edible:SoleSource?). If it is the only source of food, the pet queries if the instance is flagged for the pet or for the

player. If it is the pet's, it is eaten. If it is the player, it checks to see where the instance is located. If it is on a PetSurface, the pet eats it. If not, the pet decides if it is a thief (if yes, eats it) or if it can select something else to eat. If it can select another instance, it restarts the decision tree. If it can not select another, it checks to see if it can beg for food. If that does not work, the pet must decide if it is desperate (eating the instance) or if it can wait (exits out of the tree).

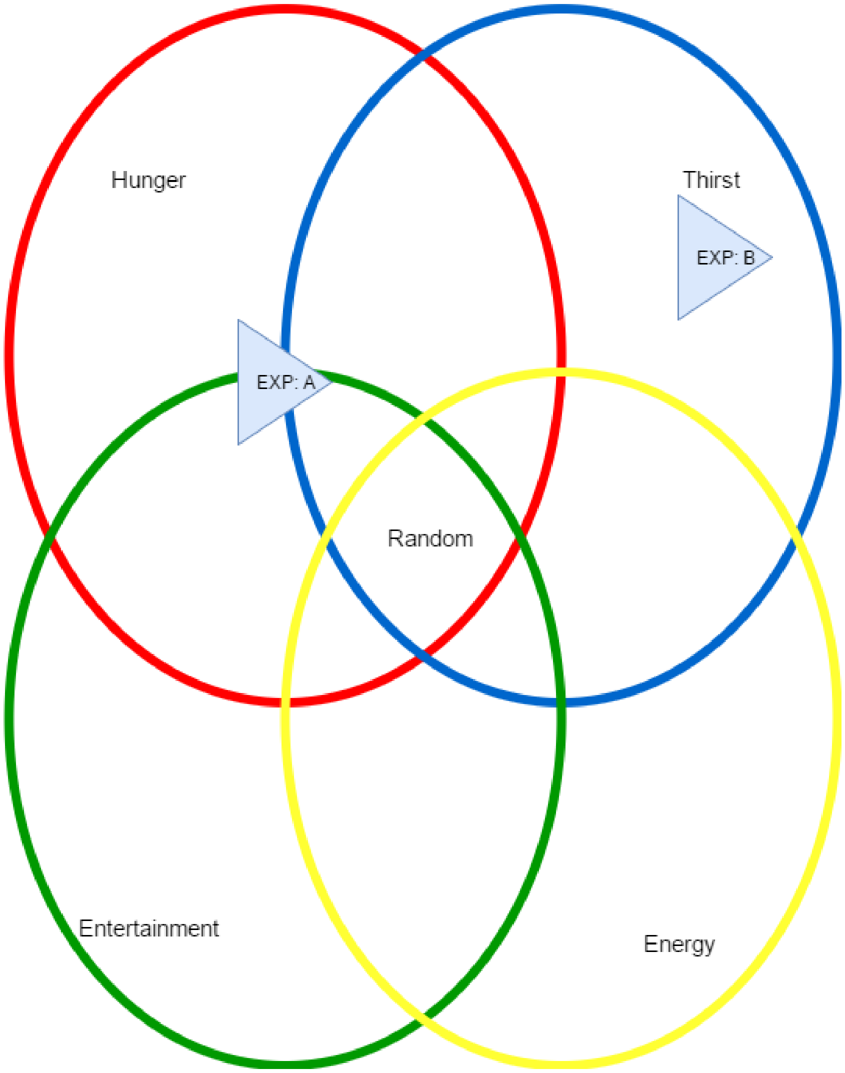


- Neural Network



  - Three Player Feedback Actions that change the desirability rating of the instance in the Value-Attribute Pair (For example, Praise the pet when it eats on a PetSurface (a Flag), and the desirability for eating on a PetSurface increases).
    - Praise: increases the Desirability of the Attribute-Value Pair by a lot
    - Punish: decreases the Desirability of the Attribute-Value Pair by a lot
    - Ignore: increases the Desirability of the Attribute-Value Pair by a little
- Fuzzy Finite State Machine: Determines the primary need from the weight of the membership. For example, in the figure below, Exp A has a strong membership in the Hunger set, a weak membership in the Thirst and Entertainment set, and no membership in the Energy set. Therefore, a Hunger action should be chosen. Exp B has a strong membership in the Thirst set, and no other memberships. Therefore, a Thirst action should be

chosen. If it comes down to being a balanced choice, choose randomly.



- Pet Model
    - The player's pet will be a dog, though we are considering supporting other pets. The pet will support animations for walking, looking at the player, eating, drinking, and any interactions that the player can make with the pet, such as being pet, watching thrown objects, or doing hand-stands when told.

VR implementation

- For VR implementation, the Unity game engine makes things easy by rendering scenes to the VR headset automatically to where the user can look around. Scripts attached to the camera and to objects in the virtual world are used to handle interaction. For instance, when the user looks at an interactive item, a ray is cast from the eye to the object. The object detects the ray colliding with it and then may call a method to do something. Another script is used to detect other actions the user might perform, such as clicks, swipes, or trigger presses. For detecting the location of the user's controllers, Unity supports an api that allows us to read the input of each touch controller, and the

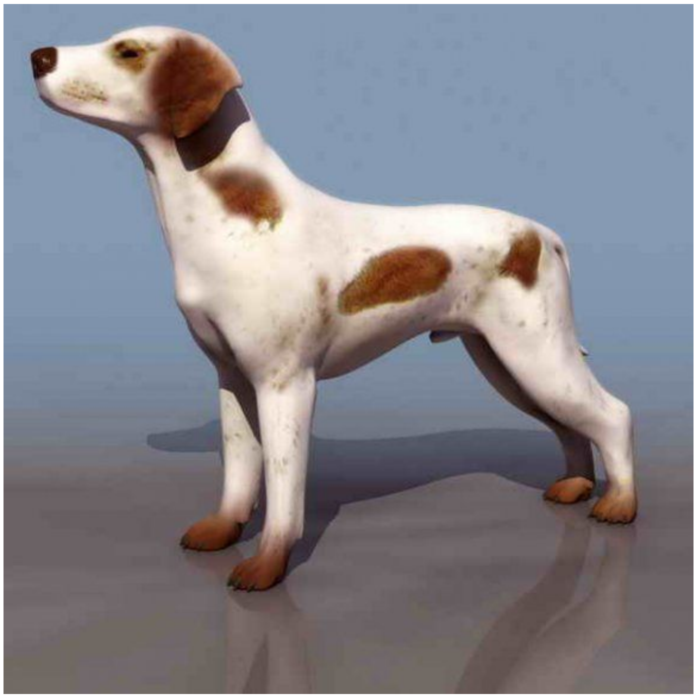
HTC Vive's sensors and api will allow us to track the location of each controller to use with our software.

Game model

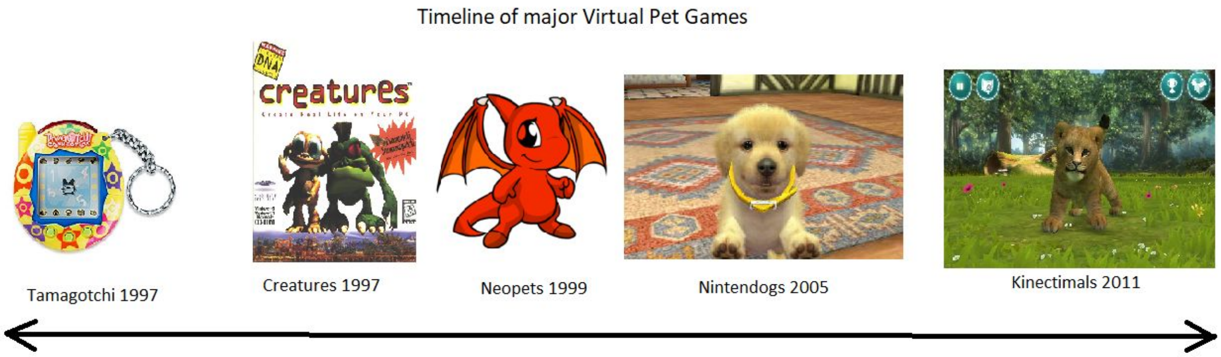
- The game model will abstract away the details of what happens when the player or pet perform particular actions or queries.
- When the pet looks for food sources, it will make a request for a resource object of type "food", given its position, and the engine will perform the search, and return an optimal path.
- Pet, and possible player, movement will be governed with Controllers that support movement methods that will queue up the action to move to a particular object or location. The controller can return an event upon completion that the Pet or Player objects can then respond to with notifications or further actions. if the Pet needs to climb up something, the controller will determine if it can, and send a failure event if it cannot.
    - NOTE possible AI learning opportunity - there is a deviation range within which pet may "think" it can/not climb up something, where it will try to climb very cautiously, and give up after a certain failure count, or learn that it can climb/jump that height.
- The game will use Unity's built-in physics system to support gravity, and physics-based object motion, but will selectively disable it to handle carrying and using of objects if we support the player doing hand interaction with the world.
- Primary Goals:
    - Pet will wander naturally when idle
    - Player can call pet to come closer
    - Player can play with pet through various toys and motions
    - Player can feed/water pet through placing objects in world
    - Pet will notify player of needs that cannot be met currently (missing water source, wants to play)
- Stretch Goals:
    - Player has some means of navigating room
        - teleportation
        - direct travel (may cause nausea/disconnect)
    - Player can earn income to buy toys and supplies for pet
    - Player can take pictures of pet
    - Player can take pet for walks
    - Multiple pets

Real-world mapping

- By using the lighthouse cameras from the HTC Vive, we are able to map the room into virtual space and create a semi realistic representation of the real space for the pet and the player. Unlike the Oculus Rift, the HTC Vive allows the detection of space and player movement by only using two cameras instead of the normal three sensors. Mapping the room to to virtual space gives the user a sense of familiarity and makes the virtual pet seem truly real in the user’s living space. Making the user feel comfortable in a virtual space is a difficult task, especially since the user is wearing a bulky headset to immerse themselves in the world. Virtual mapping is just one way to make the gap between real and virtual smaller. Instead of having a pre-made space that the user exists in, we can emulate an environment that they are already comfortable in.



This is a placeholder taken from an open-source 3D modeling site, and is of the quality I expect to achieve early-midway through the project. If time allows, a better quality in-house model will be used.



- Our project aims to capture the successes of some of these games.
  - Tamagotchi introduced the idea of a pet that you had to take care of, and showed that pet simulators could be very successful games.
  - Creatures showed that it was possible for virtual creatures to be intelligent and adaptive organisms that learned from conditioning and experience.
  - Neopets created the model for pet games used today, where players accumulate cash that they spend to feed, take care of, and dress up their pet.
  - Nintendogs extended pet interactions to create a semblance of realistic pet interactions.
  - Kinectimals showed that virtual reality software (In this case, the Microsoft Kinect camera) was an effective medium to bridge the gap between virtual creatures and real people interacting with them.
  - We plan to create a game that uses VR and pet artificial intelligence to create an immersive pet simulator.

Ethical and Intellectual Property Issues

- Ethical Issues
  - One of the ethical and moral issues that should be addressed regarding our project is that we utilize real-world mapping technology. This could be a concern for the user, as we are essential forming a rough blueprint of their living space when the user sets up our software. Which is why we will be handling this issue with a formal disclaimer before they use our software, in an effort to be as transparent as possible with the user. Additionally, we will handle the data that our software captures in an appropriate way, which may mean not saving the information unless the user requests it to be saved.
  - Another concern of ours which we should be transparent about, is that this is a VR project, and, as such, all of the disclaimers associated with using VR fall on us as well. It is important that we also warn and make clear to the user all of the concerns that one would consider when using any VR technology, such as nausea, damage to surroundings, etc.
  - Considering how our project has something of an AR aspect to it as well with our real-world mapping technology, some additional disclaimers will need to be

    provided to the user. One such disclaimer is that the HTC Vive still utilizes wired technology, and so our project will require the user to remain in the area that the Vive’s cord reaches.
    - Additionally, our mapping technology will have to be carefully tested and debugged to ensure that if a user is about to hit or touch something in the real world, that is accurately reflected in our software.
- Intellectual Property Issues
    - The software that we are developing will be our own intellectual property, however, this will come with some caveats due to the usage of third party software and resources in our development process. One such caveat is that we are using the Unity Engine for our project, specifically Unity Personal, which requires that our software cannot generate an amount larger than $100,000 US in annual gross revenue or raised funds. This particular requirement will not be of too much concern to us, as we plan to advertise this software as free and open source.
    - As our software is original and not derivative of any other existing copyrighted software, our copyright rights and protections will lie with us as the owners of the original work, meaning that we alone are entitled to the reproduction, distribution, public display, public performance, and production of derivative works.
    - An intellectual property concern for us to keep in mind is that algorithms can be patented if it is embodied in a machine. Considering a sizeable portion of our project is AI, we will need to be careful to not infringe on the rights of any inventors of such patented algorithms.
    - Additionally, the persisting concern of software development is still important for us to bear in mind, and that is that our code should remain our own, and we should not plagiarize when we are developing our algorithms and implementations of our project.

Change Log

- Moved “AI expanded for more interaction” back to 3/30/2018 to ensure completion fell on a Friday.
- Moved "Hello World VR" back to expected completion 10/27/2017
- Moved "Player movement through room" back to 12/29/2018 to reflect low priority and scheduled milestone.
- Expanded Pet AI: Research was done and fleshing out the AI.
- Moved “Room mapped into virtual space” back to 10/27/2017 due to lack of required hardware (HTC Vive headset).

Works Cited

Davis, Jeanie Lerche. "5 Ways Pets Can Improve Your Health." *5 Ways Pets Can Improve Your Health*. WebMd, 2004. Web.

"Healthy Pets Healthy People." *Centers for Disease Control and Prevention*. Centers for Disease Control and Prevention, 30 Apr. 2014. Web.

Oaklander, Mandy. "Sciene Says Your Pet Is Good for Your Mental Health." *TIME Health*. N.p., 6 Apr. 2017. Web.

Shatzman, Celia. "How Pets Improve Your Health." *Health.com*. N.p., 06 July 2015. Web.