Final Project Proposal - CART 398 Maya Marshell &Leonardo Morales

Self Locator - A third person embodied ML experience

Artist statement:

Self Locator is meant to be an exploration of gameplay as performance, asymmetrical VR local multiplayer experiences, and what real-time ML driven behavior can afford to a shared visual performance. Metaphorically is meant to vaguely (and abstractly) represent the continual act of self location in conscious experience, the way we navigate mental maps and models of reality using artifacts of 'semantic descent' from one representational medium to another.

As described in Jenann Ismael's The Situated Self, a red dot within a map serves as an anchor, a fixed point, between the map representation of a space and the space itself, in relation to one standing at that location. In the same way, a red sphere and a 3d map or level may serve as a mental space for the manifestation of two players' intentions. One player in VR has control of the sphere while another viewing a 2d screen controls the visual state of the environment remotely through an interface. The two players must coordinate to define location-to-visual configuration mappings to iteratively train an ML model in real time.

Technical Intentions:

The intention of our project will be to use Open Sound Control and Interact ML to create an interactive 3D space inside of Unreal Engine. We would also like the space to be accessible through the use of an Oculus VR headset with accompanying hand controllers. The initial goal will be to create a simple space inside of Unreal that will have certain simple, manipulatable variables such as the color of the lights or certain particle effects. Using a regression model from the Interact ML Unreal Engine plug-in, we will train the model to change these variables based on certain parameters (likely the player's X and Y coordinates). We could also possibly train two different models, one based on X and one based on Y, so that there could be two variables changing at once. Another alternative is to use Unity, in the case that the Interact ML plugin proves to be more stable in that platform.

Based on the engine crashing with our first attempt there are two routes we can take to address the problem. My first idea was the start with a smaller gamespace originally so that hopefully, when trying to train the algorithm, there will be less information and therefore less computer processing demand. We could also try a different exercise (we only tried one of the many possibilities of the Unreal x Interact ML plug in) to see if all will crash the engine or not. My other idea was to not use Interact ML at all and try to use OSC and Wekinator to try and do the mapping. We would need to find the port number and find out how to map the inputs into wekinators but after that it would be pretty easy to move the character around and train it in different areas. This would also mean that the input of Wekinator and the output would both be Unreal Engine. We would need a way for one port to send information about the XY position to wekinator which would then send the information back to Unreal in the form of lighting change and particle effect.

There is also the possibility of talking with the Interact ML guest speaker about it. We are using Unreal Engine because Leo knows it but if we had to we could switch to Unity since there is already a solid plug in for Unity (Interact ML is actually made for it so would likely not crash the computer) and I have used it a few times before.

We are seeking assistance from Marco Luna, coordinator of the Immersive Storytelling Studio, and intend to meet with him sometime in the next few weeks to learn more about the VR aspect of this project. What we learned from our chat is that in order to make this into a virtual reality experience we would need to take time to adjust the environment of the Unreal map to work inside of actual physical barriers. We need to figure out if this would change the XY input because that would change the way the ML algorithm functions. I would also need to use the headset a bit and learn more about it. Another question is will the position of the headset or the

hand controllers create multiple XY coordinates. This would mean we have to figure out which coordinates to use.

The VR lab has an HTC Vive oculus which comes with a head set and two hand held touch controllers that track your hand movements in space. The hand devices are also used to move your character in programs. The two inputs of these devices together determine X and Y value. There are two types of VR game experiences: stationary or room-scale. Ideally we would be using room-scale but that requires mapping to physical locations and will likely be much harder to pull off. Both would work for this project as with stationary you can still move the character inside of the game space.

Goals and Subgoals:

- Create a reactive 3D environment, a playable 3rd person character, and implement a trainable ML model that maps player position to changes in the environment.
    - Create a base layout of the scene.
    - Implement 3rd person movement controls.
    - Create parameterized shader assets, lights, and particle VFX assets.
    - Implement Interact ML realtime training and sampling.

Timeline:

- The beginning of our timeline (working on the actual game part of it) will be the first two weeks. I am hoping to be able to start actually using the programs next week so we can try out a few different attempts. Once we get informed about the best routes of

development with Interact ML in class, we'll have more clarity on the engine we should use. After we decide I can start learning more about the tech in question.

- The second half of the project (the VR) will depend on when we can get a meeting with Marco Luna. We can also go visit him in the VR lab but since we haven't got a game to plug into the software it is less urgent.

References:

Technical references:

- Neural Networks Engine | Epic Games Game Dev https://dev.epicgames.com/community/learning/tutorials/34q9/unreal-engine-nne-quick-start-guide-5-3

- Catlike Coding | Unity movement tutorials **https://catlikecoding.com/unity/tutorials/movement/**

- [Advanced Scan Effect PART 1 - Inputs, Blueprints & MPC](https://youtu.be/f5diUsuftEk?si=IqTBDFOUAiPGX3Z9) | [Ghislain Girardot](https://www.youtube.com/@ghislaingirardot)

- - [Unreal Engine 5.3 - Introduction To HLSL & Scratch Pad In Niagara FX](https://youtu.be/ZNPzpXKvyL4?si=ksjnKMQj2tXugo9B) | [renderBucket](https://www.youtube.com/@renderbucket)

- - [Unreal Engine 5 | Blueprint For Beginners (2023)](https://www.youtube.com/watch?v=Xw9QEMFInYU) | [SmartPoly](https://www.youtube.com/@SmartPoly)

- - [InteractML | Interactive Machine Learning Visual Scripting](https://interactml.com/)

- - [InteractML : Interactive Machine Learning System | UE5 plug in](https://www.unrealengine.com/marketplace/en-US/product/interactml-interactive-machine-learning-system)

Thematic & artistic references:

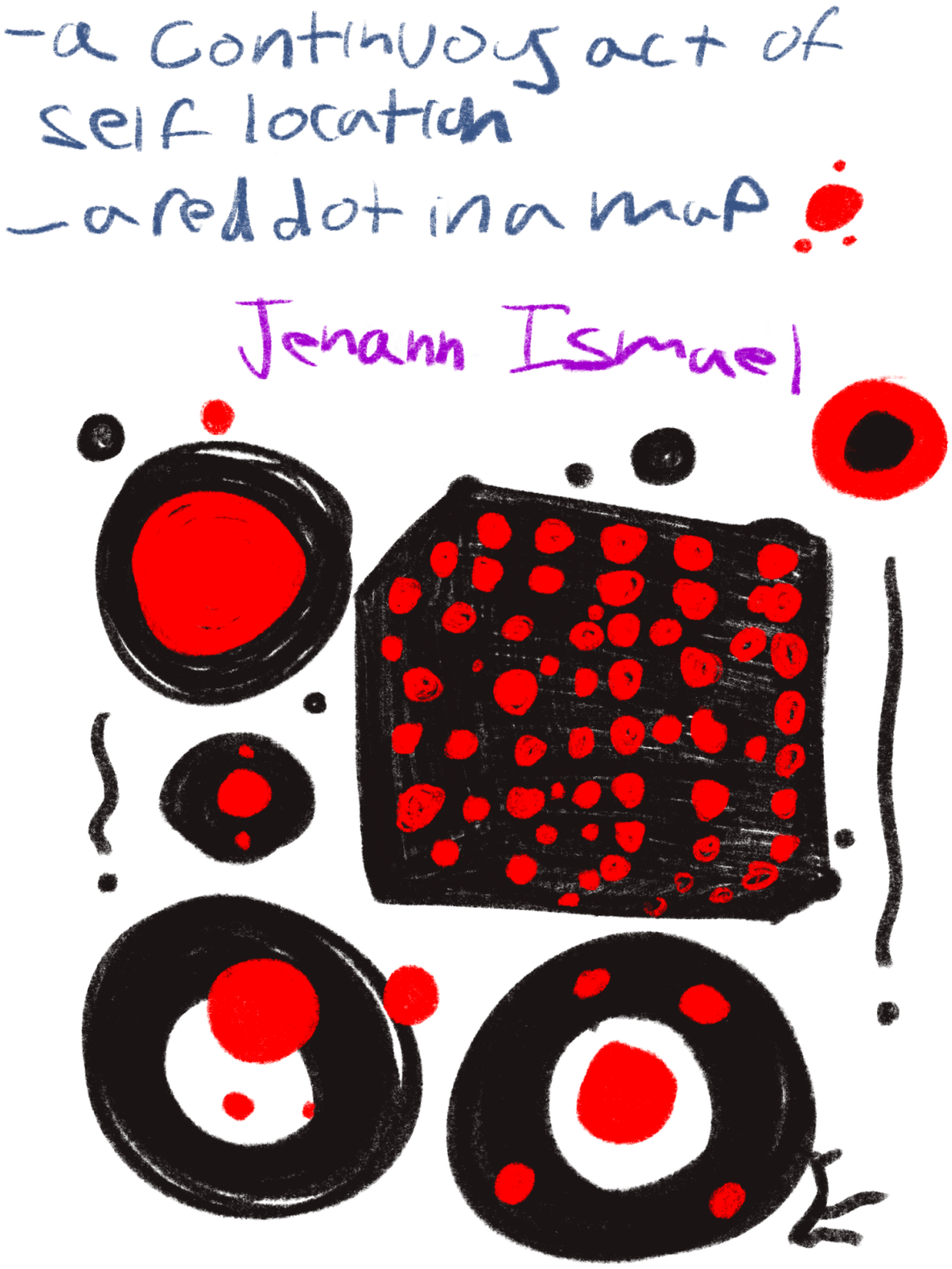
- [The Situated Self, J. T. Ismael](https://academic.oup.com/book/2503?login=true)
    - For the philosophical background of representational mediums and the nature of experience.

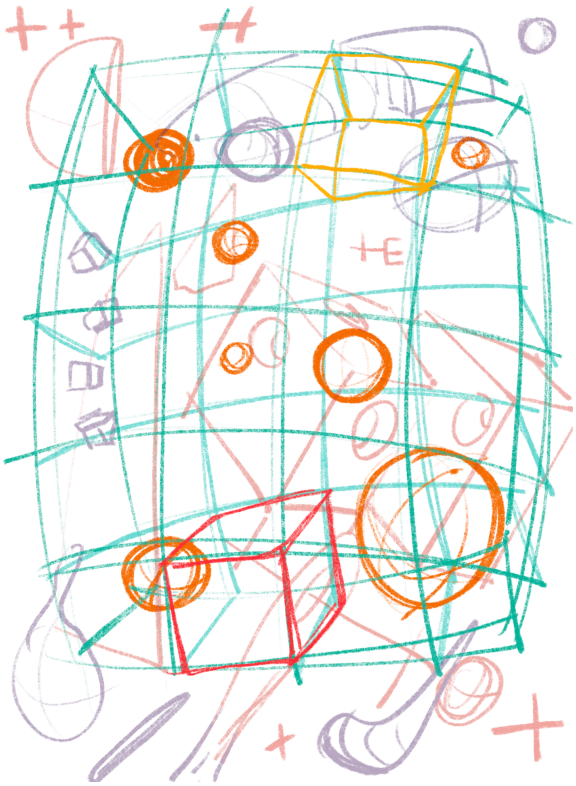
- Moss. https://store.steampowered.com/app/846470/Moss/
    - For the integration of 3rd person game mechanics in a VR setting.

- Sekamol. https://www.instagram.com/p/Cn7KpsNMrAL/?img_index=1
    - For the surreal generative 3d effects.

    More pieces by Sekamol:
    - https://www.instagram.com/p/Cn7KpsNMrAL/?img_index=1
    - https://www.instagram.com/p/CnuvlvsJCtG/
    - https://www.instagram.com/p/Cmb9b2tpVsZ/
    - https://www.instagram.com/p/CqpkPyJI9sT/
    - https://www.instagram.com/p/CnynC-4gdjn/
    - https://www.instagram.com/p/CnPHVL7ptP9/

Ideation and concept sketches:





Grid Space (Maze?)

Grid maze

- an embodied Game_Performance interaction. Driven by Dynamic real time ML
  - Shader and Particle FX
  - Position based mapping. (real time training?)
  - 3D Grid based Level Design.
  - Abstract Graphics.
  - Gameplay as Performance.

SelfLocator