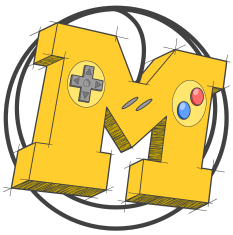

Lecture 6

Announcements

- From gamedev jobs to laboratory opportunities on-campus, use the Get-A-Job guide to help you find opportunities in XR / gamedev.

- Subtle yet impressive sheep AI.
    - Machine learning? Nah.
    - Almost certainly using a boids variant in combination with an avoidance velocity that is a function of distance-to-car (with a slight bias against running directly away from the car in a straight line. This makes it difficult for players to construct a scenario where a sheep may be seen running at 100+ mph).
    - The real test: Get multiple cars attempting to corner the sheep.

- Massively Multiplayer RollABall
    - Sometimes, all it takes is the unique combination of two simple ideas:
        - Marbles + Many of them = wild success.

Story Time : The Value of Iteration in *Greek Tragedy*

- Greek Tragedy's environments were never very good, because I'm not an environmental artist.
- One day, my teammate (an artist) begins making improvements (before and after).
    - He discovers a cool new asset that allows him to place organic ivy / vines in the environment.
    - For several days, I passively watch the screenshots rolling in and feel joy in my heart.
    - Our iterative cycle demands I playtest the game completely at least once every week.
        - I notice something strange– our loading times have jumped from 3 seconds to 30.
        - Because the aesthetic improvements were the only major change that took place that week, they are the immediate suspect.
            - There are 100+ ivy gameobjects in the scene, but they look as if they should be very lightweight and not a computational or loading / storage burden.
            - Memory / RAM profiler proves me dead wrong. Each one requires 2-4 MB (!?) and at 100+ of them, this leads to 200-300 additional MB in one additional scene.
            - **Solution :** Have partner go back through, compressing ivy data, textures, and removing some meshes. Result is an almost-same level of quality, but we're back to 3 seconds. We were fortunate this was possible.

        - **Outcome if not caught early :** Art team would have had to redo a significant amount of work, and other members of the team (such as me) would have seen our testing / iteration times increase by at least 30 seconds **per test**.
- **Takeaway :** Keeping a tight iterative process in which the build can be played / is never “broken” for more than a couple days is extremely beneficial. It allows issues to be caught early when they’re much easier to diagnose and fix.
    - **AAA Studio of a recent release my students were on :** Game was unplayable for months at a time. Dev team members couldn’t test their code or develop properly. Absolute chaos. Lengthy and lethargic iteration cycles. By the time they got the game to finally work, they were out of time to actually make it fun.

Game Design : Interesting Decisions

- Slides