Final Project Prototype

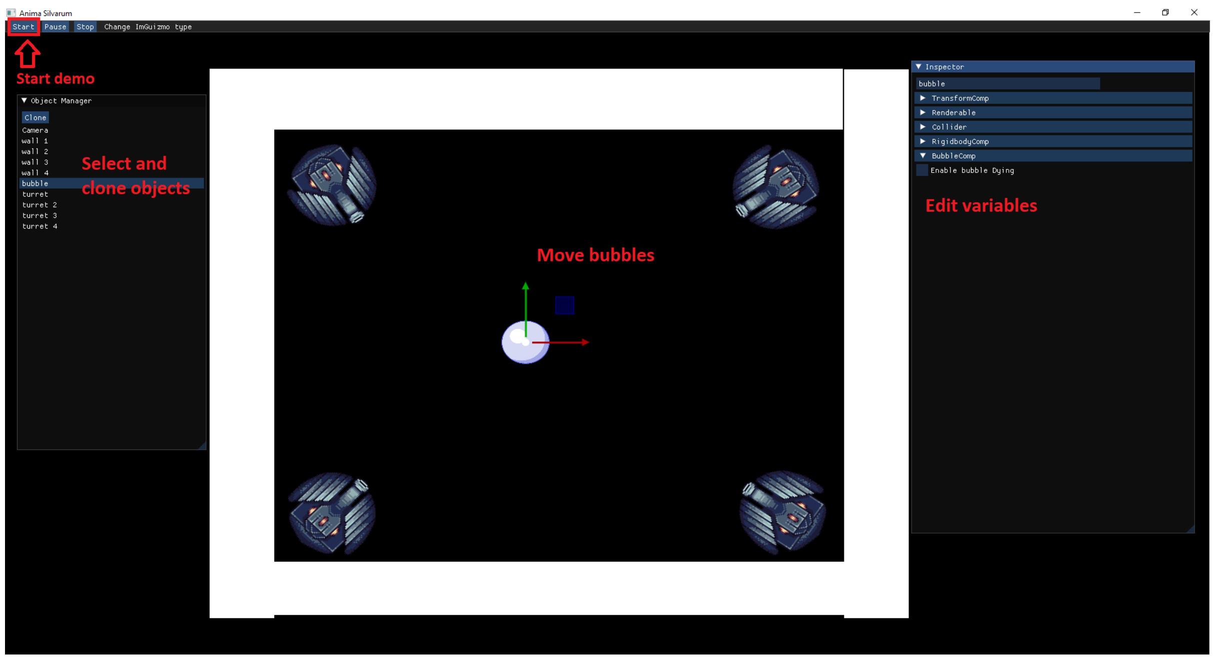
Alex Ceberio & Iñaki Soler

Instructions to execute and interpret the project

Run the ProtoypeBuild.exe. There are some features available to edit the demo. (Moving things around, duplicating objects, modifying the speed of the turrets).

Use the start button to start running the level, pause to stop the level and open the editor for that frame, and stop to reset the level to its original state.

After starting the simulation of the bubbles' AI pressing start, you just have to observe them (it's not an interactive demo). Press *esc* to end the demo.



We recommend just moving the bubble's position and cloning a few of them, but you can change whatever you want.

What code is multithreaded and why?

The logic of the bubbles is AI based which includes simulations and run-time interactions with the environment. The bubbles need to know when they are in danger to be hit by a bullet and how to avoid it, choosing between three possible methods explained on the proposal of this project:

- Dodge
- Join other bubbles
- Split into two bubbles

The simulations for choice-deciding and executing are very expensive, so they are done within a specific function called in different threads (AEX::Dodge function inside BubbleComp.cpp). Whenever a bullet is shot by any gun, each bubble creates a thread to make simulations and act depending on the results while the game loop keeps running simultaneously. This makes a huge optimization on the game flow.

What challenges do we have to overcome?

The demo may sometimes crash due to clashes between the engine and the multiple threads. If for example, a new bubble is created while the collisions are being updated, the collision list will be modified while on a for loop, which will obviously crash it. This also happens with many other systems on the engine that we'll need to take care of as new bugs come out.

We hope to fix those issues by adding more sanity checks, paying attention to shared memory and avoiding simultaneous memory accessing by using mutexes, for instance.

Furthermore, we expect to optimize our demo a little bit more by loading the assets in a multithreaded way.