

Moving Blocks In 3D with hand-tracking

Project Goals:

- Create a simple, intuitive (but hopefully fairly flexible) method to position blocks for 3D "Legos"
- Ability to render it for stereo vision using colored 3d glasses
- Ability to control it using handtracking via the nuwii project
- possibly headtracking, though this will depend on if we can get more than two points per wiimote to show up, and also on other parts of the project.
- Implementing a basic level editor.
 - Placing blocks, adding blocks, removing blocks, moving blocks, etc
 - undo/redo
 - camera control
 - save/load/new
 - export to xml
- Blocks will function like legos, ie, they can be attached to each other to form new rigid bodies
- Simulation for this class will not focus on dynamics, although it is my intention to focus my computer animation project on that task, in order to allow buildings with destruction
 - such blocks will treat the legos such that the pips at the top would be massless, and the block would have a uniform mass, i think, as the actual internal structure of a lego is rather complex and uninteresting for an intuitive program, i think.

Rough timeline:

- before I leave for break[the best laid plans...] - get nuwii working and record some sample gesture data to play around with over the break.
- over the break[divide and conquer]:
 - *get the basic level editor functionality working with non-exotic input methods, give some thought as to what sorts of gestures i would like to use to control the app, and integrate some of the test data so my app can use the input
- [week4](after the break)(the integral from week 3 to 4]:
 - *begin implementing and fully integrating the ideas for user interaction over the break
 - *though, possibly subject to overflow on the editor
 - +critically though, start testing some of the interaction ideas and get the whole thing really integrated
- week 5[the point of no return]:
 - *work on anything that might be a problem down the road, and make sure there will not be any surprises down the road [ie, make sure the project is still realistic, and try hard to tackle anything that seems like its going to be a problem]
 - +revise any plans i can't figure out how to fix/implement.

- week 6[...and so crunch mode begins]:
 - *continue implementing the features.
 - *map editor should be usable by the end of this week
 - *gestures should be usable by the end of this week
 - *stereo rendering should be ready by the end of this week
- week 7[get the noob(s) to break it]:
 - *find some students to playtest it, with an eye towards beta testing and ensuring usability
 - *try and fix any thing that comes up
- week 8[polish]:
 - *continue fixing problems found in week 7
 - *continue improving features and fixing bugs
- week 9[almost there...]
 - *try and get some more user testing in, if possible, to see how the changes went
 - *test like crazy, try to break stuff (rather, to find broken stuff)
 - *small tweaks to improve usability
 - *triage any rotting code and tie off any loose ends or unfinished features
 - *work on xml export and such
 - *look up imagine@RIT, etc
- week 10[overflow]:
 - *general work overflow (in case i get behind, this will be dedicated to catching up)
 - *code cleanup
 - *documentation
 - *other niceties
 - *make sure stuff is fully cited and other good stuff
 - *etc.
- week 11[where no java application has gone before]
 - *deployment?
 - *fancy sounding tasks?
 - *stuff?