

Meeting Agenda

Date: 2015-05-12

Facilitator: Jesper Olsson

Participants: Fredrik Bengtsson, Jesper Olsson, Anton Strandman, Ulrika Uddeborg

Objectives:

§1 Decide how to set where nodes should be created in NodeFactory

§2 Discuss how we would like to implement health etc. in the characters

§3 Decide what to do until next meeting

Reports:

- Make NPC
 - NPCs have arbitrary values assigned to their fields.
 - Added case for NPC in factory. Refactored code. Should different NPCs have different NodeType?
 - Does not yet have graphical representation
- Make NPCs spawn
 - now possible to spawn spawnables from spawnpoint without jme3 code
- Continue with SDD
 - Worked on point 2.4 Code, added text for many of the classes describing what they do and their purpose. Also approved some comments and added some information in other topics.

- Start working on AI algorithm
 - Have explored a couple of different ways to make it work, but have not yet decided on the best way to do it.
- checkpoints and spawnpoint fixes and physicswrapper
 - done, minor issues magically appeared afterwards.
- Refactor playerMoveControl
 - Added booleans for left and right (physics is in charge of up/down)
 - Made abstract class (AbstractMoveControl, currently in the same package)
 - Control Update now moves the player
- Create NPCMoveControl
 - Extends AbstractMoveControl
 - Only needs AI code
- refactor model
 - done, it is now more beautiful than before. still not beautiful enough
- Other
 - Anton has found a tool to create UML: Code 2 UML

Discussion items:

From the paragraphs above:

1. every spawncontrol is attached to a spawnpoint, do

```
Node node = nodeFactory.createModeledNode(type);
```

```
this.getParent().attachChild(node);
```

```
node.setLocalTranslation(this.getLocalTranslation);
```

kinda, sorta, something like that.

2. Keep it simple, the game does not focus on combat. Make each point of health correspond to one hit.

3.

Other:

- Code 2 UML
 - Tool that displays UML for code.

Outcomes and assignments:

- AI - Ulrika
 - Create action-enum class
 - IDLE, MOVE, MELEE_ATTACK, RANGE_ATTACK
 - AbstractNPC will implement a new interface: IAI
 - public IAction getAction()
 - public void updateAction(float distance, NodeType nodetype)
- Look over all classes implemented interfaces - Jesper
 - Use Code 2 UML
 - Make target (HostileNPC) not final
 - Refactor factory in a concern (public static) with separate factories (protected) which it delegates to
 - Ask TA about the static values in e.g. Minion
- SDD - Anton
- Finalize spawn- and checkpoint - Fredrik
- Scenecomposer -Fredrik

Wrap up:

Next meeting: 2015-05-13