Teachable Agent for Elevdata’s Matematikhuset

Layla Husain, Necmettin Meletli, Mille Boström, Kristofer Axelsson, and (Rikard Samvik)

November 2010

Lund University
Abstract

One of the most effective ways to learn a concept well is to teach it to someone else. Learning with an agent has proven to be motivating and facilitating for students. A teachable agent (TA) is a virtual character that can be taught and then tested for its knowledge. In our program, 4th graders teach the TA (that they can personalise) math skills in the hopes that it can perform exercises from Matematikhuset. The program, Smart N Up, allows the user to interact with the agent one on one, or in a group setting. It also has a social network where the user can chat with friends, do group work, view each others’ profiles and play games that require math. Not only are the activities and performance levels saved in the user’s profile, but also on CAPA, a database used by teachers. Our aim is to immerse the user in a world of math, where they can benefit from learning by teaching, peer learning, as well as injecting fun to increase motivation. We attempt to venture from the typical educational game to ‘edutainment’, mindful of the delicate balance between education and entertainment. The effect we hope to achieve is an increased intrinsic motivation, self-efficacy, confidence in math skills, as well as a desire to excel, especially for students who struggle with it. In this paper, we present the conceptual design of our program based on studies, proven pedagogical methods, and successful commercial games. We also present the portions of the concept that have been implemented. Although the ultimate test is the children’s approval, the responses so far have been positive. Still, there are some lingering issues that we discuss and options for further development.
**Table of Contents**

1. Introduction..................................................................................................................... 4
2. What we did .................................................................................................................... 5
   2.1 Conceptual .................................................................................................................. 5
       2.1.1 Interaction & Teaching ................................................................. 5
       2.1.2 Feedback................................................................................................. 8
       2.1.3 Motivation..................................................................................... 10
       2.1.4 Social Network .................................................................. 13
           2.1.4.1 Games ........................................................................ 14
   2.2 System Implementation ............................................................................... 18
3. Issues That Remain .................................................................................................. 21
4. Conclusion ............................................................................................................... 22
5. Future Work ............................................................................................................. 23
6. Our Team .................................................................................................................... 23
   Figures..................................................................................................................... 24
   References................................................................................................................. 27
1. Introduction

If you think back to the last class you took in school, you can almost hear your classmate rushing up to you a day before the exam panicking about not being ready. Being the responsible friend that you are, you take out your notes and explain the concepts. As you explain, you make new connections, and come to a whole new level of understanding, making you even more prepared for the exam. You also realise which points you need to review if you have gaps in your knowledge. This opportunity to teach another is a very effective way to learn and solidify concepts.

A Teachable Agent (TA) is a virtual character than can be taught and then tested for its knowledge. In our example, elementary school children would teach the TA mathematics skills and test its knowledge through solving Matematikhuset exercises. “TAs do not replace real students. But, they do provide unique opportunities to optimize learning-by-teaching interactions. TAs, for example, always make their thinking visible, something that not all students can do” (Blair et al, 2006, p1). Moreover, TAs can provide metacognitive support by wording concepts in a way that allow the child to see another dimension of the concept learned at school, or to make connections to other concepts they come across. Transferring their knowledge to a TA is considered proxy learning since the TA’s knowledge mirrors the student’s.

Chase et al (2009) notice another benefit called the protégé effect, where “students are more willing to make the effort towards learning on behalf of a computerized protégé than for themselves. Three factors may contribute to the protégé effect: an egoprotective buffer, the adoption of an incrementalist theory of TA intelligence, and a sense of responsibility.” (Chase et al, 2009, p2). Studies have shown that this effect is even more pronounced for low-achieving students, who are our target group. We wanted to find a way to encourage and motivate students who do not feel confident about their math skills and show them their potential. Learning with a virtual pedagogical agent has proven to ‘lead to an improved perception of the learning experience and that it makes the tasks seem easier, which increases the student’s motivation.’ (Andre et al, 2009, p1).

In order to truly motivate students and help them achieve their full potential, it is important that the program be infused with motivation, helpful and meaningful feedback, a social network, as well as a rich graphical environment. All of these factors help engage the child and keep them coming back to the program. Hopefully, they won’t even realise how much they are learning while having fun in a game context.

This report will begin by describing our program in section 2. First, we describe the conceptual idea (and goal to work towards) in 2.1, then what was implemented in 2.2. Next, we discuss lingering issues in section 3. In section 4, we conclude our thoughts surrounding the program and suggest tips for future development in section 5. We end with a description of how our team functioned in section 6.
2. What we did

2.1. Conceptual

Our program, Smart N Up, consists of two main components: teaching and social network. In the teaching component, the user teaches a teachable agent that can be personalized. Once they are taught, they can be tested for their knowledge. The social network is equipped with chat, a newsfeed of updates from friends, collaboration tools for group studies and math games. We assume that our users are children in the 4th grade who have already learned a mathematical concept in class and are now solidifying understanding, as opposed to simultaneously learning a concept, such as in Billy’s Brain. For instance, the 4th grade class has just learned fractions in school, and is ready to start practicing with the agent. We considered how the user would interact with the agent over time, choosing the character, how to motivate and engage the child, giving appropriate and meaningful feedback, and the features of the social network. For a better idea of what it looks like, please refer to our lo-fi prototype. This section begins by outlining a way to interact with and teach the agent, as well as issues that arise (2.1.1). Next, we discuss the forms of feedback and how it helps the user in 2.1.2. Following that is a discussion on motivation and how it is incorporated in Smart N Up (2.1.3). After that, we describe the second component of the program, the social network (2.1.4) and the games we included.

2.1.1. Interaction & Teaching

In the very first interaction between the user and teachable agent (TA), the user chooses their agent. Each character is animated and has a personality that is reflected in their mannerisms, facial expression, gestures, and style. These particular examples are from the Wii game Buzz! One character is geeky and always wants to get things right. Another is a cheerleader and is always excited and energetic. The list continues with different styles that can be fun for the children using the program. Please refer to figure 1. We suggest naming the characters after famous mathematicians, or word play with mathematical terms, to engage the children in the world of math. According to Lisetti (2009), children relate best to characters that are close in age and personality, so it is important for them to have such options.

The user can also learn more about each character via their ID card. It displays their name, age, likes, dislikes, hobbies and more. Please refer to figure 2 for an example. Along with each personality are matching mannerisms and gestures. For instance, the cheerleader is usually cheering, sounding ecstatic and holding pompoms that she shakes around. The user will also see some options to personalise their character, such as facial features (moustache, small nose..), clothes, hats, and gadgets. The idea here is to further engage the user and to make them invested in their agent and game. Some of the options will be greyed out until the teachable agent (TA) attains a certain level of expertise, to encourage the students to excel. Burguillo (2010) is one of many who asserts that healthy competition enhances learning and motivation. Moreover, it is important to have a clear goal to work towards. We realise that calling these characters ‘teachable agent’ is not meaningful to the users, so what the user sees is ‘student’ or ‘character’. After all, the TA is an eager student waiting to learn from the user.

In subsequent interactions, the user will not have to choose the character again. If they would like to change it, they do have the option to do so in their personal settings. Once the character is established, the user selects the level and concept to teach. The level and concept layout looks just like Matematikhuset since the agent will eventually try out exercises from there. Next, the TA is ready to learn. At this point, two main issues come up: communicating with and actually teaching
the agent. It was difficult to decide how the user would interact with the TA. We would have liked to have a free text to accommodate different approaches to teaching the same concept. Unfortunately, with our current abilities and resources, this seems almost impossible to implement in a way where the agent would interpret and appropriately respond to the child’s input. We know it can be done and can even pass the Turing Test, such as with ELIZA. In our case, however, it is not so simple. There can be an infinite number of ways to explain more advanced concepts that make it very difficult to implement. We also looked to Betty’s Brain and Billy’s Brain for inspiration on how to interact. In both examples, though, the concepts taught are vastly different from arithmetic and so do not apply for our example. So for this particular project, we used a multiple choice format since it is easy to implement and is relatively suitable for the subject. It is important to properly phrase the multiple choice questions to maximise the pedagogical effect. The TA asks questions about how to go about solving a problem and the user chooses the correct answer to fill in the gap by dragging and dropping it into the correct slot. The reason we chose the drag and drop option is to support multimodal learning, which has gained popularity recently, as it supports all kinds of learning styles and needs. The creators of Betty’s Brain use the multimodal approach as the basis of their interaction strategy. In case the user makes a mistake, they can click on the ‘oops! Scratch that!’ button to undo the latest statement.

Issues that Arise
Our issue with multiple choice vs. free text as a form of interaction is that multiple choice questions may be too rigid and coached. It does not allow for alternative ways of explaining a concept and may not necessarily reflect the user’s thoughts. It could also be an option to begin with multiple choice, then move to freer text once the user is better acquainted with concepts. Moreover, we find that in Matematikhuset, the exercises provided are simply drills. What we would like to do is have the agent learn a concept, then practice with examples in order to improve. In other words, it would be a combination of concept teaching and practice in an iterative fashion to truly learn. This is because it mirrors human learning but also because it gives the user more time to practice the concept.

Another challenge was to think about how the user will actually teach the agent. It could be through visual representations, a narrative form, giving examples, or explaining the concept itself. Since the most likely situation is that the child would explain using examples, we can use that as a way to introduce the concept to the agent. Most young students who are relatively new to a topic, may not necessarily have the vocabulary and perspective to teach a concept, but are more familiar with examples they have learned in class. Then, the agent can make the connection to the concept themselves by noticing the pattern. For instance, the agent could say ‘Oh I see! So do you always take the top number and divide it by the bottom number?’ This way, it could be the agent that exposes the pattern to the child who can use new terms (still simple at the child’s level) to understand the concept. It can also be an indicator that the agent has not fully understood the concept. Another point discussed was if beginning with explaining using visual representations of numbers, how can the child then transition to teaching the actual concept as they advance through the stages. It is not sustainable for the student to continue using visual representations throughout their academic life. Having the agent reach the conclusion and phrase it as a concept can also help with this transition. The program also accommodates multi-step instructions for more advanced concepts. For example, ‘first, you multiply a by all the numbers in the bracket. Second, you add all the numbers......’ In all of these cases, the input would begin as concrete examples (either visual, or numbers...), then become a conceptual level input before the agent gives a summary. It was difficult to balance between wording questions as drills that they do in Matematikhuset and actually teaching the agent to solve questions. Please refer to section 2.2 for an example of a question.. We assume that the TA is not a complete clean slate and can do basic mathematical operations, such as adding, subtracting, multiplying and dividing smaller numbers. The extent to
which it is a clean slate is still under discussion.

Students must have the freedom to make decisions (correct as well as incorrect ones) and observe their results (Burton & Brown, 1982). A teachable agent provides this freedom for users since the performance of the TA is completely dependant on the user’s input. Moreover, the TA can show their reasoning and how they arrived at the answer, as well as repeating a summary of the concept learned. This was yet another question for us to think about. It was difficult to know if we should have the system have an indicator that could inform the user if the concept taught was incorrect. The concept of a TA is that it is under the control of the user’s input. To incorporate system feedback would be adding another role for the TA. We did implement part of this idea in our program just to see how it would appear. This is done by looking at the skill level of the agent as they progress through stages of the program. On the conceptual level, though, if the user incorrectly teaches the agent, the results of the test will indicate a fault. Then, the student can review the reasoning, or use the Teacher Assistant’s help.

**Getting Assistance**

During the teaching process, users should also have chance to get help explaining concepts. Instead of naming this feature ‘help’, we decided on ‘Teacher Assistant’. The wording here is key to keeping the user feel empowered by being an authority. All teachers need assistants, including those using Smart N Up. The purpose is to maintain their sense of ability. It doesn’t give the actual answer to plug in, but gives hints and strategies about portions of the concept to be taught. The Teacher Assistant is used for scaffolding and only steps in when the user asks it. The student can use its services only a few times before it becomes temporarily unavailable to prevent overuse and reliance. The assistant can be chosen from a variety of available characters that come from their world, such as experienced grandparents, animals, and more. Studies have shown that these types of characters have a positive impact on children (Rai et al., 2009). According to their study, children feel a stronger sense of responsibility to teach familiar characters and maintain their sense of well being. In this example, we feel that would encourage the user to improve the TAs math skills.

**Checking Understanding & Connecting Concepts**

Once a concept is taught, the TA repeats it as a summary, to help the user review and to verify or to correct the TA’s understanding. Then, the user has the option to edit, or do a practice question. While doing the practice question, the TA reasons its actions to review it with the user and to verify if it is indeed correct. If the user is satisfied, they can continue doing practice questions, edit the rule, or connect to Matematikhuset and have the TA practice there. If the results of Matematikhuset go well, then the agent thanks the student and is ready for the next concept. Otherwise, they can review the rules together until the TA gets it right.

Another feature is that the TA will have a ‘memory’ of previous interactions. This will come in handy when connecting similar concepts. For instance, the TA would say ‘Neat! this is very similar to when you taught me about fractions!’ This would help the user make such connections, and to get feedback on the TA’s understanding. It might show the user that the TA has made an incorrect connection and may need more teaching. Good feedback and markers of understanding in a conversation are very helpful to successful communication (Clark & Marshall, 1981).

At the end of the school term, the students can look at their ‘learning space’, which displays a summary of all the concepts the TA has learned so far according to different categories (division, addition,...). It is another way of organising what they know and a way to keep track of their progress.
Social Interaction
To create a connection between the user and the agent, we thought of including a little small talk, or social interaction. Agneta Gulz mentioned that it has a very positive impact on learning. In our program, it would keep track of the time stamp of the computer. The agent could then know if it should say good morning or good evening. It can also tell the user that they miss them if they hadn’t used the program for several days in a row. The agent can also remind the user to take breaks, if they have been using it for too long. This can be done through an energy meter, indicating that the agent is too tired to continue. At that point, the user would need to leave the program for the agent to recharge. The option can vary between logging out of the system, or playing a math game with the agent. This is to avoid overuse and dependence on the program. The point is for the user to improve their knowledge of math, then transfer that knowledge into the real world. Especially children have a difficult time with limiting computer use (conclusions reached from Agneta Gulz’s lecture on Sep 17, 2010).

Personal Settings
The user can tweak the settings of the agent and the program by going into their personal settings. There, they have the ability to change or personalise character and edit their ID information. They can also view statistics of usage, achievements, passes, fails, progress charts, number of times used Teacher Assistant, as well as other features. Moreover, they can control the text size, mute the agent, control the volume, and set the background music.

2.1.2. Feedback
Providing students with accurate, intelligent, and motivating feedback is a critical aspect of learning environments and is a vital component to the learning process. There are several components of feedback that may influence the learning process, including timing, content, control, and delivery-method of the feedback (McNamara et al., 2009).

Types of Feedback
Different types of feedback fulfil a variety of purposes. For instance, feedback can indicate that the listener is following along, which keeps students engaged. It can also reveal reasoning strategies in problem solving, or the understanding of a concept by summarising and making connections, allowing the user to verify or edit the rule. Feedback also relates to mapping, as it informs the user that their actions have been registered, and that the program is running as it should, such as system feedback. In an educational context, feedback on performance is essential. Nonverbal cues, such as gestures, and facial expressions, as well as verbal cues from the teachable agent or teacher assistant can be explanatory, corrective and consequence based feedback. Formative feedback, consequences of actions, diplomas and other ways to show success promotes the user’s self efficacy and sense of progress are also used. Further, comments from the agent about their success as a teacher and excitement about their own performance will also boost the user’s confidence and engagement.

Donald Norman (2002) also discusses three concepts that relate to feedback: mapping, visibility and feedback. That entails making interactive portions visible, mapping them to the appropriate consequences, and acknowledging that the user’s input has been processed. It also includes making the types of actions that are available visible and providing immediate responses to them. For instance, if something is clickable, it will be highlight when scrolled over and make a sound when it is clicked on. Moreover, the agent will always respond to the users actions. Each character is highlighted when selected and an information box appears. This system feedback can also include the agent saying “Yay! You picked me!”, or “I am ready to learn” to indicate the beginning of a session.
According to Nehaniv et al (2005), there are several types of gestures: symbolic, interactional, and deictic gestures. In our example, the TA can give a thumbs up, or a cheer to show they have understood or when getting a question right. The TA can also point at things on the screen or areas to draw the attention of the user. Moreover, gestures can be accompanied with facial expression, mimics or emotions, and tone. If students use the program for too long, body language can be used to show the agent is tired and may need a break to encourage the user to do the same.

In addition to the gestural feedback, we use explanatory feedback to show the TA’s understanding of concepts. This comes up in the ‘guess and check’, summary of the concept learned, and when connecting to previously learned concepts. Consequence based feedback comes in the form of a diploma for a successfully completed portions or unlocked bonus material.

Whilst learning a concept, the agent will show that it is following along and perhaps make connections to previously learned concepts. At the end of a concept, the TA will summarise and do a test to check understanding. Inactivity will also make the agent appear bored or like they are waiting for you to act. Verbal and nonverbal cues will be used to prompt the child to do something or that their actions have been acknowledged. Once a rule is taught, the agent will ask to be tested or try Matematikhuset. Before that, a summary based on the multiple choice questions asked will be displayed.

Appropriate feedback
When giving feedback, it is vital that the language used is appropriate for the age group and meaningful to the user. It is equally important that the agent be as encouraging as possible. We assume that most of the children using the TA are not confident about the math abilities and do not generally enjoy working on math problems. By having an encouraging TA, we hope to empower the children using it and have their opinions about math become more positive. Peter Gärdenfors (2010) also points this out in his book ‘Lusten att Förstå’. Over time, the way the agent speaks will reflect his/her growing confidence in math abilities and become more familiar with the user.

The teacher assistant must not give too much information, but guide the child in the right direction. It can be a character that is the same age or older. Since animals and familiar people benefit children, we thought having a ‘clever fox’ or a ‘wise owl’. As a familiar person, we could include a grandmother or grandfather. It is important, though, to keep in mind who the user is and what is appropriate for each age group. Since the assistant only gives hints (as opposed to the actual answer), the grandmother could ‘forget’ the answer and thus give clues to the user. For instance, ‘Oh I don’t remember.. there was something I had to do to X in order to get Y! I think it has something to do with division.’ Another example is to remind the user of a similar concept they have already mastered. For instance, ‘Remember what you did when you calculated fractions? Try thinking about that and it might jog your memory!’ This way, students can make more connections to concepts and see patterns. To prevent overuse and dependence, the assistant is available for 5 hints within a 1.5 hour time span. This setting can be changed by the teacher if the student needs to be challenged or needs more support. The amount of use is also logged by CAPA, a database used by teachers to keep track of student progress and information.

Feedback on Agent’s Understanding
To be sure that the agent is following instructions, it is important to find a way to test the agent’s understanding. Chin et al (2009) coined the term ‘guess and check’, where the user can quiz the agent’s understanding to ensure successful learning of a concept. In our example, the user can do as many practice questions after the concept is learned as they feel necessary. For these, the agent
displays reasoning and how they arrived at the answer. The user can then confirm or edit the rule learned. Then, they can have the TA try the questions in Matematikhuset. If the results are good, then the agent gets a diploma. If the results are not good, the agent asks for more clarification. Then, the student goes back to editing the rule. If the agent continues to have trouble, it can ask the user to seek the help of the teacher assistant.

It is important to show alignment between speakers when having a conversation (Garrod and Pickering, 2004). That entails showing signs of following along, or needing more clarifications during the conversation. For this reason, we suggest that the TA says ‘Aha, so the order matters’, or ‘Good point’, ‘Interesting!, as well as nonverbal cues to show that they are engaged and listening. Moreover, the agents would make connections to previously learned concepts to show understanding or the need for more clarification.

We debated having feedback from the system indicating the agent’s readiness to take the quiz or to hint if the concept taught was done correctly. Since the point of the TA is that the student knows and imparts knowledge to the TA, we thought we would not include such a feature. Instead, the agent receives diplomas and awards for what is done well.

**Too much of a good thing**

As important and helpful as feedback can be, we should be careful not to overwhelm the user with information. Magnus Haake remarked that when testing their program with kids, they would sometimes get annoyed by the agent asking too many questions when they just wanted to play. Andre et al (2009) also point out that ‘advanced learners get annoyed by overactive agents’. We do keep this in mind and have the option to mute the agent, and not have them necessarily react to everything.

### 2.1.3. Motivation

Any struggling student needs both emotional and social motivation. Our teachable agent aims at helping students who don’t feel confident about their math abilities, which means they generally dislike math. Their low achievement level could be due to a lack of encouragement, or a lack of self efficacy. The TA can help change that attitude and empower the child by boosting those areas. By positioning the student as teachers right from the start, it instills the confidence that they do have something to offer. Moreover, by having an agent, they can externalise the failure since it is the agent that did not do as they should, instead of blaming themselves. Further, their interest in teaching math increases because of the protégé effect and because of the framing of the task in a game. McNamara et al (2009) and Cordova & Lepper (1996) find that giving user control is intrinsically motivating as it leads to better performance and greater persistence. Cordova & Lepper (1996) conclude that “both choice manipulation and personalization manipulation may be viewed as manipulations tending to increase the self-relevance of the activity.” (Cordova & Lepper, 1996, p5). Our program allows students to choose and personalise the agent they would like to teach, as well as the concepts they would like to teach and practice.

**Motivation & Learning**

McNamara et al (2009) list several constructs that are deeply rooted in motivation, such as self-regulation, self-efficacy, interest, and engagement. Self-regulation refers to the student’s self tracking of progress or failure within the learning process. Presumably, high achieving students are keeping track of such cycles and ‘regulate key cognitive, metacognitive, motivational, social, and affective processes’ (McNamara et al, 2009, p10). This means an active attempt to seek understanding. Self-efficacy is the learner’s belief of ability in a particular domain, which could be strengthened through meaningful feedback, attainment of bonus features, and awards. Interest refers to the degree to which underlying needs or desires to develop competence or a personal
investment in a particular topic or domain. For this, we have bonus features, such as locked personalisation options, characters and games that can only be unlocked once a certain level is reached. In addition, the social network allows students to share progress and achievements to encourage each other to persevere. In other words, how the content is framed is important in driving interest, such as framing educational content in a game context. Engagement is keeping students attentive and preventing boredom, which could negatively impact learning. Especially with students who begin with low opinions of math and need to find a fun way to engage with it. For this, we look at examples of existing successful games and how they engage the user.

Motivation is important in learning contexts because it can lead students to make greater effort, seek greater challenges, and have higher achievement (Schunk et al., 2007). However, our purpose is not simply adding arbitrary points or levels. It will likely not result in any significant learning changes, nor will it improve motivation. In order to be effective, the TA will be integrated within a coherent system, and must serve a bigger purpose. Jackson et al (2009) explain this “purpose” being as simple as using points to compare performance between students (competition) or getting a proof of achievement at the end of a set of tasks, such as a diploma. This notion is incorporated in the quest game that has a bigger purpose.

Features of our program
Inherently built in the concept of a teachable agent is empowering the student and putting them in the position of an expert who imparts knowledge, or has control over the outcome. According to many, such as McNamara et al (2009) and Gärdenfors (2010), this kind of active participation is a vital component of building up intrinsic motivation. To this effect, the wording of components and the agent’s interactions is important as well. The term ‘teacher assistant’ as opposed to ‘help’, and the agent’s need for the student’s help fit within this framework. In an educational context, having clear goals to attain as well as visible progression towards them is extremely valuable. It helps drive the student on the right path. In our program, the goal is to have the agent adequately answer the questions from Matematikhuset based on what they were trained for. The protégé effect in such a condition helps the student excel. Since it is an agent, any failure on the part of the agent is externalised, which means the student will avoid blaming themselves.

In terms of a clear sense of progression, we have included several bonus features that the user can attain only when the agent has achieved a certain level of expertise. For instance, certain personalisable features, certain characters and certain games are locked until the agent is able to reach a certain level. The program would also allow the option for the teacher to clearly set which of the concepts must be mastered before attaining a certain bonus feature. Moreover, the student can have a wish list of locked items saved in their settings with a list of concepts that must be mastered in order to obtain them. Further, after each concept is successfully completed, the agent receives a diploma that is saved in the agent’s ‘trophy room’. These consequence based feedback can be used as important motivation factor.

When the social network component is added, it will give the user an opportunity to chat with friends, have joint study groups, and healthy competition between each other and also collaboration on games. The chatting engages students and allows them to show their agents off to each other. They can go in and see others’ agents accomplishments and bonus features. This can encourage them to catch up and have a sense of how they fare in comparison. Users can also have their agents participate in games over the network and compete with other agents. Users can also go head to head with their own agent. In both scenarios, they can see the agents’ reasoning. It could be interesting for students to see how others’ agents reason, as it may provide an alternative way of looking at a concept. We are wary of not making it too competitive between them. To this effect, users can have their agents collaborate in a study group to help each other teach a concept.
They can have the option to discuss strategies for their agents to learn different concepts. Moreover, they can play together in a game to reach the same shared goal.

Engagement is another important factor related to both motivation and learning. McNamara et al (2009) state when students are not engaged, they are bored or inattentive. The TA’s behaviours are vital so that students will continue engage with the TA. Therefore, we supply a range of characters that the user can pick to teach. Each character is very animated and conveys a personality type. Having a graphically rich environment also will increase the students’ motivation to engage with the TA. The user can learn more about each character through their ID when they first pick them or by looking in their personal settings. These agents can be personalised in terms of how they appear and the gadgets they own. Investing so much into their character coupled with the protégé effect will keep the student drawn in to helping their agent succeed. In addition to having an agent that the user can connect with, there will also be games in the social network and very clear goals for the students to achieve.

Since the target group are students who are normally not confident in math, it is important that progress be based on smaller, do-able steps. Luckily, Matematikhuset has already built that into their system. Each level builds one small step above the previous one. By structuring it this way, it also helps teachers locate the exact source of a problem that the student may have.

**Lasting Appeal in existing games**

With so many game out on the market today, and with an industry that keep growing, there is a huge competition among game developers to produce not only a entertaining experience, but a lasting one. Motivation to keep the gamers play a game for a long time is a key factor in game developing, and most game review sites such as IGN.com rate games with categories such as “Lasting Appeal” and “Replay factor”. To give us some inspiration, we analysed some of the most successful games and tried to identify the factors that keep gamers coming back.

One way to analyse how game developers motivate people to play for a longer period is to compare a gaming franchise that has been around for a very long time and see how the games have improved in this specific area. Super Mario has engaged gamers for many decades, so it is interesting to compare one of the first Mario games, “Super Mario Bros.” from 1985 with its follow up 25 years later, 2010: “New Super Mario Bros. Wii”.

**Super Mario Bros.**

Super Mario Bros. is a classic arcade game. The story is, as in nearly all Mario games, that Princess Toadstool, later to be named Princess Peach, is kidnapped by an evil monster Bowser. The user’s task is to defeat Bowser and rescue the Princess. The game consists of 8 worlds, and a few stages in every world. Once a stage is complete you continue to the next stage until all stages are done.

What the developers implemented to keep gamers continuing to play even after saving the princess were:

1. Excellent game-play. If the game is fun to play, you want to play more. Also, the graphics were ahead of its time.
2. Collect lives/coins and get high-score. The game has no way of showing your high score but players can keep track of how many lives and coins they have collected and compare it with other times they played.
3. Secrets can be found in almost every stage in this game and gives the player motivation to play a stage several times to uncover all the secrets.

**New Super Mario Bros Wii**
This game plays out almost exactly the same as its predecessor, but of course with improved graphics and sound. However, there are several additional features that maintain user interest:

1. More content - This game has a lot more content then the 1985 release. 4-player multiplayer, a special mode in which players can compete with each other to get the most points, and of course more stages and power ups.

2. Non-linear story. Players can now choose to go different path through the worlds. The user is in complete control of the path taken. You can choose a different path each time the game is played.

3. Profile rating. Players can save their progress into profiles. Each profile gets a rating after completing the last boss. You can improve your profile rating by finding more secrets and special coins.

4. Guides - If a player gets stuck in one stage and loses 8 lives, a green block appears showing a safe way to travel through the stage. By helping the user along, they avoid frustration and make it possible for the gamer to continue.

5. Bonus content - Players can buy short videos of special moves and how to reach secret areas with special coins collected within the world, which can only be done by obtaining all the coins in the game.

**Connection to our program**
We took some inspiration from these successful, established games by incorporating stages, bonus material, profile viewing which includes rating (trophy room), guidelines from the teacher assistant, and user control in how to proceed between levels. This means that the user can play as many times as they wish and not necessarily repeat the exact same steps. More about individual quests, or games is explained below.

**2.1.4. Social Network**
Having a social network in an educational context is extremely valuable. Its benefits are recognised by many, including Gärdenfors (2010). He says that learning is intensified when there is cooperation and co-constructive behaviour. Being able to communicate with peers, such as through chat and messaging systems can greatly help students learn and improve. It is a well developed area of pedagogy that peer learning helps students learn and can even be superior to other techniques. We find that in this context it should be an essential component of the program. Not only are they discussing ideas with their agent, but also with each other. They will also have opportunities to look at their friends’ agents’ reasoning, each others’ progress, as well as a host of other information. We hope this will encourage them to persevere, and to engage them further into the program and perhaps intensify the protégé effect.

This feature allows users to chat with friends or the teacher, have joint study groups, and healthy competition between each other as well as collaboration. The chatting engages students and allows them to show their agents off to each other. They can go in and see others’ agents accomplishments and bonus features, as well as their ID. They can also send offline messages in...
case their conversation partner is offline. Users can also have their agents participate in games over the network and compete with other agents. Users can also go head to head with their own agent. In both scenarios, they can see the agents’ reasoning. It could be interesting for students to see how others’ agents reason, as it may provide an alternative way of looking at a concept. An example of a game can be similar to ‘Who Wants to be a Millionaire’ where they can ‘ask a friend’. The games would support multiple players. More games are described below.

We are wary of not making it too competitive between them. To this effect, users can have their agents collaborate in a study group to help each other teach a concept. They can have the option to discuss strategies for their agents to learn different concepts. In this mode, the chat would support a whiteboard conference so students can draw and point to things. Once they collaborate, they work together to win a game, or even in a team against another team. If the teacher is concerned about students overusing the chat feature, there can be a time limit per day set by the teacher.

In the social network, we could incorporate ideas from successful networks, such as Facebook and Twitter. There can be updates of their friends, noting if one of them has completed a level, or obtained a certain bonus feature. That update can be ‘liked’ and be commented on. They can also post comments not necessarily related to their game, but of a more social nature. Please refer to Figure 4 for an example.

We realise that what may drive users to keep using the program is mostly the social component, but nonetheless, they will need to do math in order to succeed on to higher levels and to have updates to report. In essence, they will learn while having fun and may not even realise that they are learning! As much as we strive to keep users coming back, we are also careful about making them addicted to the program. To address this, the agent can use body language to convey being tired, needing a break or using the time stamp information to suggest for the user to do something else. Another way is to use an energy meter to show that the agent is not prepared to learn at that moment and needs some ‘rest and recuperation’ time.

2.1.4.1. Games

Why Games
Some of the reasons we included games are because they are an opportunity to take what students have learned to another level and context. In order to succeed, the players must apply their knowledge of math in the game situation. Children, especially those aged 9-11 are drawn to games, which makes it a motivating way to make sure children learn. Another benefit is that discovery-based and goal-oriented games can be very effective ways of developing team building skills (Argyriou et al 2010). Argyriou et al (2010) observe that a group must work together to strategise, develop a solution, maximise the various talents of the team members, and execute their plan in concert to succeed. Carbonaro et al (2010) outline two main problems with most educational games: poor production quality and monotonous game-play, as opposed to commercial, non-educational games. The ultimate measure of success is when children try it.

Design Principles
Wang et al (2010) outline eight characteristics of good educational games:

1. Variable instructional control: level of difficulty adjustment
2. Presence of instructional support: hints/ supplementary info/ chat for multiplayer modes

3. Necessary external support - make sure users know how to use it and how to troubleshoot

4. Inviting screen design: also ability to customise avatar (in our case agent)

5. Practice strategy: practice run that doesn’t affect score

6. Established instructional principles, such as collaborative learning, repitition, incremental learning.

7. Concept credibility: game and its purpose must be clear and not overly complicated, or the user will lose interest and not take it seriously.

8. Inspiring game concept: inspire the player to invest in the game.

Based on our description of Smart N Up, we do attend to each one of these principles.

Diah et al (2010) define desirable features from a usability point of view. They find that the program should have learnability (learning the system), efficiency (once learned, how quickly can they perform tasks), memorability (even after a break), errors (easy to recover from? severe?), and satisfaction with product.

To find the answers to these questions, we would need to have our target group play the games and interact with the program. We did an initial test with our classmates, which allowed us to make improvements to the game. For instance, making instructions and possible actions in the game clearer from the start. We must also find a way to recover from errors. We did not test with children yet because our program is far from completion. We decided it was best to wait before testing with children so we could get the feedback.

Aleven et al (2010) describe a framework for educational games assessment that also helps with the design. They outline three main parts:

1. Learning Objectives must be clear. It is important to specify prior knowledge, learning and retention goals, as well as potential transfer. This steps ensures that the game is targeting the desired skill and to check whether or not the goal was accomplished

2. MDA: Mechanics, Dynamics, Aesthetics. This criteria clarifies the available actions that are possible in the game, the graphics, level of challenge, and reaction time needed. For instance, in Math Catch, players can move sideways, hide under the horizon, and jump. It is a fast paced game, which means reaction time must be quick, especially as the levels increase in difficulty.

3. Instructional principles. Aleven et al (2010) agree with Wang et al (2010) that the game must be based on sound research on educational principles. For instance, the LLL principle states that learning wrong information can be reduced if feedback is immediate. This principle can be incorporated in the game design.

Our Game
In our game, we want to immerse the player in a virtual world with a larger purpose and a series of tasks, or games, that require various levels of expertise and knowledge in math. The game is a quest with a series of nodes, representing tasks (or games) that each require math to complete.
The tasks involved are based on list of competencies a fourth grader must know before and after completing grade 4, which is our target age group. Such skills include comfortably adding and subtracting large numbers; using tools, such as rulers and thermometers, to measure; multiplying larger numbers; and solving simple math sentences that contain a variable (James, 2010).

The assumptions we made in our design were supported by the literature. Wang et al (2010) advised that to keep a game intriguing, it must have a goal and a challenge. Our quest game has a bigger goal of rescuing a friend from bandits and each individual game has its own goal that words towards that (each explained below). Wang et al (2010) continue to say that the game must also have instructional control, or ability to adjust level, either by the user, or by the system as the player improves. The game should also be somewhat rooted in reality, where skills are needed to advance and interact with the virtual world. Our games also have the ability to adjust level. In the future, we would like the system to adjust it based on the player’s performance. Wang et al (2010) quote Angela McFarlane who believes collaboration in learning is the most important field of research in educational games. Our attempt is built on that very concept by allowing different modes of play, such as group play. There, users can use the chat feature to collaborate and devise a plan together to solve the games.

Betty’s Brain and Magnus Haake’s program both have a game show, which is a great idea because it is motivating in the sense that you must perform well in front of your peers. On the other hand, it may also be intimidating and put pressure on the student. Here, we would like to present some alternatives to game shows.

The quest game begins with a map of available paths with stations (games) along the way. Each of these stations requires math to bring the player closer to the goal of rescuing the friend. Also, at first, some of the paths are greyed out because the agent must acquire more skills before even attempting them (bonus material). This gives the player motivation to reach those levels and open up the paths to have access to more games. Please refer to Figure 5. We have implemented two of the games, while the others are still on the conceptual level. The results of each game are recorded in the hall of fame, as well as the option to put it in the social network’s newsfeed and updates. The information is also recorded in the user’s profile and in CAPA.

All of the games have several modes: user (or child), agent, agent vs user, agent in collaboration with user, user vs other players, group work (all collaborate). Not all of these modes apply to all of the games since it does not apply. When the agent is playing, the agent’s character will appear as a player. The user’s character is the one we designed. Please refer to figure 6 for an example.

The energy level of the player is required to play games in order to avoid overplaying and addiction. The games are intended as a fun escape from the drills and exercises, but should not interfere with learning and the teachable agent time. As the child plays, the energy level of the player decreases over time. Once the energy level is depleted to zero, the child must allow the agent to ‘recuperate’ and regain energy, or teach the agent more skills. During the games, they can also gain bonus time extensions to avoid interrupting a good flow.

**Individual Games**

**Math Catch**

Math Catch is a high speed game aimed at achieving fluency in adding large numbers. The target number is displayed to the left of the screen, and the player(s), must catch the appropriate equation as it falls from the sky, like rain. This game requires the user to be comfortable adding large numbers and to become as quick as possible, as is expected of 4th graders. This game
supports user, user vs agent, and user collaborating with agent. Intermingled with the falling equations are speed shoes, that allow the player to walk faster to get to the desired equation, as a bonus. The player will have 3 lives before game is over. Please refer to figure 6.

**Power Dance**
Power Dance (formerly known as Smash) is another high speed game that aims at achieving fluency in multiplication. The player dances in the direction of the box containing numbers. The aim is to find the combination of numbers to multiply to get to the target. The resulting moves look like a dance, which can be entertaining for the player. There is a time limit to achieve the goal. The user will have a few chances to get it correctly and must get 10 rounds before moving on to the next number. This means that if the goal number is 20, the player must achieve 20 using the boxes for 10 rounds before the target number is changed to 38, for instance. This is to ensure that the child gets enough practice with the goal. This game supports user, user vs agent, and user collaborating with agent. Please refer to figure 7 for an example.

**Math Attack**
Math Attack is a variant of Pac Man. Instead of only tokens, the maze is strewn with subtraction equations. The ghosts that appear each have a target number. They can only be defeated if the player has eaten the corresponding equation, otherwise, they must run the other way! For instance, if the player eats 900-600, they can defeat the ghost marked 300. This game encourages fluency in subtraction of large numbers. This game supports user, user vs agent, and user collaborating with agent. Each player will have 3 lives before game is over.

**Willy the Snake**
Willy the Snake is a variant of Snake, a popular telephone game. Willy grows in number and size as it collects digits in the playing space to get to the goal number. For instance, to get to 28, the snake must collect all the numbers in the playing space to get to 28 within the time limit. This game is best as a single player game. The player has 3 lives before game is over.

**Impossible Mission**
Impossible Mission is not like the other high speed games. Instead, it requires the player(s) to solve a puzzle in order to free the friend. The player will have a set of tools, such as a ruler, rocks, spring, rope, among others to help solve the problem and reach the friend in need. The player can also use grapes to count, or use the rope to swing to the other side. The ruler measures the distance, to calculate the number of rocks needed to fill in the gap, so that the player can use the spring to jump and reach the friend. This game is based on the competency of using tools to measure and solve problems that a 4th grader must learn. It is also a game rooted in reality, but played out in a virtual world. This game supports all the player modes. In fact, it is best if it is used in a multi-player setting where the users use the chat to discuss strategy, or speak with the agent.

**Feedback we would like to implement in the future**
- An information bar should always appear containing energy level, lives, score and time.
- A test play that doesn’t affect the player’s score to test the game
- Bonus token if you get 4 correct answers in a row, and increase in time if you get 8 in a row. The bonus tokens can go towards getting some accessories to customise the agent
• If you are in the vs player mode, there would be a split screen for each team

• An option to turn off the timer to support all levels of expertise

• At the end of a game, a summary of correct answers (yes! 4+1 = 5. ....).as well as
incorrect ones (Watch out! 3+7 =10 and not 14)

• Hall of fame for high scores for each player, and for all users.

2.2. System Implementation

Features implemented
These are the features implemented in the latest version of our running program. We also added
some tools that help develop characters and test questions.

Overall Program
• Basic GUI and interaction.

• Different characters to choose from with ID cards (figure 2). Although it is very simple at
the moment, we would eventually like to expand and allow the character to speak and be
animated based on their style. The characters would be more similar to Buzz! characters,
as in figure 1.

• Library selection screen where students can choose the level and concept, which
corresponds to Matematikhuset.

• Multi-question system.

• Basic information on the agent’s current skill level, which increases with each correct
question answered (This was debated in the conceptual plan, but was something we
wanted to see implemented to help us decide how to proceed)

• Ability to test the agent’s skill by giving him a test. The test/examination is composed of
three parts:

1. The user creates a test by choosing from a number of questions.
2. The agent writes the test.
3. The user corrects the test to ensure that the agent has fully understood or to identify a
problem area.

• Medals/diplomas for the agent if he performs well on a test.

• Sound feedback based on user’s actions

• Two of the games including feedback and power ups, as well as the quest map.

• A form of interaction using multiple choice questions.
**Interaction method**

The multiple choice questions are of two types: method and concrete examples. The interaction begins with 5-6 rounds of concrete examples, before switching to method. For example:

**I. Method:**

1. How should I start solving 582+347?
   a. add the last two digits of each number (2+7) and carry 1 over
   b. add the middle two digits (8+4) and carry 1 over
   c. add the first two digits (5+3)
   d. add the last two digits (2+7)

   Then user selects d. (If the user didn’t select d. in the first question, the wording of question 2 would not appear as in the example below).

2. So now that I’ve added the last two digits, do I..
   a. add the first two digits (5+3)
   b. add the middle two digits (8+4) and carry 1 over
   c. add the first two digits (5+3) and carry 1 over
   d. add the middle two digits (8+4)

   Agent: So that leaves us with adding the first two digits of each number, plus the 1 carried over. Is the answer 919?
   a. Yes
   b. No, let’s review

**II. Concrete example:**

1. To solve 582+347, I first..
   a. add 2+7 and carry 1 over
   b. add 8+4 and carry 1 over
   c. add 5+3
   d. add 2+7

   User selects d, then:
   2. So now that I’ve added 2+7, I should..
      a. add 5+3
      b. add 8+4 and carry 1 over
      c. add 5+3 and carry 1 over
      d. add 8+4

   The user then selects b.

   Agent: So that leaves us with adding 5+3, plus the 1 carried over. Is the answer 919?
   a. Yes
   b. No, let’s review

*Please refer to figure 3 for an example.*
Games
The two games implemented were Math Catch and Power Dance (formerly known as Smash). The quest map with some of the path greyed out as bonus material and:

In Math Catch:
- The character is animated in 4 frames for all 4 directions (walk left, right, jump or hide)
- The player can catch the falling equations
- Single and player&agent modes (each distinguished by a color)
- Sound feedback based on user actions and successes or failures
- Background music

In Power Dance:
- The player can tap the boxes to multiply numbers.
- Feedback for correct answers are a green box for each round, or a red one for incorrect answers.
- Animated background character
- Two of the direction animations
- Background music

Tools
These tools are developed with a XML-form and are stored in files that are loadable by the program.

- Question editor - An easy way to create questions or problems.
- Character editor - An easy way to create and add chose-able characters to the program.

The reason that we wanted to build this high-fi prototype in a high level programming language (C# and XML in our case), was that we wanted something that we could continue working on rather than the image of a prototype in Power-Point. It was also an opportunity to implement some of the concepts we discussed into a usable program.

Future Look
The full concept we have in mind is explained in section I and in the lo-fi prototype, but due to time and resource constraints, we could only program a simplified version of it. We felt that the basic interaction was the most important part, so the motivational and graphic details are not implemented yet. While none of the conceptual plan conflicts with the programming, some of the features cannot be implemented because they require a lot more time, such as:

- Multimodal approach - As mentioned in the interaction and teaching part, multimodal learning is very effective and caters to different kinds of learning. Instead of the drag and
drop form, we implemented drop-boxes since they are easier to implement. Moreover, other interaction details, such as options to edit and test, etc., are not included yet.

- The ability to personalise character with facial features, clothing and belongings (gadgets) is part of the motivation, but was not high on the implementation priority list.
- Graphical design and animation: While some work has been done to make the program graphically appealing, there is still more work to be done. Namely, implementing the character animation.
- The artificial intelligence behind the agent and its learning has not been implemented.
- The personal settings are also important, but were not yet implemented.
- The social network has not been implemented yet, but two of the games have been, as well as the quest map.
- Tools, such as the teacher assistant and other similar features were not added yet.
- The connection to Matematikhuset was also not made because we would need access to their source code, but it does have a similar set up in the teaching and testing part.

3. Issues that Remain

As we discussed the conceptual plan and implemented, we came across several issues that have still not been settled. The main stumbling blocks that we faced with the conceptual model was how to interact with the agent to allow for the different learning styles and ways of explaining a concept, how to actually teach the agent (visually, narratively, using examples, etc.) and whether or not the students should get an indication of incorrectly teaching a concept to the agent or allowing them to make mistakes. In the second part of the course, we were challenged by the scoring method, the overarching goal of the quest game, as well as the base knowledge of the agent.

There are a few ways of interacting with agents that have been implemented, such as Betty’s Brain, Billy’s Brain, Milly’s World, among others. In our program, none of these methods applied because the exercises that our agent would perform are arithmetic from Matematikhuset. Here, it is not as useful to have a concept map, a graph, or explaining using objects in Milly’s vicinity. Moreover, it was important to balance between the mode of interaction and the child’s age and level. It shouldn’t be overwhelming for the child to interact with the agent from the beginning. Our solution was to opt for multiple choice, since it was simple enough to implement and was simple enough to use. We will just be careful about the wording of the question to allow for more flexibility. In addition, the agent could learn from examples, then put the concept into words based on a pattern from the examples. For instance, ‘So do you always take X and do Y?’ Then, the child confirms or edits.

Actually teaching the agent was difficult to implement. We began by discussing visual input as a means of explaining fractions. While it is helpful for the first simple examples, it may become too complicated for larger numbers and for more advanced concepts. A possible way to transition is
to have the agent put the concept into words, such as in the previous example, so it can be applied to larger numbers.

We debated having the system indicate to the user if an error has been made in the way it taught the agent. In the end, we decided to not show the student because we must assume that the TA does not have any previous knowledge of the concept and cannot possibly know that the student is not teaching it correctly. It is difficult to isolate the role of the agent in our example. Not only is it difficult to show the progress of the agent, but also to synchronise progress to performance on exercises and games in terms of ability and speed.

Another point of discussion is what the base knowledge of the agent should begin with. We decided to supply the agent with very basic math operations on smaller numbers. We would like to maintain the role of the agent as a learner, but do not want to make the task tedious for the users by explaining how to add, subtract, multiply and divide basic numbers. We were unable to find relevant literature and studies to investigate this.

As for the game, we were not sure how to handle scoring in a way that is still motivating, but doesn’t keep the problem of disparities between the levels of students, thus discouraging them. During the game testing, we received some suggestions to think about. For instance, there would not be a personal score, but only a school score. Then schools would compete against each other. While this may eliminate competition between classmates, it may not give the child enough feedback on personal performance, which is vital to learning. Another suggestion is to erase scores on a weekly basis. This may be an option, but it may also deter students from working towards better performance in the long run. Another way is to give each user only one chance in the hall of fame. That means they can only outdo their own previous score, and not flush others out of their chance. This option still does not resolve overly competitive behaviour, or reinforcement of the different levels of performance. A last option is to only have the agent’s score be displayed. Again, this may not provide the child with enough feedback on their own performance. The solution agreed on for the moment is to have only one spot in the hall of fame and a personal best score. Their progress over time will be recorded in their profile and CAPA.

The second issue that is still under debate with the games, is what the overarching purpose of the game should be. At the moment, it is to save a friend from being taken away by bandits. This storyline may work for most existing commercial games, but we would prefer to have it more related to math to fully immerse the child. Still, it must be motivating enough to have the child want to resolve it. Along the same lines, is whether the games should be more cohesive and interrelated, or if they should be as they are now, separate quests requiring different levels of expertise and skills. For instance, Math Catch requires different skills and approach to Impossible Mission. We need more time to know if the games should be more similar to each other, or if it is better to combine high speed games, such as Math Catch and Power Dance with games that require reasoning, such as Impossible Mission.

4. Conclusion

In our conceptual plan, we took into account tried and tested theories, looked to successful games for inspiration and used our experience as teachers. We infused the program with motivation, ways of interaction, meaningful feedback, and educational components. The most determining factor, however, is how the children react to the game since they are the users.

This field is not established yet, so the problems we have come across are challenging to all of
those doing research with teachable agents, such as an effective way to interact with the agent and making the experience as natural as possible. We have noticed that most math games do not make efforts to incorporate ideas from existing, successful games. Perhaps the focus on math skills should be moved to a bigger goal or purpose that requires math to advance along the levels. The child should enjoy solving the task without realising they are learning at the same time!

There are so many possibilities for a teachable agent beyond math training. It is also possible to expand on the different roles of the teachable agent and the different purposes it serves. For instance, it can be a teacher, a guide, a critic, as well as a learner. A teachable agent can even be developed to accommodate a virtual class or pretend play.

5. Future Work

If we had more time and resources, there would be quite a few features that we would have liked to work on. The program described here is simply a basis for many possibilities to work on. For instance, connecting the program and the user information to CAPA, the teacher’s database to help them keep track of their students’ progress. It would help them identify the areas that require more attention and help them plan their teaching. It can also help them create worksheets and practice for their students. In addition, we would develop the teacher’s settings to help tailor the program for her students individually or on a group level. Moreover, we would have liked to look into facial expression imitation and from there allow the animated agent to react appropriately to further engage students and make the experience seem more real. An interesting area to delve into would be the realism of the character and how that affects learning. It would also be helpful to test the program and see how children react to it and what changes we can incorporate. We would also like to think more about the bigger goal of this program to allow the users to play a game using math skills to succeed. Moreover, it is worth exploring more realistic examples of math so that children can relate the concepts they learn to their own experiences, which intensifies learning. Additionally, integrating social interaction with the agent can be explored. There are so many areas within TA that can be further discussed and developed. We also suggest working on the remaining issues listed in section 3 or implementing more of the conceptual design.

6. Our Team

Our team worked loosely based on SCRUM methodology. We relied a great deal on Googledocs for a summary of our meetings, a list of tasks for each person, and a daily log accounting for what we did, problems we had encountered and what we were going to continue doing. Although we set it up, we didn’t use ‘weekly sprint goals’ to guide our weeks. Instead, we met twice a week. On Mondays, we discussed the portion of the program we were going to work on and brainstormed together the ideas to work on for the concept. The programming team would begin implementing the concepts that were agreed upon. On Wednesdays, we met to address any issues we came across and to keep working on the weekly goal. By Friday, we posted up the document for the week, while the programmers demoed their work, and began thinking about the following week’s goal. It was good to break the project into parts and focus on each component on a weekly basis. In the second part of the course, we felt the time constraint. We were also more focused on the part of the program we were working on: games. We split up into programmers and conceptual planners. The conceptual group mostly looked up the theory, background and designed the program, while the programmers mostly implemented the game and designed the characters used. We also looked at interaction theories and updated the multiple choice questions.
Figures

Figure 1: Buzz Characters taken from http://ps3.ign.com/dor/objects/958423/buzz-quiz-tv/images/buzz-quiz-tv-20080907050307825.html

Figure 2: Characters to choose from with ID card preview
Figure 3: Teaching the Agent (first step)

Figure 4: Social Network with newsfeeds
**Figure 5:** Quest game map

**Figure 6:** Math Catch (agent & player mode)

**Figure 7:** Power Dance
References


