International Journal of Multimedia and Ubiquitous Engineering
Vol.9, No.5 (2014), pp.121-130
http://dx.doi.org/10.14257/ijmue.2014.9.5.11

Virtual Farmers Training: Realistic Simulation with Amusements using Historic Simulation and Game Storyline

Hwan-Soo Yoo[1)] and Seong-Whan Kim[2)]

School of Computer Science, University of Seoul,
13 iripdae-gil, Dongdaemun-gu, Seoul 130-743, Republic of Korea
[1)]raryu7@gmail.com, [2)]Corresponding Author: swkim7@uos.ac.kr

Abstract

In this paper, we present realistic farming trainer for experiencing crop cultivation and livestock breeding simulation. Farming simulation utilizing game experience has been presented by various formats but current researches have not presented balanced farming experience and amusement to users. We achieved both education purpose and entertainment feature by realistic farming trainer using historic simulation and gaming elements. To achieve realistic educational purpose, we implemented realistic farming simulation using historic simulation, which covers last ten years of weather, disease, and other factors. Also, we use storyline with quest design to give amusement to users. Technical details include (1) Crop simulator, (2) Livestock simulator, (3) Weather simulator, (4) Disease and insect simulator, (5) Fertilizer simulator and (6) Business simulator. Experimenting with sample user groups, they can easily learn how to farming and livestock management with amusements.

Keywords: *Farming, livestock, crop, agriculture simulation, serious game*

1. Introduction

Farming simulation and game experience has been presented by various formats. On-line agriculture simulation game provides textual information with photo images on farm working, and on-line agriculture simulator has been focused on mass simulation of farming and breeding. However, users cannot experience what is really farming and breeding by non-realistic text and photo images in such simulators and image based games. In this paper, we implement a 3D virtual collaborative space for agriculture training using multi-user on-line game framework. To make it more realistic, we design and implement a simulation engine, which supports (1) L-system [1] generator with smooth interpolation scheme for plant model, (2) historic environment simulation for weather, disease, and insect simulation, and economic changes. The paper is organized as follows: In Section 2, we review previous research works. Section 3 analyzes what is required to achieve agricultural education with entertainment purpose, focusing on what is necessary for realistic simulation of farming. We introduce (1) Crop simulator, (2) Livestock simulator, (3) Weather simulator, (4) Disease and insect simulator, (5) Fertilizer simulator and (6) Business simulator. In Section 4, we show the client and server system architecture and implementation issues. We conclude in Section 5.

ISSN: 1975-0080 IJMUE

2. Related Works

Several approaches have been proposed to experience farming and livestock breeding. We enhanced previous research [1-2] by add various simulators and DSL. I-Farm [3] educates users in growing crops, breeding livestock, and biomass production It provides nutrient balance, P-index, labor requirements at the farm scale and gives economic impacts to users and import/export of livestock and biomass from the farm. But it is for mass farm's economic simulation and not adequate for who wants learn farming itself. Advances in farming systems analysis and intervention [4] recognize two key components of farming systems, namely the biophysical ' Production System' of crops, pastures, animals, soil and climate, together with certain physical inputs and outputs, and the 'Management System', made up of people, values, goals, knowledge, resources, monitoring opportunities, and decision making. It provides solid biophysical performance of production system but not provides real farming simulation. Simagri [5] provides multi-user web based on-line capability. Users can select specified region (France, Belgium, Canada and America) to play game and learn farming. But text and image lacks effect of user's learning motivation and learning rate. AgriVillage [6] is an interactive game that places the user in a virtual environment where the player has the responsibility of operating a farm that produces food for a small village. The ideas of game are to make the player thinking more carefully about the impact of the actions. The game was tested in a pilot study with few subjects in Portugal and an experiment with more numbers of subjects in Thailand. It mainly focuses on food security and food safety not farming. CropSyst [7] suggests cropping system model to serve as an analytical tool to study the effect of climate, soils, and management on cropping systems productivity and the environment. American farmer [8] and Farming simulator [9] provides 3D based farming game. Users can grow crops and breed livestock by hire employees, build cattle shed and buy farm utilities. But its absence of learning material and lacks of reality, users cannot learn agriculture knowledge from it. Figure 1(a) [3] is start screen of I-FARM in which user can simulate mass agriculture. Figure 1(b) is a snapshot of driving tractor in American farmer that carry crops, Figure 1(c) is a snapshot of investigating the status of pig such as condition of growth and injection status, and Figure 1(d) shows a farm with selecting worked fields.

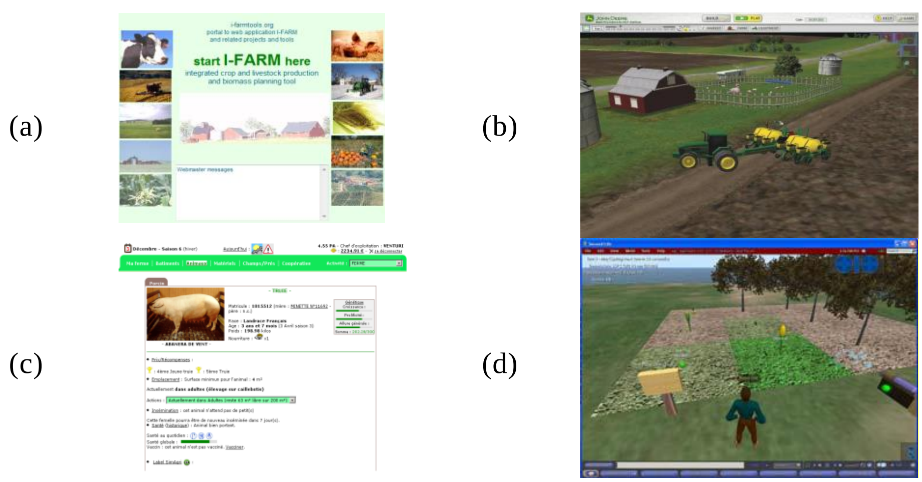


Figure 1. (a) I-Farm, (b) American Farmer, (c) Simagri, and (d) AgriVillage

3. Realistic Farming Trainer

Previous research works do not provide realistic farming experience to users. In this paper, we implement a 3D virtual collaborative space for agriculture training using multi-user on-line game framework. By using Realistic farming trainer users can get realistic farming experience and share it with other users to share knowledge. In our system, we focus on storyline design and realistic farming simulation. It gives a user both sense of reality and fun. Figure 2 depicts pepper's user experience scenario according to storyline design. In this scenario, user firstly cultivate farm, install pillar, buy fertilizer, mulching, harvest and get money by selling pepper.

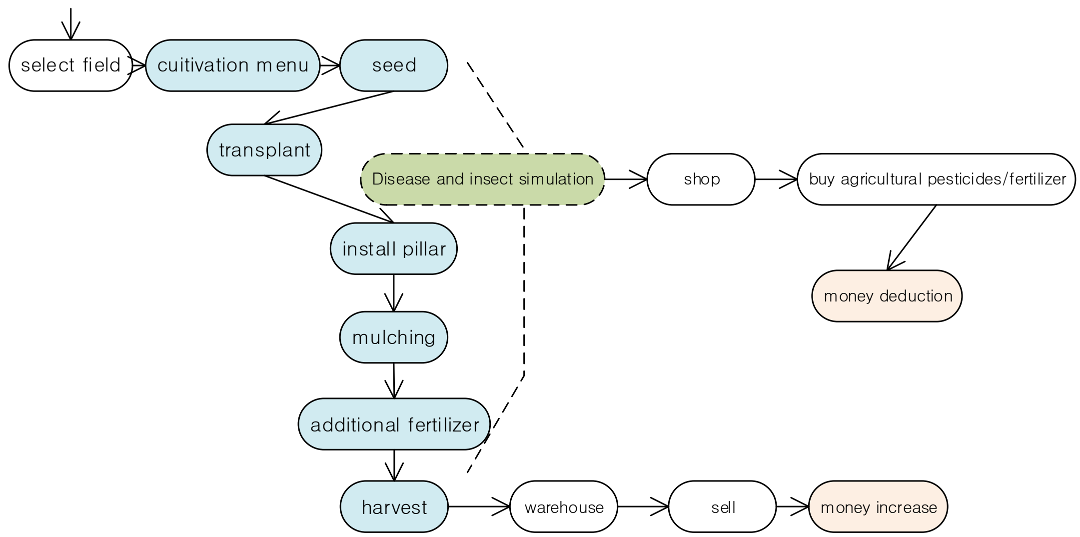


Figure 2. User Experience Scenario According to Storyline Design

We use storyline design tool to add quest and story seamlessly into game. The design of storyline mainly uses DSL[10] script generator that integrates map tool, character tool, and UI design generation tool. DSLs are languages that define the jargon of a particular class of problem domains or set of domain aspects. Executable DSLs hide software implementation [11]. DSL is referred to as either domain specific language or just simply domain language but is often referred to as domain specific language by academic community. Terminology is in contrast to general languages such as C or Java language and general modelling language such as UML. It is a limited language that is concentrated to specific problem domains by appropriate abstraction and notations. It is a problem-centric language[12] built for one particular task in mind. Development teams use same vocabulary to represent program module. For example, if there is a requirement for bond trading module within problem domain, same vocabulary is used during code writing as Figure 3 [13, p. 8]

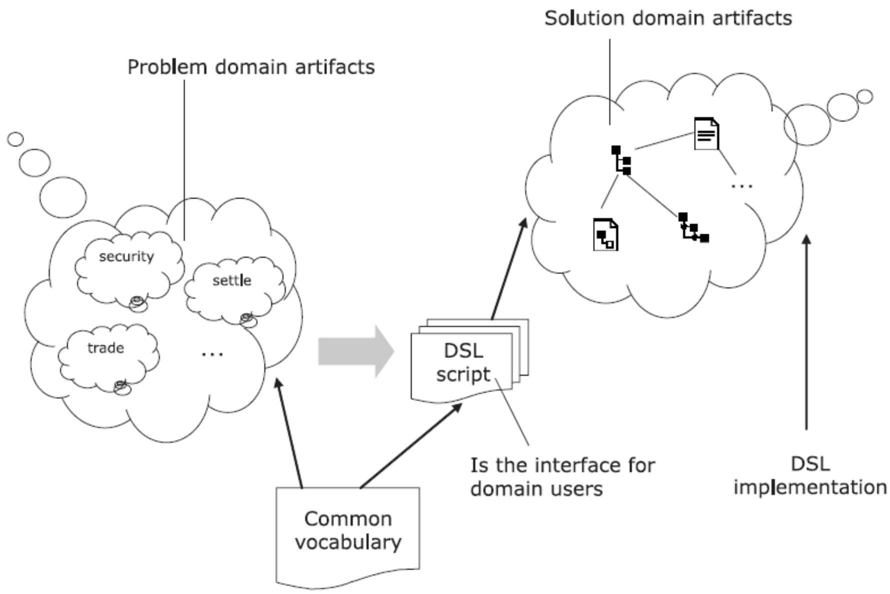


Figure 3. The Problem Domain and the Solution Domain need to Share a Common Vocabulary for Ease of Communication

Below is the partial source code of generated script. It is user's very first quest data. It contains climate data, the status of soil, user's winning condition of game. Line 1 is for selecting appropriate BGM by seasons, lines 2~5 is for climate data, lines 8~14 is for the status of soil, and lines 15~19 is for winning condition.

```
...
1.  if season is fall then farm BGM is 'fourseasons/fall.mp3'
2.  climate:
3.    temp is 20
4.    weather is rainy
5.  end
6.  row1 has potato
7.  row2 has strawberry
8.  row1 status:
9.     humidity is 60
10.    PH is -1.2
11.    Nitrogen is 5.5
12.    Calcium is 6.1
13.    Potash is 7.3
14. end
15. winning condition:
16.    potato is 80
17.    strawberry is 20
18.    timeout is 30m
19. end
...
```

To make it more realistic, our Simulator system includes simulation engine as shown in Table 1. Examples of simulator's data are illustrated in Figure 7. Figure 3(a) is an example of fertilizer simulator data. It simulates disease rate, harvest rate of potato upon soil's nitrogen, calcium and potash. Figure 3(b) is an example of disease and insect simulator data, which shows the rate of disease of botrytis calculated according to humidity, temp and seasons. We also use business simulator, in which gamers can buy and sell their products to learn how to trade.

3.1. Crop Simulator

It simulates six most popular crops (carrot, potato, pepper, strawberry, cabbage, melon, pumpkin and sweet potato). We simulate installing pillar, sowing seed, fertilizing, thinning out, weeding, tie-up crops and harvesting process based on crop database. Figure 4 depicts the process of pepper simulation.

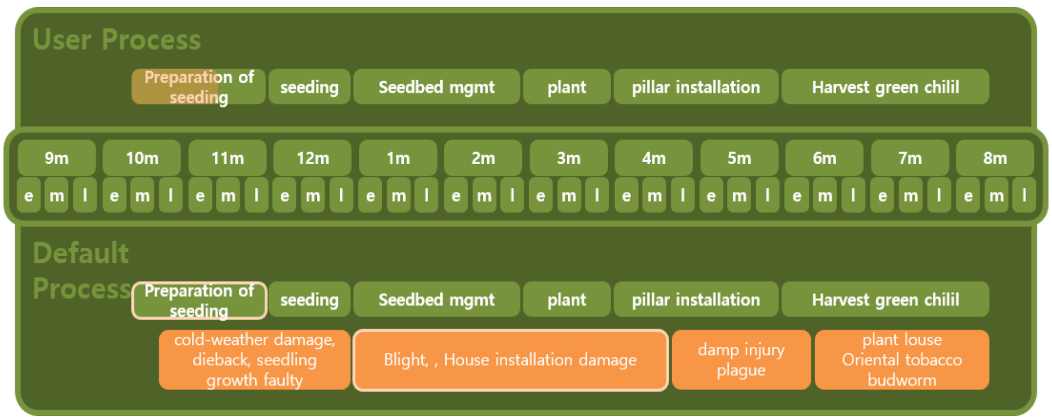


Figure 4. Process of Pepper Simulation

3.2. Livestock Simulator

It simulates three livestock (cow, pig and chicken) based on livestock database. Livestock can be vaccinated to prevent disease and user can clean animal husbandry for keeping livestock clean. Figure 5 depicts breeding process and simulated schedule of pork pig.

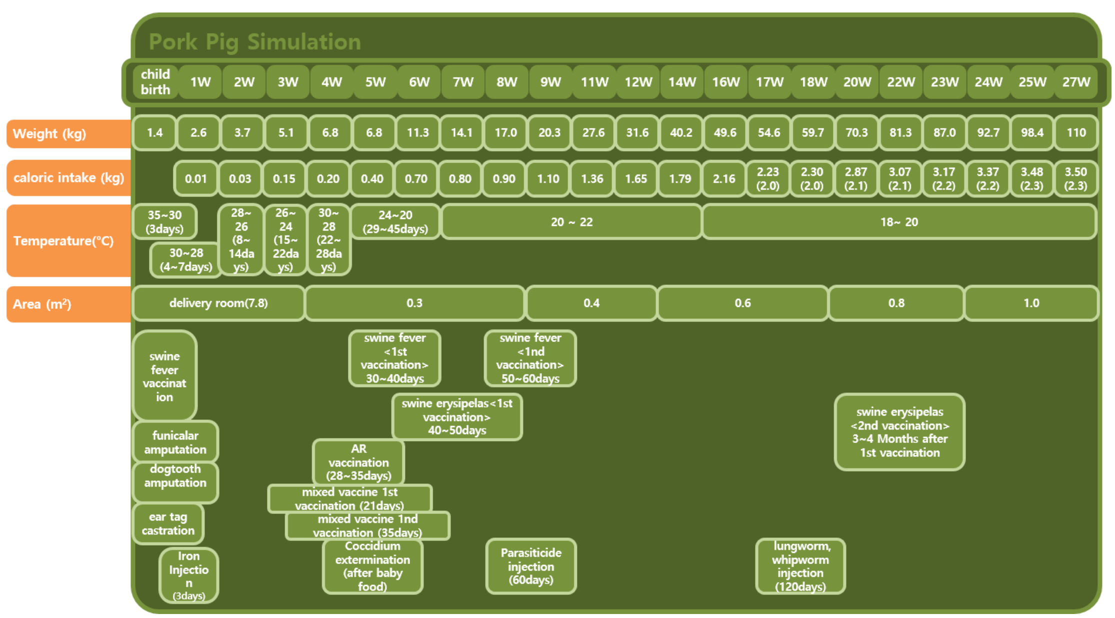


Figure 5. Pork Pig Breeding Process and Simulated Schedule

3.3. Weather Simulator

It's based on the user selection on the specific area in South Korea, we simulate temperature and precipitation effect using last ten years of historic weather data for the area.

3.4. Disease and Insect Simulator

We define an activation condition for disease and insect occurrence, and once met, we simulate dieback, downy mildew, plague, anthracnose, tobacco budworm and click beetle based on disease and insects database. User can buy insecticide and spray it to cure crops. Figure 6 is a strawberry's example data of disease and insect simulator. For each crop, we simulate dieback, downy mildew, plague, anthracnose, tobacco budworm and click beetle based on disease and insects database if the occurrence condition of disease and insects is met. At this time of occurrence of disease and insects, user can buy insecticide and spray it to cure crops.

3.5. Fertilizer Simulator

It simulates nitrogen, phosphates and potash fertilizer. User can buy and shot fertilizer.

	temp		humidity		temp		humidity			temp		humidity		temp		humidity	
2000									2001								
	Apr		Apr		Mar		Mar			Apr		Apr		Mar		Mar	
	10	normal	25	normal	14.5	normal	68.1	normal		6.7	normal	58.5	normal	16.3	Botrytis	52.1	normal
	11.9	normal	26.6	normal	13.5	normal	63.1	normal		9.5	normal	63.8	normal	15.8	normal	62.1	normal
	12.5	normal	42.4	normal	15	normal	58.1	normal		9.2	normal	49.9	normal	16.4	Botrytis	61.6	normal
	13.2	normal	46.1	normal	16.9	Botrytis	48.3	normal		10.2	normal	39.3	normal	16.8	Botrytis	67.5	normal
	9.3	normal	42	normal	18.6	Botrytis	37.9	normal		10.7	normal	47.1	normal	17.4	Botrytis	65.9	normal
	7.6	normal	42.4	normal	14.7	normal	70.8	Botrytis		12.4	normal	58.1	normal	18.9	Botrytis	66.9	normal
	8.9	normal	31	normal	14.3	normal	43	normal		14.9	normal	68.6	normal	18.3	Botrytis	92.4	Botrytis
	8.8	normal	31.8	normal	16.6	Botrytis	33.5	normal		16.9	Botrytis	57	normal	20.8	Botrytis	67.8	normal
	11.9	normal	44.1	normal	18.6	Botrytis	40.1	normal		20.5	Botrytis	43.3	normal	15.5	normal	66.5	normal
	6.5	normal	52.3	normal	14.8	normal	74	Botrytis		19	Botrytis	46.1	normal	16.5	Botrytis	52	normal
	8.6	normal	49.9	normal	13.1	normal	76	Botrytis		8.5	normal	82.4	Botrytis	16.5	Botrytis	54.1	normal
	12.7	normal	47.3	normal	14.3	normal	72.6	Botrytis		9.3	normal	55.9	normal	18	Botrytis	50.4	normal
	14.4	normal	64.8	normal	15.3	normal	70.3	Botrytis		12.5	normal	46.6	normal	18.8	Botrytis	47.9	normal
	12.4	normal	45	normal	17.1	Botrytis	61.5	normal		10.9	normal	54.4	normal	20.7	Botrytis	36.8	normal
	12.4	normal	42.4	normal	16.9	Botrytis	64.5	normal		14.1	normal	51.6	normal	18.3	Botrytis	65.5	normal
	11.6	normal	46.9	normal	16.3	Botrytis	71.6	Botrytis		15.6	normal	44.5	normal	18.6	Botrytis	47.3	normal
	13.6	normal	52.6	normal	16.8	Botrytis	72.5	Botrytis		15.7	normal	36.6	normal	21.1	Botrytis	45.3	normal
	14.8	normal	44.1	normal	17.2	Botrytis	75.3	Botrytis		17.1	Botrytis	38.1	normal	21.9	Botrytis	61.9	normal
	11.1	normal	90	Botrytis	19.1	Botrytis	67.9	normal		14.7	normal	58.5	normal	22.7	normal	60.5	normal
	14.8	normal	68.6	normal	15.3	normal	79	Botrytis		14.5	normal	56.8	normal	23.6	normal	60.4	normal
	13.1	normal	82.1	Botrytis	17.4	Botrytis	65.6	normal		14.1	normal	38.1	normal	23.5	normal	61.1	normal
	11.1	normal	65.4	normal	20	Botrytis	65	normal		13.8	normal	41.5	normal	17.7	Botrytis	82.3	Botrytis
	11.6	normal	44.9	normal	21.6	Botrytis	62.8	normal		14.8	normal	46.1	normal	20.2	Botrytis	68.6	normal
	12.5	normal	39.6	normal	22.5	normal	59.4	normal		14.3	normal	44.6	normal	22	Botrytis	63.8	normal
	15.1	normal	41.6	normal	22.8	normal	62.8	normal		12.4	normal	41.9	normal	22.7	normal	63.6	normal
	12	normal	78.5	Botrytis	20.2	Botrytis	72.1	Botrytis		13.6	normal	38.6	normal	20.3	Botrytis	75.5	Botrytis
	11.8	normal	69.9	normal	17.5	Botrytis	92.1	Botrytis		15.2	normal	37.8	normal	21.6	Botrytis	53.8	normal
	13.5	normal	58.1	normal	17.7	Botrytis	70.9	Botrytis		15.8	normal	48.3	normal	21.3	Botrytis	44.4	normal
	15.3	normal	49.5	normal	19.9	Botrytis	57.5	normal		16.7	Botrytis	75.6	Botrytis	23.7	normal	39.3	normal
	14.9	normal	54.9	normal	16.9	Botrytis	73.3	Botrytis		18.2	Botrytis	57.1	normal	19.9	Botrytis	67.5	normal
		normal		normal	19	Botrytis	64.1	normal			normal		normal	20.1	Botrytis	48.3	normal

Figure 6. Strawberry's Example Data of Disease and Insect Simulator

3.5. Business Simulator

It simulates user's buying seed and breed, user's selling crops and adult livestock, user's storing harvested crops for selling it at proper price. We use wholesale crop price and livestock price of Korea from 2010~2013.

Potato Nitrogen	Disease rate(%)	pH	Harvest Rate(%)
0 ~ 5.4	15	(-) 1.2	70
5.5 ~ 6	0	0	95
6.1 ~ 7	15	(-) 1.2	70
7 ~ 10	20	(-) 2.0	50
Potato Calcium	**Disease rate(%)**	**pH**	**Harvest Rate(%)**
0 ~ 5.4	15	(-) 1.2	70
5.5 ~ 6	0	0	95
6.1 ~ 7	15	(-) 1.2	70
7 ~ 10	20	(-) 2.0	50
Potato Potash	**Disease rate(%)**	**pH**	**Harvest Rate(%)**
0 ~ 6.6	15	(-) 1.2	70
6.7 ~ 7.2	0	0	95
7.3 ~ 8	15	(-) 1.2	70
8.1 ~ 10	20	(-) 2.0	50

(a)

(b)

Figure 7. (a) An Example Data of Fertilizer Simulator, (b) An Example Snapshot of Disease and Insect Simulator

4. Implementation

We use our 3D game engine platform (Client and Server engine) to implement Simulator, and add storyline design tool to create agricultural environments as shown in Table 2.

4.1. Storyline design tool

It comprise of farm map tool, characterization tool (for user and NPC animation), automatic UI design generation tool, Script generator, and resource packing tool.

4.2. Client engine

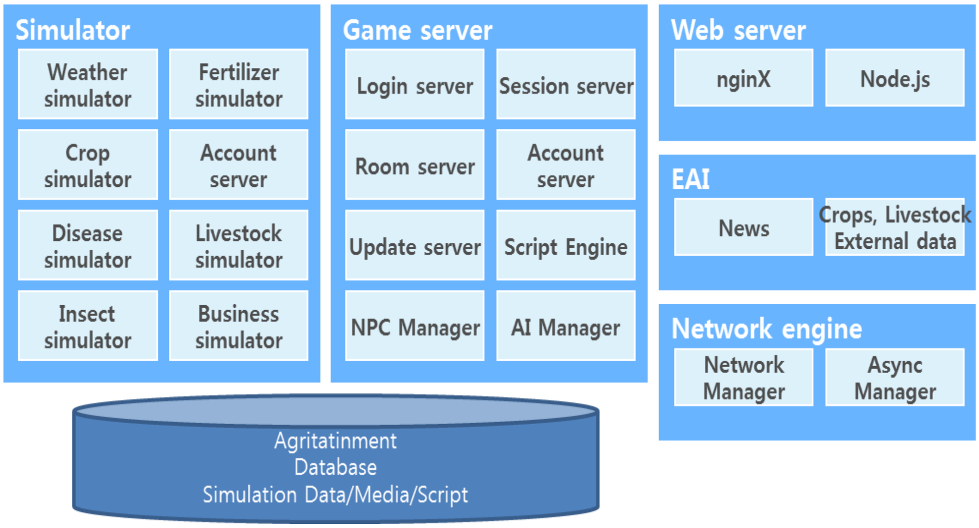
It comprise of (a) OpenGL based 3D engine, (b) OpenAL/DirectX based Sound engine, (c) animation and character module, (d) particle and texture module network based game engine, (e) GUI interface, (f) network engine, and (g) quest and tutorial engine.

4.3. Server engine

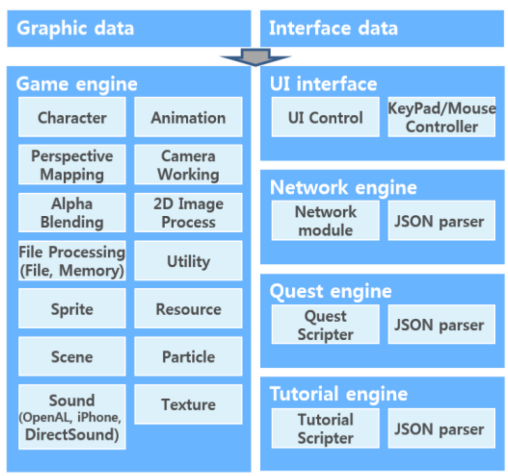
Server and SNS engine comprises of NoSQL [14] MongoDB [15] database and multi-user server module, JSON [16] parser (we implement JSON based agriculture simulators), RESTFul [17] network module for HTTP and WebSocket [18] for realtime communication, search module, and user management module including visiting users' farm management.

Figure 8(a) shows server architecture, which includes game server, simulators, web server, EAI for getting external data, network engine, and database. Simulator game server incudes (1) login and account server for user subscription, (2) login room and session server for participation management, (3) script engine for automation, (4) NPC manager, and (5) AI manager to control game flow. Web server receives client's HTTP and JSON request data. Simulator heavily use WebSocket and HTTP to communicate client and server. We use NGINX [19] as web server and Node.js [20] as application server. Client engine is divided into four parts as shown in Figure 8(b). Game engine includes animation, camera working, alpha blending, sprite, scene rendering engine, sound engine, utility, and game resource. Game engine of Simulator is based on OpenGL ES to easily be ported to smart devices. Network engine uses HTTP, and

WebSocket, and it handles game's synchronous and asynchronous data. Tutorial engine guides and teaches novice user at early stage of game. Although Simulator provides realistic farming experience to users, quest engine controls flow of game, thereby it can provide fun to user.



(a)



(b)

Figure 8. Simulator Architecture: (a) Server and (b) Client Architecture

Various features of Simulator snapshot are illustrated in Figure 5(a). User is sowing seed and the current weather is shown at upper right (a), user spray young strawberry with agrichemicals (b), empty pigpen (c), and shows land status (d).

(a)

(b)

(c)

(d)

Figure 9. (a) Crop Simulator with Sowing Seed, (b) Crop Simulator with Disease and Insect Simulator, (c) Pigpen (d) Land Status (Water, Humidity, PH)

5. Summary and Conclusion

We propose a simulator framework, where users can learn how to cultivate plants and to breed livestock. To make an agricultural training joyful, we implement 3D collaborative space for training agricultural experience, which transforms monotonous training experience into realistic experience. We tested our system with sample group who have strong interest in farming, and the result show that most users can learn the basic agricultural knowledge without fatigue or give-up. This research will be used in K-12 school to teach agriculture. Future research includes making DSL which cover all range of game to get knowledge of domain expert and to simulate cultivating, feeding, weather simulation and insect simulation elaborately. SCORM[21] type research is will be conducted to interface internal environment of simulation with external education contents.

References

[1] Y. Hwan-Soo and K. Seong-Whan, "Agritainment: 3D collaborative space for training agricultural experience with entertainment elements," in ACM SIGGRAPH 2010, 2010, p. 46.

[2] Y. Hwan-Soo and K. Seong-Whan, "22. Agritainment 3D Collaborative Space for Training.pdf," in Agritainment: 3D Collaborative Space for Training Agricultural Experience with Entertainment Elements, Jeju, 2013, pp. 123–130.

[3] E. N. van Ouwerkerk, D. E. James, T. L. Richard, and M. Liebman, "A multi-model approach for sustainable agriculture in the US corn belt," in ASAE Annual International Meeting, 2003, pp. 387–393.

[4] B. A. Keating and R. L. McCown, "Advances in farming systems analysis and intervention," Agric. Syst., vol. 70, no. 2, pp. 555–579, 2001.

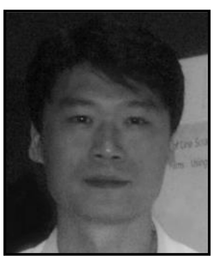
[5] L. Christophe, "Simagri," Simagri, 2010. [Online]. Available: http://www.simagri.com/. [Accessed: 26-Oct-2013].

[6] P. Yongyuth, R. Prada, A. Nakasone, A. Kawtrakul, and H. Prendinger, "AgriVillage: 3D multi-language internet game for fostering agriculture environmental awareness," in Proceedings of the International Conference on Management of Emergent Digital EcoSystems, 2010, pp. 145–152.

[7] C. O. Stöckle, M. Donatelli, and R. Nelson, "CropSyst, a cropping systems simulation model," Eur. J. Agron., vol. 18, no. 3, pp. 289–307, 2003.

[8] "Gabriel Interactive - Portfolio Gallery," American farmer. [Online]. Available: http://www.gabrielinteractive.com/portfolio/?JD_AmFarm. [Accessed: 26-Oct-2013].
[9] "Farming Simulator," Farming Simulator, 2000. [Online]. Available: http://www.farming-simulator.com/. [Accessed: 27-Oct-2013].
[10] M. Fowler, Domain-specific languages. Upper Saddle River, NJ: Addison-Wesley, 2011.
[11] F. P. Stappers, S. Weber, M. A. Reniers, S. Andova, and I. Nagy, "Formalizing a domain specific language using SOS: an industrial case study," in Software Language Engineering, Springer, 2012, pp. 223–242.
[12] B. Langlois, C.-E. Jitia, and E. Jouenne, "DSL classification," in OOPSLA 7th Workshop on Domain Specific Modeling, 2007.
[13] D. Ghosh, DSLs in action. Greenwich, Conn.: Manning, 2011.
[14] N. Leavitt, "Will NoSQL databases live up to their promise?," Computer, vol. 43, no. 2, pp. 12–14, 2010.
[15] K. Chodorow and M. Dirolf, MongoDB: the definitive guide. Beijing; Farnham: O'Reilly, 2010.
[16] C. Douglas, "JSON: The Fat-Free Alternative to XML," 2006. [Online]. Available: http://www.json.org/xml.html. [Accessed: 12-Sep-2013].
[17] J. Webber, S. Parastatidis, and I. Robinson, REST in practice. Farnham; Sebastopol, Calif.: O'Reilly, 2010.
[18] I. Fette and A. Melnikov, "The WebSocket Protocol," 2011. [Online]. Available: http://tools.ietf.org/html/draft-ietf-hybi-thewebsocketprotocol-17. [Accessed: 04-Sep-2013].
[19] S. Igor, "nginx," nginx, 2002. [Online]. Available: www.nginx.org. [Accessed: 28-Oct-2013].
[20] S. Tilkov and S. Vinoski, "Node. js: Using JavaScript to build high-performance network programs," Internet Comput. IEEE, vol. 14, no. 6, pp. 80–83, 2010.
[21] J. Poltrack, N. Hruska, A. Johnson, and J. Haag, "The Next Generation of SCORM: Innovation for the Global Force," in The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC), 2012, vol. 2012.

Authors



Hwan-Soo Yoo, he is a Ph.D. candidate in computer science at the University of Seoul. From 1998 to 2008, he worked at Telecommunication software Development Company on Internet browser, SMSC (Short message system center) server, LBSC (Location-based system center) server and mobile game platform. His current research focuses on web-based multimedia authoring system, multimedia system development using domain specific language.

Seong-Whan Kim, he received his Ph.D. degree in EECS from KAIST, in 1999. He has worked for IMT-2000 infrastructure division as a software team leader in LG Electronics from 1997 to 2000, and involved in wireless LAN infrastructure design team as a senior software engineer in Cisco Systems from 2000 to 2001. He is currently a full professor at the school of computer science, University of Seoul, Seoul, Korea. His research interests include mobile communication systems, mathematical modeling of human visual system, and its application to multimedia.

