

УДК 004.946

## ВИЗУАЛЬНЫЙ РЕДАКТОР СЦЕНАРИЕВ ДЛЯ ВИРТУАЛЬНЫХ ЛАБОРАТОРИЙ

В.Д. Абрамов<sup>1</sup>, М.М. Абрамский<sup>2</sup>, В.В. Кугуракова<sup>3</sup>, Н.Р. Манахов<sup>4</sup>,  
А.М. Маславиев<sup>5</sup>

*Высшая школа информационных технологий и информационных систем  
Казанского (Приволжского) федерального университета*

<sup>1</sup>ivitazour@gmail.com, <sup>2</sup>ma@it.kpfu.ru, <sup>3</sup>vlada.kugurakova@gmail.com,  
<sup>4</sup>nadirmanakhov@gmail.com, <sup>5</sup>erasumus@mail.ru

### **Аннотация**

Представлена реализация визуального редактора сценариев, позволяющего проектировать серии экспериментов для виртуальных образовательных лабораторий. Описана связь приведенного подхода с возможностью явного управления вовлеченностью студентов и методами ее оценки.

**Ключевые слова:** виртуальные лаборатории, виртуальные симуляции, node-based editor, графический редактор сценариев, оценка вовлеченности

### **ВВЕДЕНИЕ**

В последние годы виртуальные лаборатории перестали считаться принципиально новой и редко используемой образовательной технологией и рассматриваются сегодня как важное и необходимое средство подготовки и обучения специалистов [1–4], так как позволяют добиться увеличения вовлеченности обучающихся и, как следствие, повышения подготовленности, улучшения визуального восприятия технического процесса, сокращения расходов организации, масштабируемости процесса обучения, беспристрастной оценки процесса обучения, обеспечения коллективной учебно-методической работы [5]. Наиболее результативным способом освоения лабораторного практикума стало комбинирование традиционного лабораторного практикума студентов естественнонаучных специальностей с окружением, создаваемым виртуальными лабораториями

[6]. В [7] подробно описан один из сценариев, реализованных в рамках виртуальной лаборатории, – сценарий иммуноферментного анализа крови для определения красной волчанки, являющейся серьезным аутоиммунным заболеванием. Реализованный эксперимент представлял собой линейную последовательность шагов, не допускавшую вариативности действий, присущих реальной лабораторной практике [8]. В то же время для обеспечения реалистичности образовательного процесса в виртуальных лабораториях необходимо должны быть реализованы различные варианты действий, совершаемых обучаемым, которые могут привести как к ошибочным, так и к правильным результатам [9].

Таким образом, возникает необходимость разработки **редактора сценариев** экспериментов для виртуальных лабораторий, позволяющего создавать серии экспериментов, допускающие большую степень вариативности. Для этого необходимо перейти от «жестко» запрограммированной (hardcoded) сценарной системы к так называемую «нодовой» (node-based) [10], к достоинствам которой относят легкую модифицируемость и масштабируемость решений [11]. Кроме того, создаваемый инструмент разработки виртуальных экспериментов должен быть удобен для работы без специальной технической подготовки.

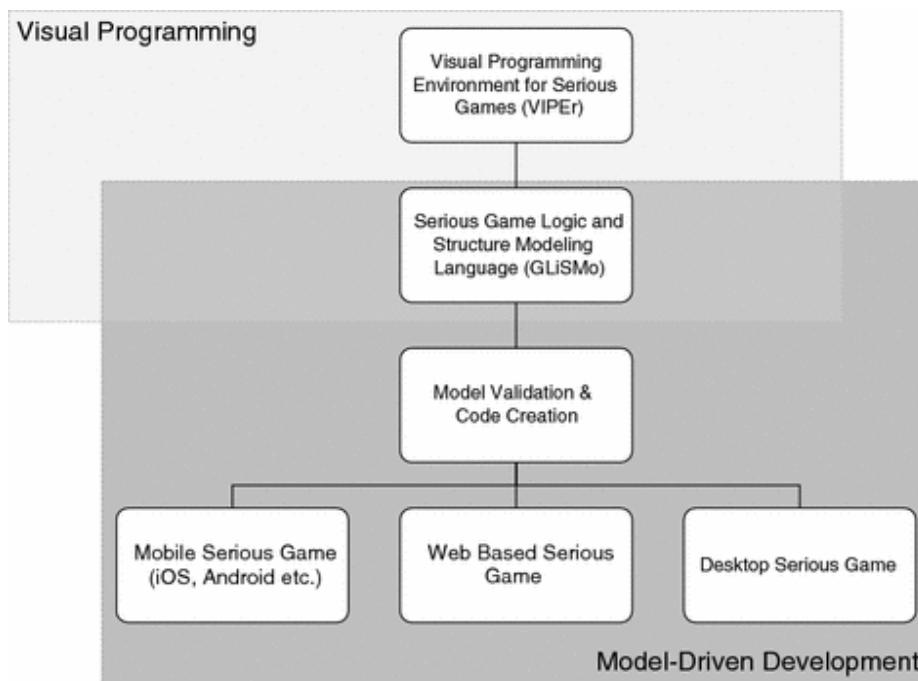


Рис. 1. Общий процесс модульной разработки серьезных игр, включая визуальный редактор VIPER

Node-based подход успешно применяется и для разработки компьютерных игр [12], его развитие представлено в [13], где этот подход успешно использован в визуальном редакторе программирования VIPEr, который применяется в сочетании со средствами других слоев архитектуры – языком моделирования для серьезных игр GLiSMo и управляемой моделью Toolchain (рис. 1).

## РЕАЛИЗАЦИЯ РЕДАКТОРА СЦЕНАРИЕВ

Для реализации комплекса виртуальных лабораторий и собственно нодового редактора был выбран программный продукт для создания интерактивных и мультимедийных приложений Unity [14], в первую очередь из-за гибких возможностей расширения базового редактора.

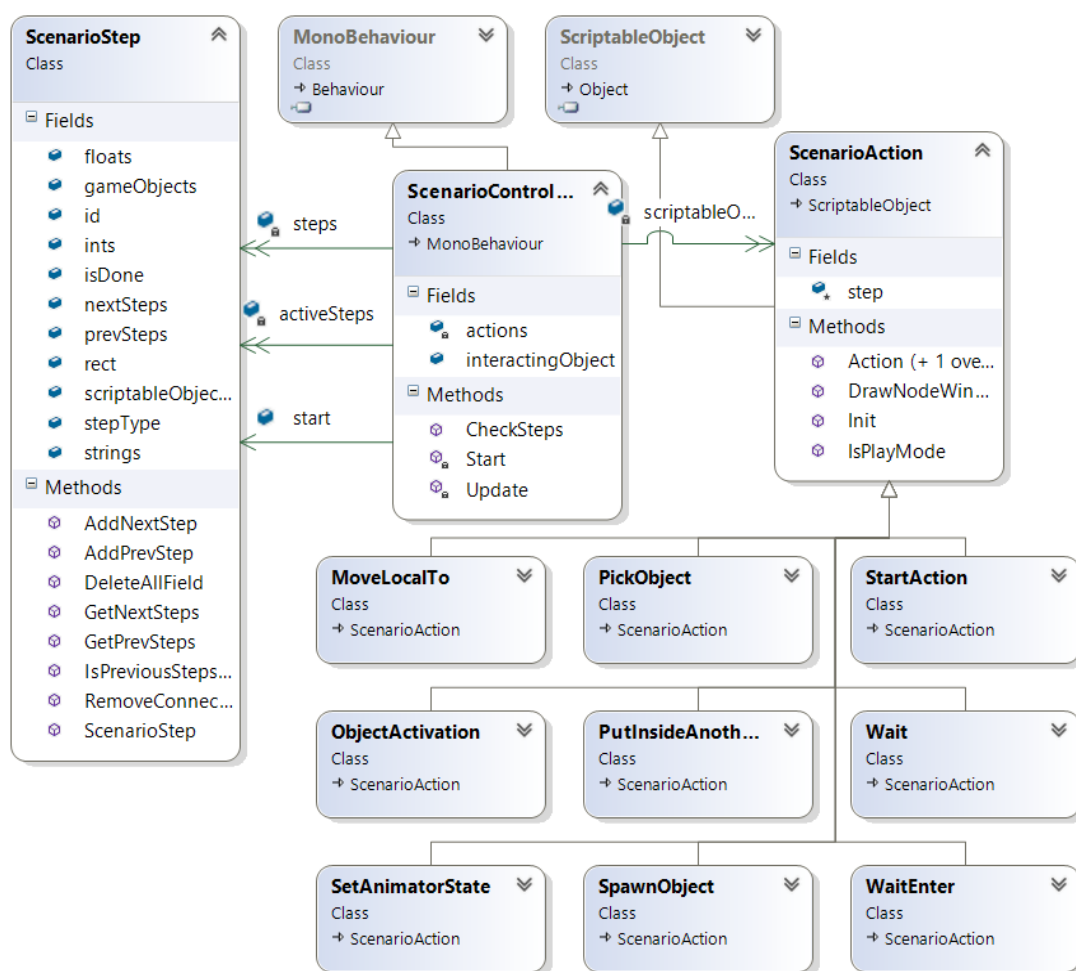


Рис. 2. Структура классов

Основой нашего подхода является визуальное построение модифицированного конечного автомата, который представляется в виде графа переходов, где узлы графа (или node) и его ребра описывают выполняемые процедуры и поток данных соответственно. Первоначально архитектура редактора сценариев основывалась на полиморфизме, но в Unity указатели на экземпляры дочерних классов неправильно сериализуются в файл сцены. Из-за этого при следующей загрузке сцены создавались экземпляры базового класса и терялись некоторые ссылки на другие объекты. Ещё одной проблемой оказалось то, что неверно сохраняются ссылки на один и тот же объект. Если на него ссылаются несколько объектов, то после повторной загрузки сцены создаются экземпляры объектов в количестве ссылок на него. В итоге структура классов приняла вид, представленный на рис. 2. На каждом шаге хранятся переменные разных типов, которые могут быть использованы при выполнении действия сценария: текущее состояние, указатели на следующие и предыдущие шаги, тип действия, которое выполняется через экземпляр ScriptableObject, и номер шага. Текущим состоянием управляет специальный контроллер, который хранит активные сценарные шаги и в случае их выполнения переходит к следующим шагам. В дочерних классах ScriptableObject указывается, как в окне редактора должно выглядеть действие. Каждый узел представляет собой шаг сценария и оформляется в виде графического компонента editor window, встроенного в Unity, с полями параметров. Два узла соединяются ребром, в результате сценарий имеет вид графа переходов (рис. 3).

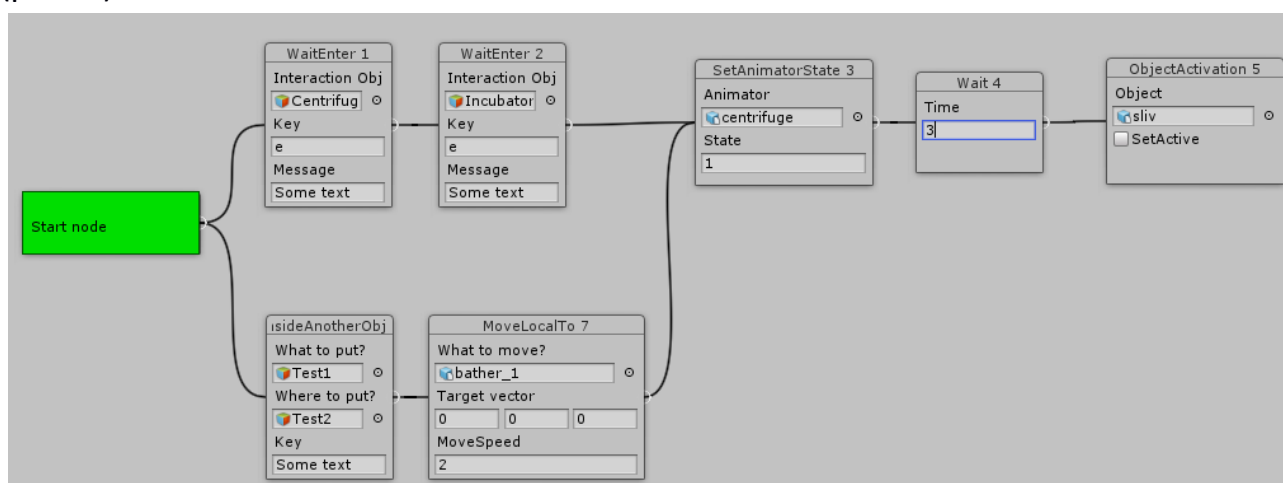


Рис. 3. Пример сценария, описанного в нодовом редакторе

Шаги сценария могут выполняться как последовательно (рис. 4), так и параллельно, в случае отсутствия зависимости шагов друг от друга (рис. 5).

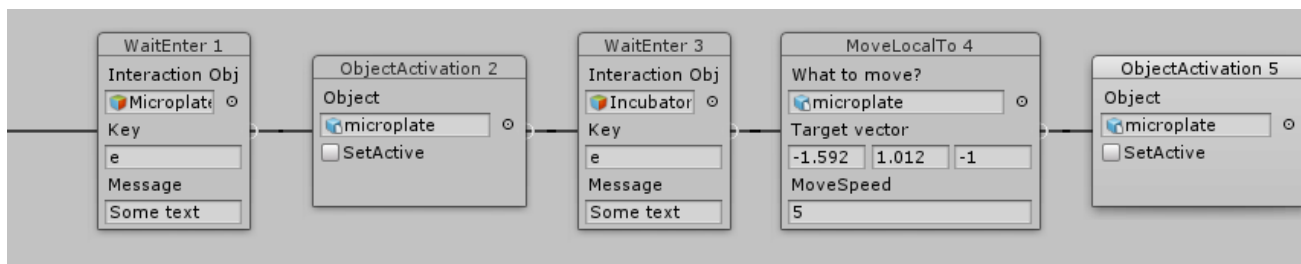


Рис. 4. Пример цепочки последовательных связей нескольких шагов

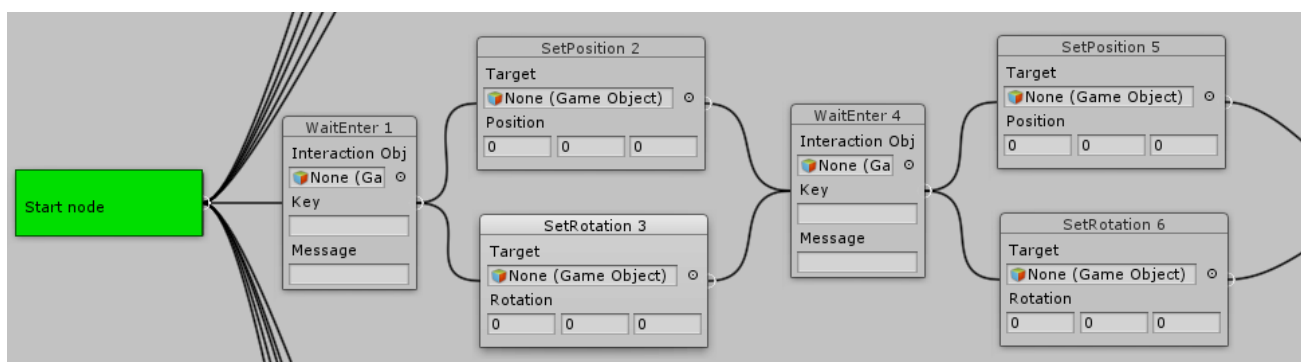


Рис. 5. Пример отрывка сценария с распараллеливанием

Обязательным компонентом сценария является стартовый узел (ScenarioStartAction, рис. 6), с которого должны начинаться все ветки сценария.

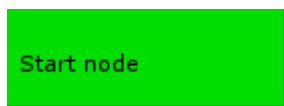


Рис. 6. Стартовый узел (ScenarioStartAction)

Компонент Loop (рис. 7) предназначен для реализации повторения действий, начиная с указанного компонента, определяющегося по идентификатору (Start id), фиксированное количество раз (Iteration count).

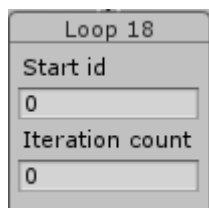


Рис. 7. Компонент повторения действий (Loop)

Компонент MoveLocalTo (рис. 8) предназначен для перемещения объекта из исходного координатного положения в заданное значение (Target vector) с определенной скоростью (MoveSpeed).

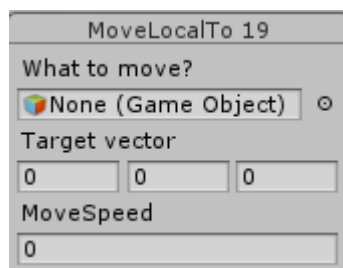


Рис. 8. Компонент перемещения объекта (MoveLocalTo)

Компонент ObjectActivation (рис. 9) переводит необходимый элемент из активного состояния в неактивное и обратно, делая объект видимым или невидимым.

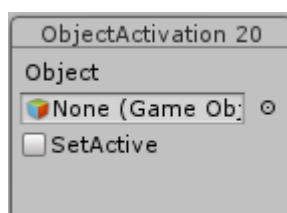


Рис. 9. Компонент изменения активности (ObjectActivation)

Компонент PutInsideAnotherObject (рис. 10) создает дочерний объект (What to put) для иерархически родительского (Where to put) по нажатию кнопки (Key).

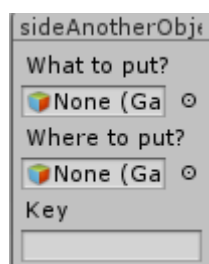


Рис. 10. Компонент создания дочернего объекта (PutInsideAnotherObject)

Компонент SetAnimatorState (рис. 11) включает анимацию объекта (с заданным идентификатором анимации State).

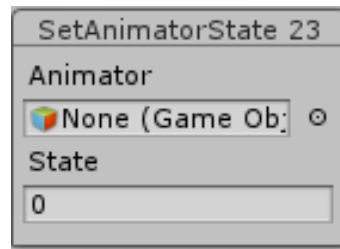


Рис. 11. Компонент включения анимации (SetAnimatorState)

Набор компонентов SetPosition, SetRotation, SetScale (рис. 12) изменяет соответственно местоположение, угол поворота и масштаб. SetPosition отличается от MoveLocalTo тем, что он мгновенно переносит объект в необходимое место. Все три компонента задают изменения по трем координатам.

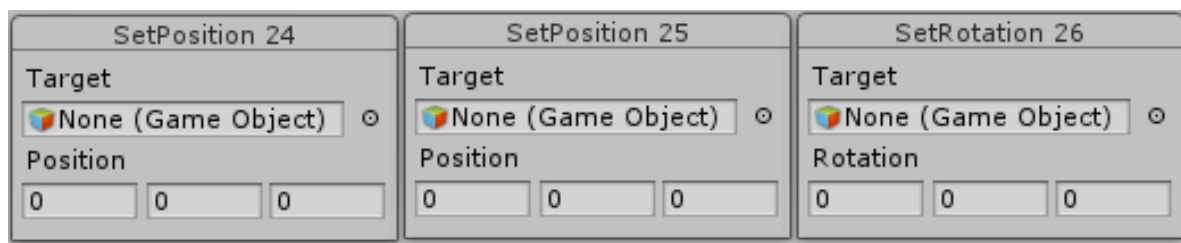


Рис. 12. Компоненты SetPosition, SetRotation, SetScale

Компонент ShowMessage (рис. 13) отображает текстовое сообщение (Message) в течение определенного времени (Show Time, в секундах).

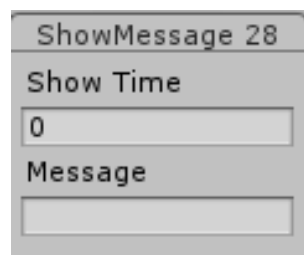


Рис. 13. Компонент добавления сообщений пользователю (ShowMessage)

Компонент SpawnObject (рис. 14) вставляет объект (Prefab) в заданное местоположение (Local Pos), с выбранным ракурсом обзора (Rotation).

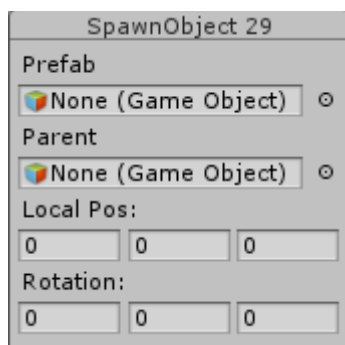


Рис. 14. Компонент вставки (SpawnObject)

Компонент Wait (рис. 15) задаёт паузу на определенный промежуток времени (Time, в минутах).

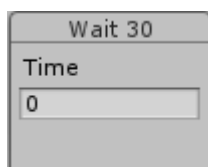


Рис. 15. Компонент паузы (Wait)

Компонент WaitEnter (рис. 16) создает паузу до нажатия кнопки (Key) либо до выбора указанного объекта с тегом активности (ActiveObject), с отображением текстового сообщения (Message) при наведении курсора на объект или в случае, когда пользователь находится в нужном месте.

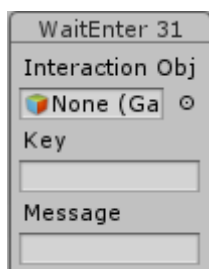


Рис. 16. Компонент выбора (WaitEnter)

Описанный базовый набор компонентов для реализации сценариев работы пользователя с элементами сцены (виртуальной лаборатории) при необходимости можно расширить, непосредственно программируя действия новых компонентов, например, компонента озвучки какого-то действия или компонента справки об объекте сцены.



На рис. 17 приведен отрывок описания сценария переноса (SetPosition5) с анимацией (SetAnimationState4) микропланшета с сывороткой (объект microplat указан в WaitEnter3) по нажатию заданной кнопки с поворотом (SetRotation6) в инкубатор (указан как цель – dispenser), причём влияние поворота существенно для корректного отображения процедуры (рис. 18). Все три действия (перенос, поворот, анимация) выполняются одновременно. Пока все они не будут выполнены, сценарий не перейдёт к следующему шагу.

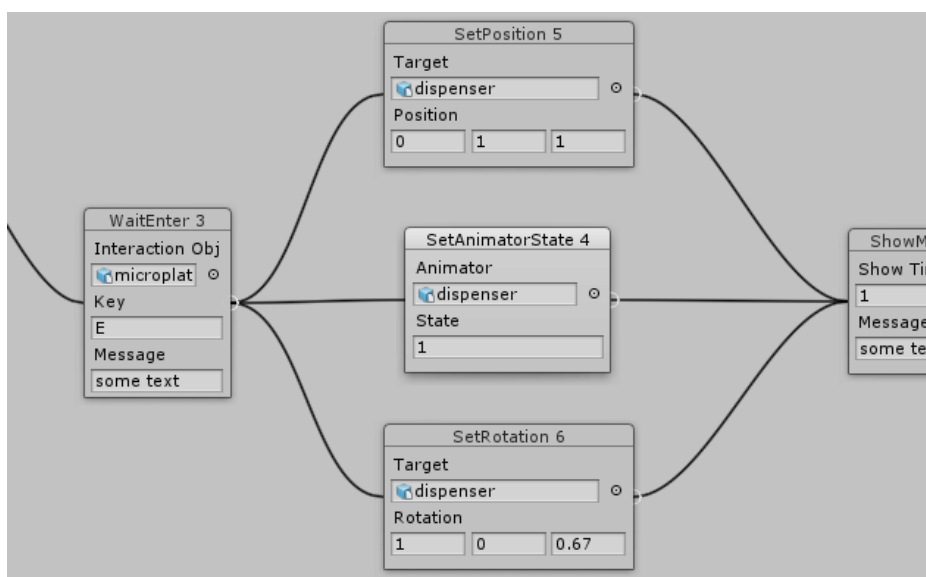


Рис. 17. Отрывок сценария переноса микропланшета в инкубатор

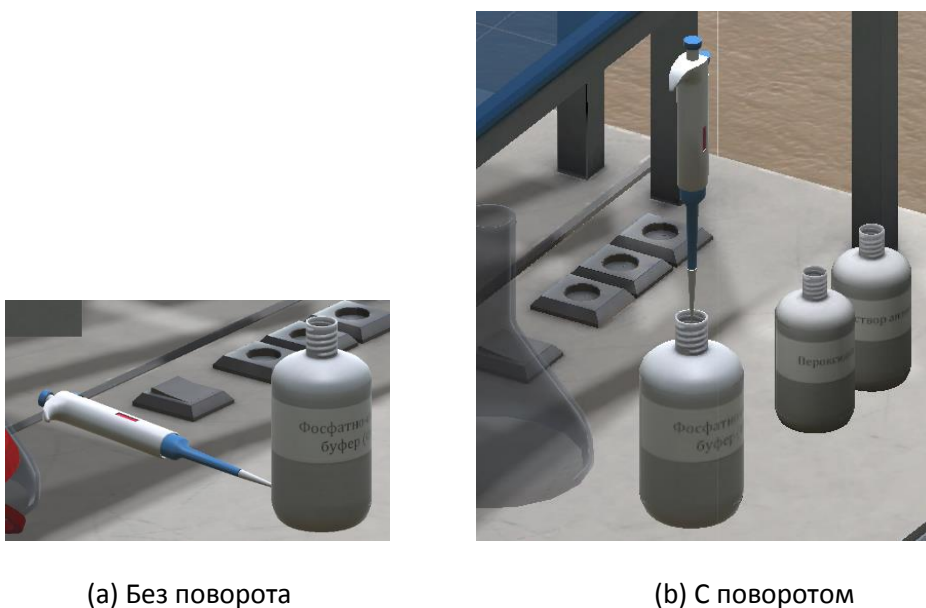


Рис. 18. Влияние поворота при переносе микропланшета в инкубатор



- доступность игрового (для виртуальной лаборатории – образовательного) уровня для познания,
- мотивация (достижение, ожидание поощрения или интересного продолжения, награда за прохождение уровня, соперничество и др.),
- вариативность сценария.

Соответственно, любой продукт можно представить в как точку в данном треугольнике, находящуюся на разных расстояниях от его вершин, что иллюстрирует степень баланса между тремя указанными параметрами.

Соблюдение подобного баланса является нетривиальной задачей, поскольку некоторые когнитивные и мотивационные цели находятся в противоречии друг с другом, и, как следствие, максимизация одной цели может отрицательно отразиться на другой. Достижение вовлеченности должно быть реализовано ещё в геймдизайне.

Сценарий виртуального эксперимента должен отражать этот баланс «железного треугольника» вовлеченности. Использование визуального редактора позволяет реализовать эти аспекты сценария, соотнеся компоненты редактора сценариев с геймификационными элементами геймдизайна, тем самым в явном виде решая одновременно задачи как проектирования сценария, так и повышения вовлеченности (как следствие, и качества освоения материала).

Один из реализованных подходов вовлеченности предложен в [18] для студентов, изучающих программирование: в рамках игры студенты анализируют задачи, формулируют и пишут программы для потенциально выигрывающих стратегий. Основываясь на опыте, студенты могут совершенствовать свои стратегии для достижения лучших результатов – и, как следствие, мотивация студентов к заданию значительно повышается. Они готовы тратить больше времени на повторяющуюся практику.

## **ЗАКЛЮЧЕНИЕ**

Разработанный редактор сценариев позволяет гибко создавать серии экспериментов для виртуальных лабораторий. Кроме того, в силу своей архитектуры он позволяет проектировать работу виртуальной лаборатории, удовлетворяя всем критериям «железного треугольника».

Следующей важной задачей является оценка вовлеченности игрока, для ее решения используются различные методики. Стоит отметить предложенный в [19] подход, основанный на трекинге взгляда игрока.

Для оценки удобства и простоты использования разработанной визуальной среды для программирования сценариев экспериментов необходимо провести ряд исследований по оценке юзабилити. Я. Нильсен [20] различает несколько типов методов проверки юзабилити – эвристические оценки [21, 22], когнитивные маршруты сценариев [23], формальные проверки [24], вариативность сценариев [25], обзор функциональных возможностей [26] и согласованности [27].

Необходимо определить, будет ли пользователь в состоянии освоить поставленные перед ним задачи по самостоятельному визуальному программированию сценариев с меньшими затратами и без каких-либо инструкций. Важно выявить непоследовательные нюансы в использовании редактора и трудности пользователей в его освоении.

#### **СПИСОК ЛИТЕРАТУРЫ**

1. *Veljko Potkonjak, Michael Gardner, Victor Callaghan, Pasi Mattila, Christian Guetl, Vladimir M. Petrovic, and Kosta Jovanovich.* Virtual Laboratories for Education in Science, Technology, and Engineering: a Review // *Computers & Education*. April 2016. V. 95. P. 309–327.

2. *Balamuralithara B., Woods P.C.* Virtual Laboratories in Engineering Education: The Simulation Lab and Remote Lab // *Computer Applications in Engineering Education*, March 2009. V. 17. Issue 1. P. 108–118.

3. *Daniel Fernández-Avilés, Diego Dotor, Daniel Contreras, Jose Carlos Salazar.* Virtual Labs: A New Tool in the Education: Experience of Technical University of Madrid // *13th Int. Conference on Remote Engineering and Virtual Instrumentation (REV)*, 24–26 Feb. 2016. P. 271–272.

4. *Zoric Nedic, Jan Machotka, Andrew Nafalski.* Remote Laboratories Versus Virtual and Real Laboratories // *Frontiers in Education*. 2003. FIE 2003 33rd Annual, T3E-1-T3E-6. V. 1.

5. *Михайлов В.Ю., Гостев В.М., Кузуракова В.В., Чугунов В.А.* Виртуальная лаборатория как средство обеспечения коллективной научно-методической ра-

боты // Сб. трудов XII международной конференции «ИТО-2002». Часть IV (Москва, 4–8 ноября 2002 г.). М.: МИФИ, 2002. С. 31–34.

6. *Abramov V.D., Kugurakova V.V., Rizvanov A.A., Abramskiy M.M., Manakhov N.R., Evstafiev M.M.* Virtual Biotechnological Lab // *BioNanoSci*. 2016. <http://link.springer.com/article/10.1007/s12668-016-0368-9>

7. *Абрамов В.Д., Кугуракова В.В., Ризванов А.А., Абрамский М.М., Манахов Н.Р., Евстафьев М.Е.* Виртуальные лаборатории как средство обучения биомедицинским технологиям // *Russian Digital Libraries Journal*. 2016. V. 19. No 3. P. 129–148.

8. *Gameplay*: <https://en.wikipedia.org/wiki/Gameplay>.

9. *Nonlinear Gameplay*: [https://en.wikipedia.org/wiki/Nonlinear\\_gameplay](https://en.wikipedia.org/wiki/Nonlinear_gameplay).

10. *Silva P.B., Müller P., Bidarra R., and Coelho A.* Node-based Shape Grammar Representation and Editing // *Proceedings of the Workshop on Procedural Content Generation in Games (PCG '13)*, May 2013. 8 p. <https://graphics.tudelft.nl/Publications-new/2013/SMBC13a/SMBC13a.pdf>.

11. *Patow G.* User-friendly Graph Editing for Procedural Modeling of Buildings // *IEEE Computer Graphics and Applications*. 2012. V. 32. No 2. P. 66–75.

12. *Thillainathan N.* A Model Driven Development Framework for Serious Games // *Information Systems*. 2013. Informatik 2013, Doctoral Consortium, Koblenz, Germany. P. 81-92. <https://pdfs.semanticscholar.org/67b5/c60d12628e3aebe64e61c133147d60451049.pdf>.

13. *Thillainathan N., Leimeister J.M.* Educators as Game Developers — Model-driven Visual Programming of Serious Games // *Knowledge, Information and Creativity Support Systems*. 2016. V. 416 of the book series “Advances in Intelligent Systems and Computing (AISC)”. P. 335–349. [http://link.springer.com/chapter/10.1007/978-3-319-27478-2\\_23](http://link.springer.com/chapter/10.1007/978-3-319-27478-2_23).

14. *Manual Unity*: <https://docs.unity3d.com/ru>.

15. *A. Rollings and D. Morris.* *Game Architecture and Design—A New Edition*, New Riders, 2004. 930 p. <https://gamifique.files.wordpress.com/2011/11/6-game-architecture-and-design-a-new-edition.pdf>

16. *Derbali L., Ghali R. and Frasson C.* Assessing Motivational Strategies in Serious Games Using Hidden Markov Models // *Proceedings of the 26th Int. Florida Artifi-*

cial Intelligence Research Society Conference, 2013. P. 538–541.

<http://www.aaai.org/ocs/index.php/FLAIRS/FLAIRS13/paper/viewFile/5941/6133>.

17. *Boyer K.E., Phillips R., Wallis M., Vouk M., and Lester J.* Balancing Cognitive and Motivational Scaffolding in Tutorial Dialogue // Lecture Notes in Computer Science (LNCS) (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2008. V. 5091. P. 239–249.

18. *Jiau H.C., Chen J.C., and Ssu K.* Enhancing Self-motivation in Learning Programming Using Game-based Simulation and Metrics // IEEE Transactions on Education. 2009. V. 52, No 4. P. 555–562. <http://ieeexplore.ieee.org/document/5164890/> authors

19. *Leidig S.* Analysis of User Attention for Adaptive Serious Games – Design and Implementation of an Evaluation Framework // Master’s Thesis. Karlsruhe Institute of Technology, 2016. 86 p.

20. *Nielsen J.* Usability Inspection Methods// Conference Companion on Human Factors in Computing Systems, Denver, Colorado, USA, ACM (1995). P. 377–378.

21. *Nielsen J.* Enhancing the Explanatory Power of Usability Heuristics // Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, 1994.

22. *Nielsen J.* Usability Engineering. Elsevier, 1994. 362 p.

23. *Polson P.G. et al.* Cognitive Walkthroughs: a Method for Theory-based Evaluation of User Interfaces// Int. J. Man Mach. Stud. 1992. V. 36. No 5. P. 741–773.

24. *Kahn M.J., Prail A.* Formal Usability Inspections// In: Usability Inspection Methods. Wiley, New York, 1994. P. 141–171.

25. *Bias R.G.* The Pluralistic Usability Walkthrough: Coordinated Empathies// In: Usability Inspection Methods. Edited by Jakob Nielsen and Robert L. Mack, published by John Wiley & Sons, New York Wiley, New York, 1994. <https://www.nngroup.com/books/usability-inspection-methods/>

26. *Bell B.R.* Walkthroughs. Using Programming: Using Programming Walkthroughs to Design a Visual Language. University of Colorado at Boulder. Boulder, CO, USA, 1992.

27. *Wixon D. et al.* Inspections and Design Reviews: Framework, History and Reflection // In: Usability Inspection Methods. Edited by Jakob Nielsen and Robert L.

Mack, published by John Wiley & Sons, New York Wiley, New York, 1994.  
<https://www.nngroup.com/books/usability-inspection-methods/>

---

## VISUAL EDITOR OF SCENARIOS FOR VIRTUAL LABORATORIES

V.D. Abramov<sup>1</sup>, M.M. Abramskiy<sup>2</sup>, V.V. Kugurakova<sup>3</sup>, N.R. Manakhov<sup>4</sup>,  
A.M. Maslaviev<sup>5</sup>

*Higher School of Information Technologies and Information Systems. Kazan (Volga Region) Federal University*

<sup>1</sup>ivitazour@gmail.com, <sup>2</sup>ma@it.kpfu.ru, <sup>3</sup>vlada.kugurakova@gmail.com,

<sup>4</sup>nadirmanakhov@gmail.com, <sup>5</sup>erasumus@mail.ru

### Abstract

The paper presents the implementation of a visual script editor that allows you to design a series of experiments for virtual educational laboratories. The connection of the above approach with the possibility of explicit control over students' involvement and methods of its evaluation is described.

**Keywords:** *virtual lab, virtual laboratory, node-based editor, involvement*

### REFERENCES

1. Veljko Potkonjak, Michael Gardner, Victor Callaghan, Pasi Mattila, Christian Guetl, Vladimir M. Petrovic, Kosta Jovanovich. Virtual Laboratories for Education in Science, Technology, and Engineering: a Review // *Computers & Education*. April 2016. V. 95. P. 309–327.

2. Balamuralithara B., Woods P.C. Virtual Laboratories in Engineering Education: The Simulation Lab and Remote Lab // *Computer Applications in Engineering Education*, March 2009. V. 17. Issue 1. P. 108–118.

3. Daniel Fernández-Avilés, Diego Dotor, Daniel Contreras, Jose Carlos Salazar. Virtual Labs: A New Tool in the Education: Experience of Technical University of Madrid // 13th Int. Conference on Remote Engineering and Virtual Instrumentation (REV), 24–26 Feb. 2016, P. 271–272.



4. *Zoric Nedic, Jan Machotka, Andrew Nafalski*. Remote Laboratories Versus Virtual and Real Laboratories // *Frontiers in Education*. 2003. FIE 2003 33rd Annual, T3E-1-T3E-6. V. 1.

5. *Mihaylov V.Yu., Gostev V.M., Kugurakova V.V., Chugunov V.A.* Virtualnaya laboratoriya kak sredstvo obespecheniya kollektivnoy nauchno-metodicheskoy raboty // *Sb. trudov XII mezhdunarodnoy konferentsii «ITO-2002»*. Chast IV (Moskva, 4–8 noyabrya 2002 g.). M.: MIFI, 2002. S. 31–34.

6. *Abramov V.D., Kugurakova V.V., Rizvanov A.A., Abramskiy M.M., Manakhov N.R., Evstafiev M.M.* Virtual Biotechnological Lab // *BioNanoSci*. 2016. <http://link.springer.com/article/10.1007/s12668-016-0368-9>.

7. *Abramov V.D., Kugurakova V.V., Rizvanov A.A., Abramskiy M.M., Manakhov N.R., Evstafiev M.E.* Virtualnyie laboratorii kak sredstvo obucheniya biomeditsinskim tehnologiyam // *Russian Digital Libraries Journal*. 2016. V. 19. No 3. S. 129–148.

8. Gameplay: <https://en.wikipedia.org/wiki/Gameplay>.

9. Nonlinear Gameplay: [https://en.wikipedia.org/wiki/Nonlinear\\_gameplay](https://en.wikipedia.org/wiki/Nonlinear_gameplay).

10. *Silva P.B., Müller P., Bidarra R., and Coelho A.* Node-based Shape Grammar Representation and Editing // *Proceedings of the Workshop on Procedural Content Generation in Games (PCG '13)*, May 2013. 8 p. <https://graphics.tudelft.nl/Publications-new/2013/SMBC13a/SMBC13a.pdf>.

11. *Patow G.* User-friendly Graph Editing for Procedural Modeling of Buildings // *IEEE Computer Graphics and Applications*. 2012. V. 32. No 2. P. 66–75.

12. *Thillainathan N.* A Model Driven Development Framework for Serious Games // *Information Systems*. <https://pdfs.semanticscholar.org/67b5/c60d12628e3aeb64e61c133147d60451049.pdf>.

13. *Thillainathan N., Leimeister J.M.* Educators as Game Developers — Model-driven Visual Programming of Serious Games // *Knowledge, Information and Creativity Support Systems*. 2016. V. 416 of the book series *Advances in Intelligent Systems and Computing (AISC)*. P. 335–349. [http://link.springer.com/chapter/10.1007/978-3-319-27478-2\\_23](http://link.springer.com/chapter/10.1007/978-3-319-27478-2_23).

14. Manual Unity: <https://docs.unity3d.com/ru>.



15. A. Rollings, and D. Morris. Game Architecture and Design—A New Edition, New Riders, 2004. 930 p. <https://gamifique.files.wordpress.com/2011/11/6-game-architecture-and-design-a-new-edition.pdf>

16. Derbali L., Ghali R. and Frasson C. Assessing Motivational Strategies in Serious Games Using Hidden Markov Models // Paper presented at the FLAIRS 2013 – Proceedings of the 26th Int. Florida Artificial Intelligence Research Society Conference, 2013. P. 538–541. <http://www.aaai.org/ocs/index.php/FLAIRS/FLAIRS13/paper/viewFile/5941/6133>.

17. Boyer K.E., Phillips R., Wallis M., Vouk M., and Lester J. Balancing Cognitive and Motivational Scaffolding in Tutorial Dialogue // Lecture Notes in Computer Science (LNCS) (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2008. V. 5091. P. 239–249.

18. Jiau H.C., Chen J.C., and Ssu K. Enhancing Self-motivation in Learning Programming Using Game-based Simulation and Metrics // IEEE Transactions on Education. 2009. V. 52, No 4. P. 555–562. <http://ieeexplore.ieee.org/document/5164890/authors>

19. Leidig S. Analysis of User Attention for Adaptive Serious Games – Design and Implementation of an Evaluation Framework // Master's Thesis. Karlsruhe Institute of Technology, 2016. 86 p.

20. Nielsen J. Usability Inspection Methods// Conference Companion on Human Factors in Computing Systems, Denver, Colorado, USA, ACM (1995). P. 377–378.

21. Nielsen J. Enhancing the Explanatory Power of Usability Heuristics// Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM. 1994.

22. Nielsen J. Usability Engineering. Elsevier, 1994. 362 p.

23. Polson P.G. et al. Cognitive Walkthroughs: a Method for Theory-based Evaluation of User Interfaces// Int. J. Man Mach. Stud. 1992. V. 36. No 5. P. 741–773.

24. Kahn M.J., Prail A. Formal Usability Inspections// In: Usability Inspection Methods. Wiley, New York, 1994. P. 141–171.

25. Bias R.G. The Pluralistic Usability Walkthrough: Coordinated Empathies// In: Usability Inspection Methods. Edited by Jakob Nielsen and Robert L. Mack, published

by John Wiley & Sons, New York Wiley, New York, 1994. <https://www.nngroup.com/books/usability-inspection-methods/>

26. *Bell B.R.* Walkthroughs. Using Programming: Using Programming Walkthroughs to Design a Visual Language. University of Colorado at Boulder. Boulder, CO, USA, 1992.

27. *Wixon D. et al.* Inspections and Design Reviews: Framework, History and Reflection // In: Usability Inspection Methods. Edited by Jakob Nielsen and Robert L. Mack, published by John Wiley & Sons, New York Wiley, New York, 1994. <https://www.nngroup.com/books/usability-inspection-methods/>

### СВЕДЕНИЯ ОБ АВТОРАХ



**АБРАМОВ Виталий Денисович** – лаборант-исследователь Научно-исследовательской лаборатории «Виртуальные и симуляционные технологии в биомедицине» Высшей школы информационных технологий и информационных систем Казанского (Приволжского) федерального университета (КФУ).

**Vitalii Denisovich Abramov**, junior research scientist of Kazan Federal University. Current scientific interests: virtual laboratories.

email: [ivitazour@gmail.com](mailto:ivitazour@gmail.com)



**АБРАМСКИЙ Михаил Михайлович** – старший преподаватель кафедры инжиниринга программного обеспечения Высшей школы информационных технологий и информационных систем КФУ

**Mikhail Mikhailovich ABRAMSKIY**, Senior Lecturer of Higher School of Information Technology and Information Systems of Kazan Federal University.

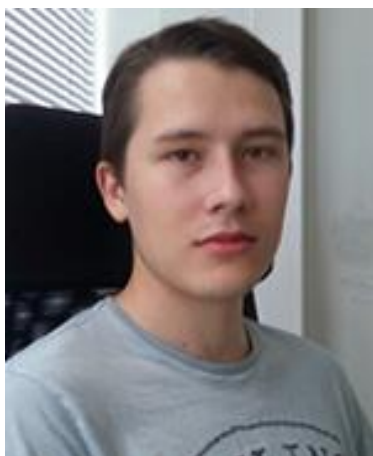
email: [ma@it.kfu.ru](mailto:ma@it.kfu.ru)



**КУГУРАКОВА Влада Владимировна** – старший преподаватель Высшей школы информационных технологий и информационных систем КФУ, руководитель Научно-исследовательской лаборатории «Виртуальные и симуляционные технологии в биомедицине».

**Vlada Vladimirovna KUGURAKOVA**, senior Lecturer of Higher School of Information Technology and Information Systems, Head of Laboratory "Virtual and simulational technologies in biomedicine".

email: vlada.kugurakova@gmail.com



**МАНАХОВ Надир Ринатович** – младший научный сотрудник Научно-исследовательской лаборатории «Виртуальные и симуляционные технологии в биомедицине» Высшей школы информационных технологий и информационных систем КФУ.

**Nadir Rinatovich MANAKHOV**, junior research scientist of Kazan Federal University. Current scientific interests: virtual laboratories.

email: nadirmanakhov@gmail.com



**МАСЛАВИЕВ Азат Мунирович** – Студент 4 курса Высшей школы информационных технологий и информационных систем КФУ, направление «Прикладная информатика».

**Azat Munirovich MASLAVYEV**, student of 4 grade of Higher Institute of Information Technologies and Information Systems of Kazan Federal University.

email: erasmus@mail.ru

*Материал поступил в редакцию 10 ноября 2016 года*