**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ**

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФГБОУ ВО

**«БРЯНСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**Кафедра** «Информатика и программное обеспечение»

**КУРСОВАЯ РАБОТА**

По дисциплине

**«Визуальное программирование»**

**Тема:** *«Игра Snake»*

Вариант № 6

всего листов 18

Выполнил студент гр. 18-ИВТ-2

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Жирешонков Н.О.

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г.

Руководитель

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ к.т.н., доцент. Трубаков Е.О.

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г.

г. Брянск 2020 г.

Оглавление

[Введение 3](#_Toc42187120)

[1. Анализ предметной области 4](#_Toc42187121)

[1.1. *Примеры существующих проектов:* 4](#_Toc42187122)

[1.1.1. Snake 4](#_Toc42187123)

[1.1.2. Snake II 4](#_Toc42187124)

[1.1.3. Эра 3D 5](#_Toc42187125)

[1.2. Выводы 6](#_Toc42187126)

[2. Конструкторская часть 7](#_Toc42187127)

[2.1. Движок игры 7](#_Toc42187128)

[2.2. Интерфейс 9](#_Toc42187129)

[3. Эксперементальная часть 11](#_Toc42187130)

[Заключение 17](#_Toc42187131)

[Список литературы 18](#_Toc42187132)

# Введение

Целью курсовой работы является приобретение дополнительных навыков для работы на языке C++/CLL.

В качестве темы курсовой выступит создание игры «Snake».

# Анализ предметной области

## Примеры существующих проектов:

### Snake

Оригинальная «Змейка» (Snake) от Nokia появилась в 1997 году благодаря стараниями разработчика Танели Орманто. В том же году компания выпустила первый телефон с этой игрой — Nokia 6110. Уже тогда игра была многопользовательской: телефоны общались через ИК-порты, ведь ни Bluetooth, ни тем более Wi-Fi в телефонах в то время не было. Сама змейка состояла из чёрных квадратов и могла двигаться в четырёх направлениях. Игровая зона, по которой передвигалось пресмыкающееся, была ограничена размерами экрана телефона: при ударе головы змейки о край телефона игра завершалась. «Змейка» приобрела невероятную популярность, сравнимую разве что с популярностью современных хитов «Angry Birds» и «Cut the Rope».

### Snake II

Вторая часть «Змейки» — "Snake II" — обзавелась «бесшовной» игровой зоной, так что змейка не «умирала», врезаясь в край экрана, а выползала из противоположной части дисплея. Также были добавлены карты с препятствиями, врезаться в которые было нельзя. В новой версии был и чит: если успевать ставить игру на паузу в момент поглощения пищи змейкой, она не увеличивалась в размерах. С помощью этой уловки усидчивые, но не очень честные геймеры зарабатывали умопомрачительный счёт, чтобы потом похвастаться перед друзьями. В итоге, не вполне честным путём люди зарабатывали и 20, и 30 тысяч очков — это при том, что в первой «Змейке» больше 4500 баллов набрать было невозможно.

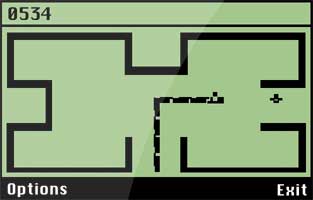


Рис. 1. Snake 2

Впоследствии Nokia выпустила игру Snake Xenzia для монохромных и бюджетных цветных телефонов, Snake EX и Snake EX2 с улучшенной графикой и мультиплеером через Bluetooth для цветных телефонов на S60 и S40.

### Эра 3D

В январе 2005 года специально для N-Gage компанией IOMO (являющейся крупнейшим разработчиком мобильных игр в Европе на то время) по заказу Nokia была разработана новая, шестая, версия игры — «Змейки», или же "Snakes". На этот раз игра была уже трёхмерной, а графика соответствовала уровню Sony PlayStation. В «Snakes» присутствовал мультиплеер через Bluetooth на 4-х человек, была реализована «вирусная» быстрая передача самой игры на N-Gage друзей.



Рис. 2. Snakes

Следующей версией игры стала трёхмерная "Snake III". Она повторяла игру «Snakes» и отличалась лишь реалистичностью графики — всё-таки в «Snakes» змеи были довольно абстрактные. В остальном же игра содержала те же режимы игры и тот же мультиплеер через Bluetooth.

Наконец, в 2008 году вышла последняя версия «Змейки» — "Snakes Subsonic". «Сабсоник» являлся продолжением игры «Snakes», но уже для игрового сервиса «N-Gage», доступного на двух десятках устройств Nokia.

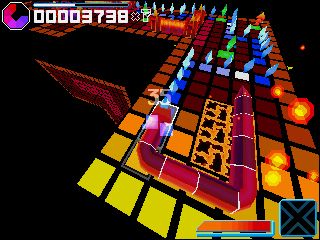


Рис. 3. Snakes Subsonic

## Выводы

В следствие анализа продуктов аналогов было принято решение реализовывать следующие особенности:

1. Работа будет выполнена в двухмерном пространстве.
2. Игровая зона будет поделена на квадраты.
3. Увеличение размера змейки.
4. Возможность столкновения с препятствиями.

# Конструкторская часть

## Движок игры

Целью работы является создание классической «Змейки» с базовыми правилами игры, но в ходе работы было реализовано несколько механик, наиболее интересными из которых я хотел бы поделиться.

Одной из самых интересных механик является сама змейка - она реализована как динамическая структура данных. Основное преимущество этого метода реализации заключается в механике перемещения: ДСД при перемещении позволяет не переписывать координаты каждого блока, а просто переставлять последний блок вперед первого.

|  |
| --- |
| void Snake::Crawl(int direction\_)  {  Node\* tmp = tail;  tail = tail->next;  tail->prev = nullptr;  tmp->prev = head;  tmp->next = nullptr;  head->next = tmp;  head = tmp;  switch (direction\_)  {  case UP:  head->x = head->prev->x;  head->y = head->prev->y - 1;  head->prev->direction = head->direction = UP;  break;  case RIGHT:  head->x = head->prev->x + 1;  head->y = head->prev->y;  head->prev->direction = head->direction = RIGHT;  break;  case DOWN:  head->x = head->prev->x;  head->y = head->prev->y + 1;  head->prev->direction = head->direction = DOWN;  break;  case LEFT:  head->x = head->prev->x - 1;  head->y = head->prev->y;  head->prev->direction = head->direction = LEFT;  break;  }  } |

Листинг 1. Функция перемещения

Для увеличения размера вызывается функция “Growing”, которая создает новый блок в конце змейки, и, что примечательно, присуждает ему данные блока хвоста. Таким образом на конце змейки отрисовываются сразу 2 блока, один из которых исчезает при следующем перемещении.

|  |
| --- |
| void Snake::Growing()  {  Node\* tmp = new Node;  tmp->next = tail;  tmp->prev = nullptr;  tmp->x = tail->x;  tmp->y = tail->y;  tmp->direction = tail->direction;  tail->prev = tmp;  tail = tmp;  SnakeSize++;  } |

Листинг 2. Функция “Growing”

Функция проверки на столкновения вызывается после каждого перехода, и проверяет, не пересеклась ли голова с каким-либо объектом.

|  |
| --- |
| private: System::Void Collision()  {  if (mapMas[snake->head->y][snake->head->x] == 1)  {  Annul();  snake->CreateSnake(mapSize);  return;  }  for (Node\* cur = snake->head->prev; cur != nullptr; cur = cur->prev)  {  if (snake->head->x == cur->x && snake->head->y == cur->y)  {  snake->Cut(cur->next);  break;  }  }  if (mapMas[snake->head->y][snake->head->x] == 2)  {  mapMas[snake->head->y][snake->head->x] = 0;  snake->Growing();  apple = false;  }  } |

Рис. 4. Функция проверки на столкновения

## Интерфейс

Рассмотрим интерфейс программы.

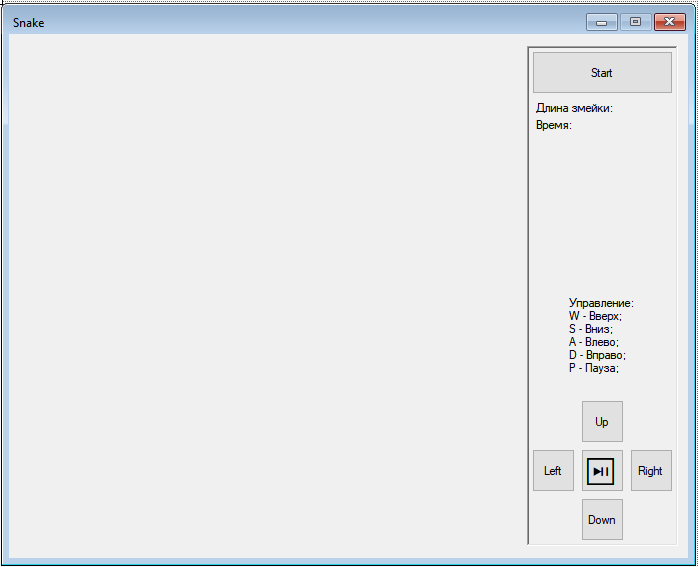


Рис. 5. Интерфейс программы

Большую часть экрана занимает игровое поле, при этом в правой его части расположена панель управления и информации. На ней можно увидеть кнопку запуска игры, кнопки управления направлением движения, кнопку паузы, и области с информацией.

Кнопка “Start” запускает игру. При повторном нажатии на нее игра перезапускается. Под этой кнопкой находятся текстовые окна, в которых отображаются длинна змейки, время игры в секундах и подсказка с управлением.

В нижней части панели расположены клавиши управления. Они расположены интуитивно, положение кнопки относительно остальных соответствует направлению движения змейки. По центру, между этими кнопками, находится кнопка паузы, которая останавливает внутренний таймер игры. Каждой клавише из этого блока соответствует физическая кнопка. Клавишами управления классически выступает набор WASD, а кнопке паузы соответствует кнопка “P”.

|  |
| --- |
| private: System::Void Snake\_Game\_KeyDown(System::Object^ sender, System::Windows::Forms::KeyEventArgs^ e)  {  switch (e->KeyCode) {  case Keys::D:  if (direction != LEFT)  direction = RIGHT;  break;  case Keys::A:  if (direction != RIGHT)  direction = LEFT;  break;  case Keys::W:  if (direction != DOWN)  direction = UP;  break;  case Keys::S:  if (direction != UP)  direction = DOWN;  break;  case Keys::P:  if (timer1->Enabled)  timer1->Stop();  else  timer1->Start();  break;  }  } |

Листинг 3. Функция считывания нажатия клавиш

# Эксперементальная часть

Чтобы удостовериться в корректности работы проекта, проведём несколько тестов с целью выявления ошибок.

Для начала попробуем выползти за пределы карты. Дадим команду «ползти вверх» и подождем, пока змейка не дойдёт до края.

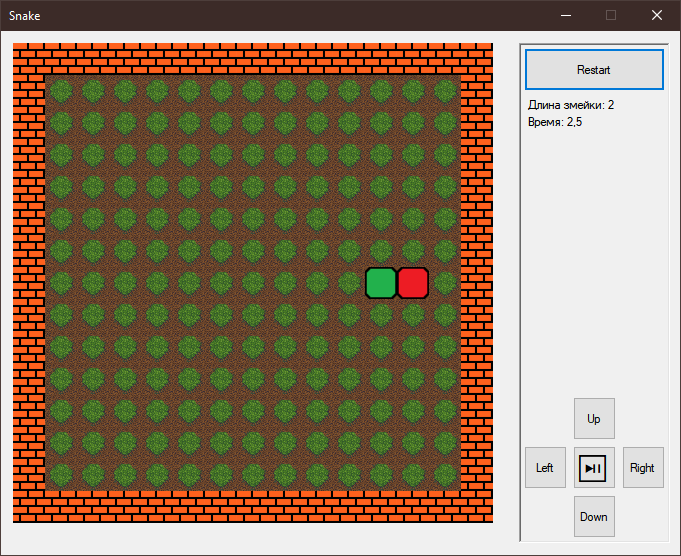


Рис. 6. Проверка коллизии

Обработка коллизии на краях карты происходит без ошибок: змейка погибает, не уходя за край карты. Проверка пройдена.

Теперь проверим функцию роста. Направим змейку на яблоко.

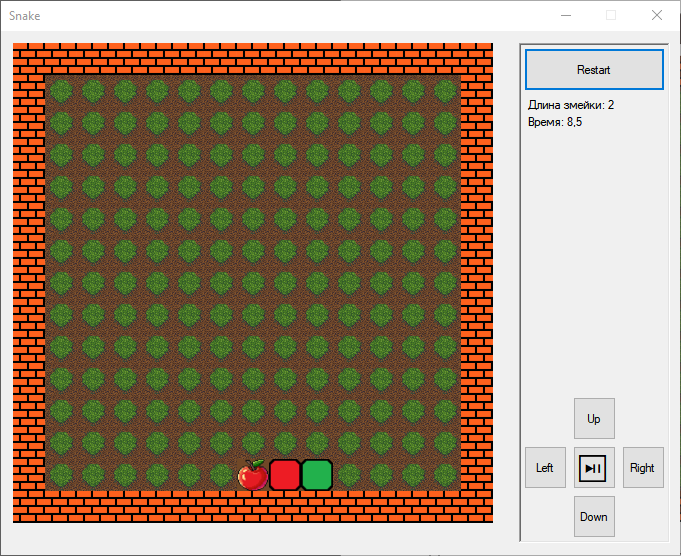


Рис. 7. Проверка функции “Growing”

После «столкновения» с яблоком змейка увеличилась на 1 сегмент, а яблоко пропало с карты. Проверка пройдена.

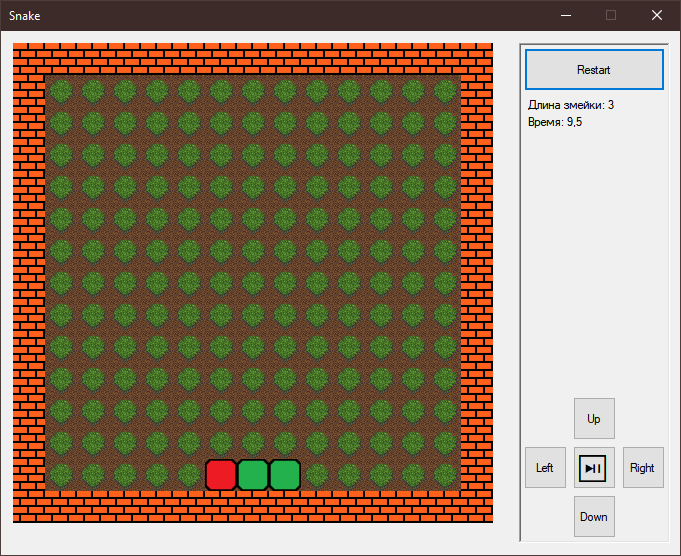


Рис. 8. Результат проверки функции “Growing”

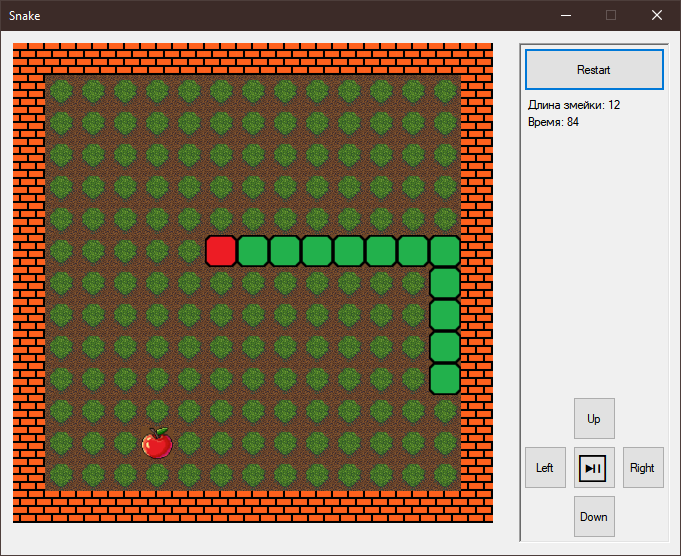
Настало время проверить столкновение змейки с хвостом. Накормим змейку яблоками. 

Рис. 9. «Откормленная» змейка

Теперь направим ее на свой хвост.

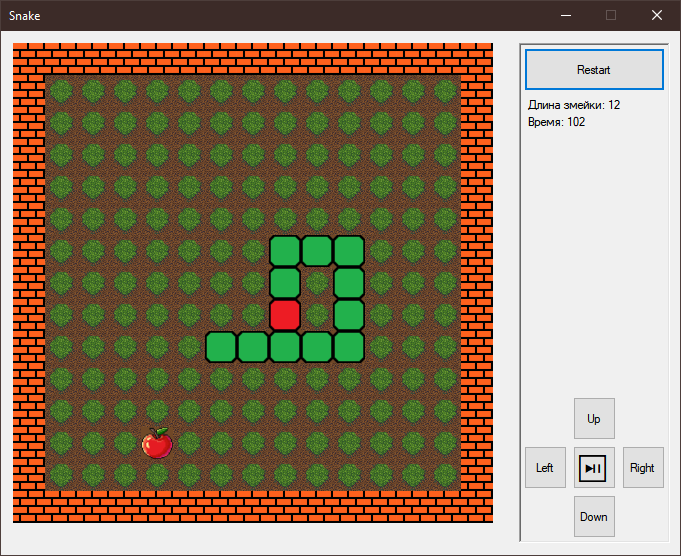


Рис. 10. Проверка на откусывание хвоста

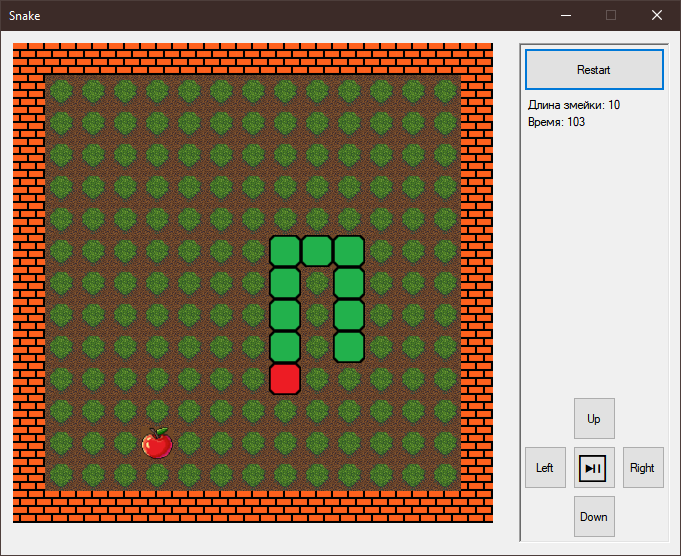


Рис. 11. Результат проверки на откусывание хвоста

В результате были откушены 2 сегмента. Проверка пройдена.

# Заключение

В ходе курсовой работы был создан продукт, демонстрирующий приобретённые мною навыки при работе с C++/CLI. Опираясь на уже существующие проекты, обозначенные в части «Анализ предметной области», была написана программа, заимствующая определённые механики, улучшающая некоторые из них и вводящая новые. После чего для проекта был проведён тест, в результате которого ошибок обнаружено не было.

С полным кодом программы можно ознакомиться в репозитории GitHub: <https://github.com/Hurrinecane/Snake_Cpp_CLL>.

# Список литературы

1. Habr.com «История: «Змейка» — самая популярная мобильная игра на границе тысячелетий / Блог компании Microsoft Lumia / Хабр», 2012,

— Режим доступа: <https://habr.com/ru/company/microsoftlumia/blog/136629/>

1. ru.wikipedia.org «Snake (игра) — Википедия», 2020,

— Режим доступа: <https://ru.wikipedia.org/wiki/Snake_(игра)>

1. github.com «Hurrinecane/Snake\_Cpp\_CLL: Snake», 2020,

— Режим доступа: <https://github.com/Hurrinecane/Snake_Cpp_CLL>