Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский экономический университет имени Г.В. Плеханова»

**Московский приборостроительный техникум**

Индивидуальный проект

На тему: Разработка игрового приложения «Simple Tamagotchi».

Дрюпина Андрея Александровича

обучающегося 1 курса группы ИCиП-2-18

специальность 09.02.07 "Информационные системы и программирование"

Руководитель: / Р.Ю. Волков /

«14» июня 2019 г.

Студент: / А.А. Дрюпин /

«14» июня 2019 г.

2019

Оглавление

[1. Введение 3](#_Toc9860047)

[2. Цель разработки 3](#_Toc9860048)

[3. Средства разработки 3](#_Toc9860049)

[4. Постановка задачи 3](#_Toc9860050)

[5. Тестирование и отладка 4](#_Toc9860051)

[6. Результаты работы программы 6](#_Toc9860052)

[7. Заключение 7](#_Toc9860053)

[8. Список использованных материалов 7](#_Toc9860054)

## Введение

Играть в компьютерные игры довольно увлекательно, но еще увлекательней самому разрабатывать их. Именно поэтому в качестве итогового проекта я выбрал разработку собственной компьютерной игры. Знания, полученные в процессе разработки, стали бесценным опытом в программировании, которые можно использовать и в других моих разработках. В игре присутствуют функции, которые я детально проработал, и теперь их можно переносить и в другие проекты, тем самым экономя ресурсы и время.

## 2. Цель разработки

Приложение разработано для достижения следующих целей:

1. Повышение качества исходного кода.
2. Самоконтроль знаний, полученных на занятиях по предмету «Информатика».
3. В качестве итоговой практической работы по предмету «Информатика».

## 3. Средства разработки

1. Компьютер на базе ОС Microsoft Windows 10 (Сборка 17763.529, 64 бита)
2. Набор стандартной периферии ПК (Монитор, клавиатура, мышь)
3. Интерактивная среда разработки Pascal ABC.NET

## Постановка задачи

Сценарий игры:

Вы обладатель компьютерного существа - Тамагочи, за которым необходима постоянная забота и уход. Его нужно кормить, лечить, играть с ним и т.д., чтобы он мог виртуально жить и стать взрослым. Если Вы окружите животного надлежащим уходом, то увидите, как ваш питомец растёт каждый день, становясь все больше и красивей.

Описание основных игровых функций:

1. Кормление. Можно проверить на дисплее, насколько голоден ваш любимец, а затем решить, стоит ли его кормить или нет. Помните, что слишком много еды может привести к потере настроения и здоровья Тамагочи.
2. Развлечение. Это самое приятное занятие для вашего Тамагочи. В игре Вы получаете очки счастья, которые показывают уровень настроения Вашего питомца.
3. Медицина. Когда Ваш Тамагочи плохо себя чувствует, он дают об этом знать уменьшением очков здоровья. И тогда Вы используете аптечку первой помощи, содержащую лекарства от всех болезней.

Важно: Вы всё время должны проверять жизнедеятельность любимца. Голодный, неухоженный Тамагочи становятся несчастным, начинает болеть, и в конечном итоге умирает, если вовремя не принять меры.

Управление:

* Кнопка H – Использовать аптечку
* Кнопка E – Покормить питомца
* Кнопки A, S, D, F, G – Поиграть с питомцем
* Кнопка N – Новая игра

Системные требования:

* Процессор с тактовой частотой 1200 MHz или выше
* ОС Microsoft Windows XP или выше
* Оперативная память 256 Мб или больше
* Свободное место на жёстком диске от 80 Мб или больше

Используемые переменные, процедуры и функции:

* Health – Переменная для хранения уровня здоровья.
* Hungry – Переменная для хранения уровня голода.
* Happy – Переменная для хранения уровня счастья.
* Age - Переменная для хранения возраста питомца.
* Die - Переменная для хранения смерти питомца.
* Time - Переменная для хранения времени.
* MainTimer() – Процедура основного таймера.
* HealthTimer() – Процедура для выполнения действий по времени для переменной “Health”.
* HungryTimer() - Процедура для выполнения действий по времени для переменной “Hungry”.
* HappyTimer() - Процедура для выполнения действий по времени для переменной “Happy”.
* KeyPressing(key: integer) – Процедура обработчика кнопок.
* CloseWindow() – Процедура обработчика системных событий.
* GetTime() – Функция, возвращающее системное время в секундах (Timestamp).
* WriteSettings() – Процедура записи переменных в файл.
* ReadSettings() - Процедура считывания переменных из файла в программу.
* DrawAnimal() – Процедура для вывода графического представления домашнего питомца.
* DrawDie() – Процедура для вывода графического представления смерти питомца.
* DrawLevels(var Health, Hungry, Happy, Age: integer) - Процедура для вывода графического представления уровней жизнеобеспечения питомца.

## 5. Тестирование и отладка

Таблица 1. – Тест переменных. Функциональное тестирование.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № п/п | Значения переменных | | | | | Выходные данные | Примечание |
| Health | Hungry | Happy | Die | Time | Уровни, сообщения, графическое представление |  |
| 1 | 5 | 5 | 5 | 0 | 0 | Уровни состояния питомца полные. | «Нулевое» состояние, характерное для новой игры. |
| 2 | 0 | 5 | 5 | 0 | 0 | Уровень здоровья снизился до нуля. Питомец умер. |  |
| 3 | 5 | 0 | 5 | 0 | 0 | Уровень голода снизился до нуля. Питомец умер. |  |
| 4 | 5 | 5 | 0 | 0 | 0 | Уровень счастья снизился до нуля. Питомец теряет очки здоровья и голода в два раза быстрее. |  |
| 5 | 5 | 5 | 5 | 1 | 0 | Питомец умер.  Отрисовка смерти. |  |
| 6 | 5 | 5 | 5 | 0 | 1560108538 | Снижение всех уровней. | Проверка времени. Получение timestamp’а, расчет разницы и вывод результата в переменные. |

В процессе отладки использовались стандартные средства интерактивной среды разработки Pascal ABC.NET (точки останова, браузер глобальных переменных, консоль).

При отладке возникли сложности с файловой системой, т.к. встроенные функции очень требовательны к типу сохраняемых данных в файл.

Дополнительные сложности появились при получении timestamp’а, т.к. в языке Паскаль нет встроенной функции на получение времени в секундах. Для выхода из сложившейся ситуации, я разработал собственную функцию, производящую расчеты времени.

## 

## 6. Результаты работы программы

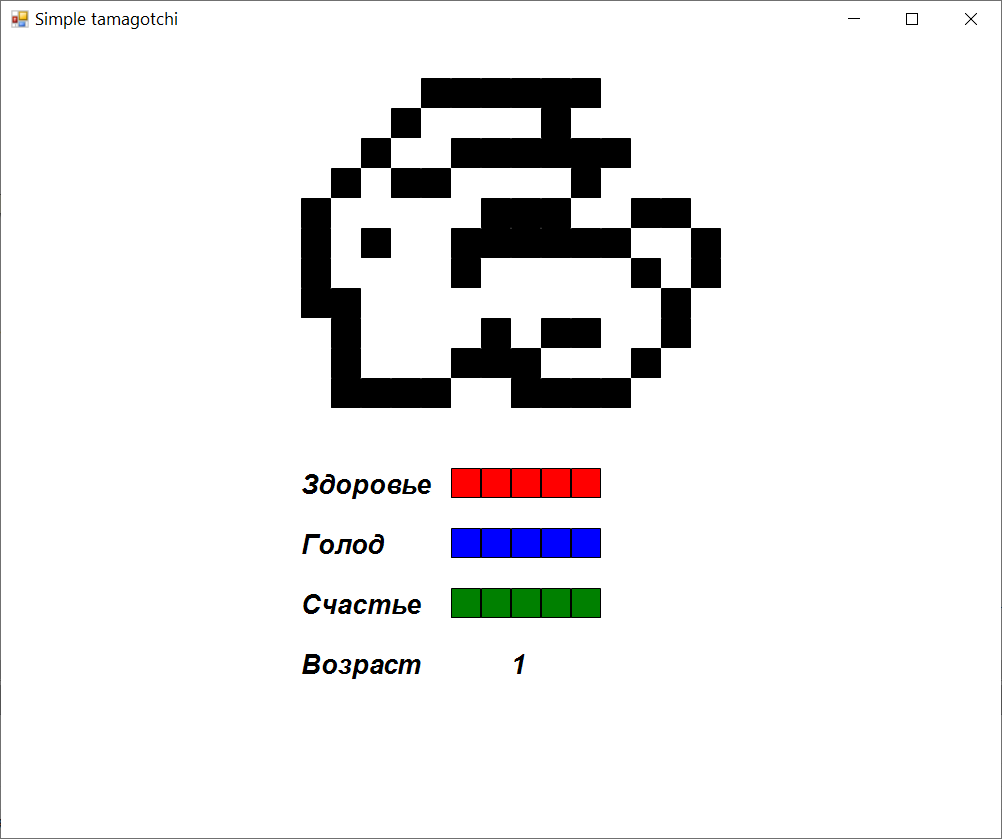


Рис. 1 «Нулевое» состояние питомца. Новая игра.

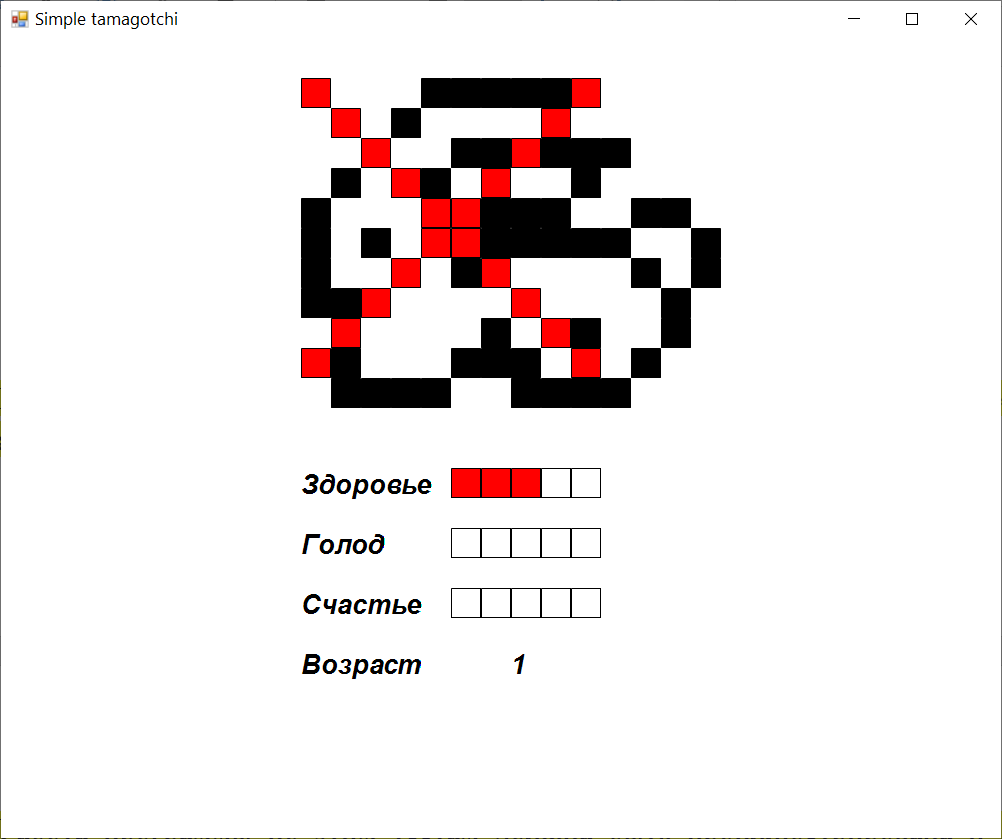


Рис. 2 Смерть питомца из-за потери всех очков еды.

Результаты:

* Выполнен минимальный объем требований к программе (Программа/игра на языке Паскаль).
* Точный подсчет времени и очков.
* Минималистичный дизайн и звуковое сопровождение.
* Игра повторяет “минимальные требования” любого домашнего животного.

## 7. Заключение

Во время работы над приложением я научился:

* Работать в интерактивной среде разработки Pascal ABC.NET
* Работать с файловой системой
* Работать с системной датой и временем
* Создавать отдельные модули приложения(Unit)
* Использовать процедуры и функции

Программа получилась вполне работоспособной, но в ней еще много недостатков, ошибок и уязвимостей. Все проблемы будут устраняться по мере возможностей.

Для повышения качества графической составляющей и функциональности планируется переход на другой язык программирования и подключение игрового движка. Планируется добавление модуля офлайн-платежей для ускорения развития питомца.

## 

## 8. Список использованных материалов

1. Конспекты занятий по Информатике и Введению в специальность
2. Сайт <http://labs-org.ru/pascal-12/> - «Работа с файлами в Паскале»
3. Сайт <http://sch61.ru/index.php/education-activity/programing/pascal/317-pascal-date-time> - «Pascal - дата и время»