

Задание 1. Интерактивная 2D-графика

Аннотация

Цель выполнения задания - освоить основы разработки интерактивных графических приложений и работы с изображениями. Материал подлежащий освоению:

- работа с изображениями - наложение с использованием альфа-смешивания, фильтры;
- работа с оконной системой и буфером кадра;
- обработка пользовательского ввода/вывода.

В качестве интерактивного графического приложения необходимо реализовать простую 2D-игру в жанре roguelike в соответствии с вариантом.

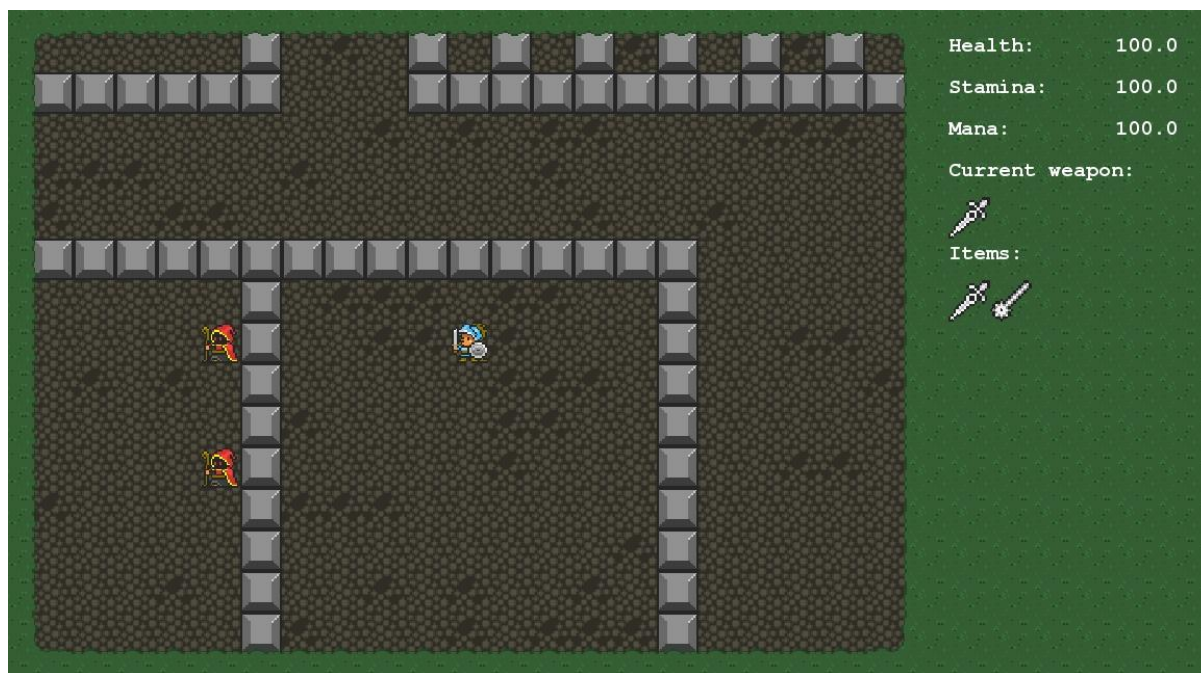


Рис. 1. Пример выполненного задания

Как определить свой вариант

Вариант складывается из двух компонентов.

1. Четный номер в списке группы - вариант А, нечетный номер - вариант Б.
2. (Порядковый номер второй буквы фамилии в алфавите) $\text{mod } 4 = \text{вариант } 0, 1, 2 \text{ или } 3$.

Например:

Иванов Иван, 8 в списке группы.

8 в списке \Rightarrow вариант А. Вторая буква фамилии - "в", третья в алфавите $\Rightarrow 3 \text{ mod } 4 = 3$

Итого - вариант А3.

Лисова Алиса, 13 в списке группы.

13 в списке => вариант Б. Вторая буква фамилии - "и", десятая в алфавите => $10 \bmod 4 = 2$

Итого - вариант Б2.

Базовая часть (10 баллов)

Вариант А. "Классический" rogue-like

Игроку необходимо выбраться из лабиринта, состоящего из нескольких (минимум 2-х) уровней. Обязательными элементами задания являются:

- Чтение карты лабиринта из текстового файла.
- один из нескольких вариантов:
 - Базовый набор:
 - пустое пространство: ' ' (пробел)
 - стена: '#'
 - пол: '.'
 - игрок: '@'
 - выход с уровня: 'X'
 - По вариантам:
 - **(0 вариант)** стены, которые игрок может разрушить: '%'
 - **(1 вариант)** ловушки в полу (например, шипы, выскакивающие из-под пола): 'T'
 - **(2 вариант)** двери/решетки, которые можно открыть/закрыть: 'D'
 - **(3 вариант)** пол, по которому можно пройти ограниченное количество раз (после чего он разрушается): ','
- Визуализация карты при помощи графики с обязательным отображением всех игровых элементов разными изображениями (тайлами/спрайтами).
- Реализация движения и взаимодействия с окружением игрока при помощи управления с клавиатуры (например, W, A, S, D, пробел). Игрок не должен проходить сквозь стены.
- При попадании в пустоту/ловушку игрок должен умирать: с помощью графики необходимо вывести сообщение о проигрыше и завершить игру.
- Если игрок достиг выхода с уровня, необходимо при помощи графики вывести сообщение о переходе на следующий уровень, либо об успешном окончании игры, если это был последний уровень.

Вариант Б. Rogue-like с лабиринтом из комнат (на подобие Hades, Binding of Isaac, подземелий в Legend of Zelda: A link to the past etc.)

Игроку необходимо выбраться из лабиринта, состоящего из соединенных между собой квадратных/ прямоугольных комнат нескольких разных типов. Из очередной комнаты игроку доступно от 1 до 4 выходов (по одному возможному выходу в каждой из стен).

Обязательными элементами задания являются:

- Чтение карты общего лабиринта и карты комнат разных типов из текстовых файлов. Лабиринт состоит минимум из 20 комнат 4 разных типов. Одна комната целиком помещается на экран.

- Каждый символ в текстовом файле общего лабиринта задаёт один из нескольких типов комнат - 'A', 'B', 'C', ...:
- Каждый символ в текстовом файле для комнаты задает один из нескольких вариантов тайла:
 - Базовый набор:
 - пустое пространство: ' ' (пробел)
 - стена: '#'
 - пол: '.'
 - игрок: '@'
 - выход из комнаты: 'x'
 - выход из всего лабиринта: 'Q'
 - По вариантам:
 - **(0 вариант)** сокровища, которые игрок может подобрать: 'G'
 - **(1 вариант)** ловушки в полу (например, шипы, выскакивающие из-под пола): 'T'
 - **(2 вариант)** ключи, которые позволяют открывать закрытые выходы из комнат: 'K'
 - **1**
- Визуализация карты при помощи графики с обязательным отображением всех игровых элементов разными изображениями (тайлами/спрайтами).
- Реализация движения и взаимодействия с окружением игрока при помощи управления с клавиатуры (например, W, A, S, D, пробел). Игрок не должен проходить сквозь стены.
- При попадании в пустоту/ловушку игрок должен умирать: с помощью графики необходимо вывести сообщение о проигрыше и завершить игру.
- Если игрок достиг выхода из лабиринта, необходимо при помощи графики вывести сообщение об успешном окончании игры.

Дополнительная часть (максимум - 15 баллов)

- Реализовать врагов, которые перемещаются по лабиринту (вариант А) или по комнате (вариант Б) - патрулируют по маршруту (2 балл) и/или двигаются на игрока (4 балла).
- Более продвинутые реализации врагов: до 4 баллов. Например, противники атакуют с некоторой дистанции и отбегают от игрока, при его приближении.
- Анимация статических объектов - например, сокровища и шипы ловушек блестят. (2 балла)
- Плавная спрайтовая анимация динамических объектов - походка игрока и врагов, открытие дверей и т.д. (от 2 до 5 баллов)
- Реализовать графический эффект перехода между уровнями (вариант А) и комнатами (вариант Б) - постепенное "угасание" и появление игровой карты (fade out / fade in), эффект "мозаики", плавное "перетекание" одного изображения в другое и т.п. (3 балла)
- Эффекты пост-обработки всего изображения - "дрожание" воздуха (heat haze), размытие/туман и т.п. (3 балла)
- Источники света - факелы, лампы и т.д., которые "освещают" (=изменяют цвет) соседних тайлов в некотором радиусе. (4 балла)

- анимация источников света (+1 балл)
- bloom (+4 балла)
- Реализация и графическое отображение инвентаря (3 балла)
- Графическое отображение характеристик игрока и соответствующие им игровые механики - например, если выводится здоровье, то игрок может его потерять (ловушки, враги) и, возможно, восстановить. (2 балла)
- Механика ближнего боя с анимацией. (2 балла)
- Механика дальнего боя (стрелковое оружие и/или магия - огненные шары, волшебные стрелы и т.д.) с анимацией летящего снаряда. (от 3 до 6 баллов в зависимости от субъективно оцениваемого визуального качества и сложности)
- Визуальные эффекты боя - “вылетающие” спрайты цифр повреждений, искры, “тряска” экрана и т.п. (от 2 до 4 баллов)
- Финальный босс игры: от 2 до 6 баллов. Боссы, отличающиеся от обычных врагов только числовыми параметрами(жизни, урон и т.д.), оцениваются в 2 балла. Чтобы получить больше необходимо реализовать хотя бы одну оригинальную механику для босса. Этой механики не должно быть ни у одного другого противника.
- Реализация большого лабиринта (вариант А): 5 баллов. В данном пункте предполагается, что карта уровня больше чем отображаемая в окне область. Необходимо реализовать отображение видимой области, которая располагается вокруг игрока и перемещается вместе с ним.

Шаблон для задания

Для выполнения задания предоставляется базовый шаблон, реализующий создание окна, вывод в него буфера кадра и обработку ввода с клавиатуры. Шаблон доступен в вариантах на языках C++ и Ada.

Для выполнения задания потребуется набор тайлов - маленьких изображений отдельных игровых элементов. Наборы тайлов можно скачать на одном из интернет-ресурсов или нарисовать самостоятельно в графическом редакторе. Некоторые ссылки:

- <https://itch.io/game-assets/free/tag-tileset>
- <http://pymapper.com/tile-downloads/>
- <https://pixanna.nl/materials/celiannas-tileset/>
- <https://craftpix.net/categorys/tilesets/>

Формат сдачи задания

Выполненное задание загружается в Google Classroom. Необходимо создать архив с исходными кодами, скриптами сборки (Cmake, GPR и т.п.) и всеми ресурсами (тайлы, карты в текстовых файлах и т.д.), название архива должно соответствовать формату: [номер_группы]_[Фамилия]_[номер_задания]. Например: 333_Ivanov_z1.7z

Вместе с архивом необходимо загрузить отчет в readme.pdf формате, содержащий перечисление реализованных пунктов задания со скриншотам, демонстрирующими их реализацию. ИЛИ текстовый файл readme.txt со списком реализованных пунктов и видео геймплея, в котором можно увидеть все реализованные пункты.

Требование по управлению программой:

- 1) управление движением должно осуществляться по клавишам WASD либо стрелочками;
- 2) меню инвентаря должно вызываться по клавише 'i';
- 3) Стрельба должна производиться по клавише 'q';
- 4) Дополнительное действие (например для открытия дверей) по клавише 'e';

Инструкция по инсталляции

```
git clone https://gitlab.com/vsan/msu_cmc_cg_2021.git
```

C++:

```
sudo apt-get install cmake  
sudo apt-get install libglfw3-dev
```

Ada:

```
sudo apt-get install gnat-gps
```

```
sudo apt-get install libxi-dev  
sudo apt-get install libxinerama-dev  
sudo apt-get install libxrandr-dev  
sudo apt-get install libxxf86vm-dev  
sudo apt-get install libxcursor-dev
```

В GPS на вкладке scenario для параметра Windowing_System выставить значение соответствующее вашей операционной системе (x11 для Linux).

Можно собирать, запуск вручную из папки bin

Можно запустить из меню в GPS