Требования

- На уровнях появляются двери (>), позволяющие выйти с этих самых уровней.
- Если игрок выходит с уровня, игра завершается победой.
- У уровня есть лимит шагов. Если его превысить, то игра завершается поражением.
- У персонажа теперь есть здоровье. При снижении здоровья до нуля игра завершается поражением.
- По завершению игры на экране показывается сколько собрано монет и сделано шагов.
- На поле присутствуют враги (E), которые могут преследовать персонажа.
- При столкновении игрока и врага происходит сражение, т.е. у каждого из них отнимается определённое количество здоровья.
- Индикаторы количества монет, текущего кол--ва шагов, максимально разрешенного кол-ва шагов и текущего уровня здоровья отображаются на экране.

- Игровая логика и рендер разделены
- Игровые сущности представляют из себя только набор данных, вся их логика вынесена в отдельные классы
- Вся логика по обработке данных находится в классах-системах (реализован паттерн ECS)

- CppLint отрабатывает при сборке проекта и не находит ни одной ошибки.
- Сборка производится при помощи make.
- В игре не должно быть платформозависимых функций, существующих не во всех компиляторах и не на всех платформах: srandom(), random(), localtime_r(), и так далее. В противном случае ваша игра попросту не соберется под разные платформы (Windows, Linux, macOS). Информацию об используемых функциях можно смотреть в Интернете.
- Во всех путях к файлам, с которыми вы работаете (если работаете), должны использоваться символы /. Этот формат используется в Linux, а Windows при необходимости автоматически преобразует их в нужный формат.

- Создать отдельную ветку, если у вас её всё ещё нет или с ней что-то случилось, например develop git checkout -b develop, вся работа у вас должна будет происходить в этой ветке.
- Опубликовать код на gitlab.7bits.it в вашем репозитории и создать merge requests, из develop в master.
- Коммиты в репозитории имеют информативные сообщения.
- Задание должно быть сдано на проверку в Google Classroom (с прикреплением ссылки на merge request).

Для сдачи задания необходимо выполнить всё вышеописанное и выбрать свободное время в этой таблице.

Дополнительно:

- В чистом ECS сущности представляют из себя только ID, все данные о сущности хранятся только в компонентах, которые связаны с этим ID

Рекомендуемые воркшопы

- WS 9. Паттерны
- WS 10. Entity-Component-System

Рекомендуемые лекции

- Что такое ECS и с чем его едят
- Паттерны #1. Лекция

Дополнительные ресурсы

- Habr: Шаблон проектирования Entity-Component-System — реализация и пример игры
- Исходники игры Арканоид от Ильи, в которой используется ECS
- Рассказ об использовании ECS и не только для разработки рогалика