

Чеклист для кода игры по ИиП

Бережное использование памяти

- виртуальные функции без виртуального деструктора.
- как вы бережно используете память компьютера в своей игре?
- доказать отсутствие утечек, например, с помощью vld.
- доказать отсутствие выхода за границы массивов (vld не поможет, но есть настройки проекта runtime checks).
- доказать отсутствие разыменований нулевого указателя.
- доказать отсутствие мусора в указателях.
- доказать отсутствие двойного удаления.
- доказать отсутствие обращения к памяти после удаления указателя.

Соблюдение контрактов функций

- проверить контракты для unit health (не может быть меньше 0)
- проверить контракты для математических функций (random, cartesian, т.д.)
- где есть спецификации диапазона допустимых значений аргумента функции?
- как проверяются допустимые значения аргументов функций?

Структурное программирование

- проверить отсутствие глобальных переменных
- проверить несколько функций на ясность выполняемой ей задачи
- проверить корректную работу с C-Style массивами, включая определение их размера без магических чисел

ООП (если рискнут)

- списки инициализации в конструкторе либо отказ от конструктора
- наличие деструктора, если надо, виртуального либо отказ от virtual функций
- корректное применение public/private либо полный отказ от них (struct вместо class)
- override у перегруженных функций либо отказ от virtual функций
- доказать корректное поведение в условиях исключения std::bad_alloc

Чеклист для предоставленных студентом материалов

- итеративный подход (коммиты в git) в локальном или remote репозитории
- минимум два роадмапа: на начало проекта и текущий
- документ: отчёт о курсовой
- видеофайл: с геймплеем и сутью игры
- документ: отчёт о работе за семестр
- документ: презентация по игре