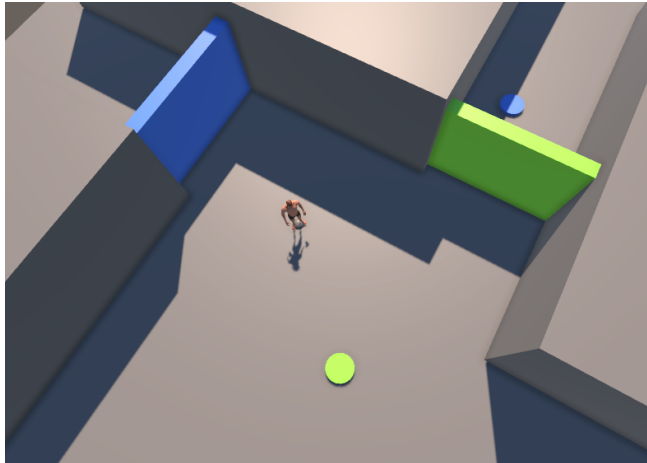


## Тестовое задание на Unity с EcsLite

*Цель данного тестового задания: разобраться и вникнуть в основы ECS, научиться правильно **декомпозировать** код на компоненты и системы.*

**EcsLite** можно скачать отсюда <https://github.com/Leopotam/ecslite>.



Зд, изометрия. Персонаж игрока перемещается по карте через point and click.

В сцене заранее расставлено произвольное количество пар дверей и **КРУГЛЫХ** кнопок. Каждая кнопка привязана к своей двери и открывает только ее.

Дверь открывается не мгновенно, а плавно пока игрок стоит на кнопке. Когда же игрок сходит с кнопки - дверь опять замирает.

Основные системы и компоненты, отвечающие за игровую логику **не должны зависеть от Unity**. Представьте, что этот код должен уметь выполняться на real-time сервере в среде, где нет Unity.

Это означает, что если взять большую (не все) часть этих компонентов и систем, отвечающие за логику и добавить их в **пустой консольный C# проект**, то он должен хотя бы компилироваться успешно.

Учтите, что:

- **можно** использовать мат. либу из Unity namespace (Vector3 и прочие) и не нужно делать свои обертки
- **не требуется** обход препятствий
- **не требуется** просчет столкновений
- **не требуется** использовать NavMesh (хотя бы потому что он завязан на Unity)
- не используйте EcsInject

На персонаже достаточно двух анимаций Idle и Move.

Прочее:

1. Укажите в **readme** количество часов в сумме потраченных на разработку.
2. Перечислите в **readme** список систем, которые могут и должны быть запущены на сервере и которых должно быть достаточно для полной **realtime** симуляции логики игры.
3. Проект необходимо заливать в публичный **github** репозиторий, много сделанных **commits** по ходу разработки - приветствуется.
4. Приветствуется использование **Zenject**.
5. Пожалуйста, складывайте все компоненты в одной общей папке **Components**, а все системы в своей одной общей папке **Systems**.
6. Складывайте, пожалуйста, все зависимости в проект, чтобы он открывался и запускался без лишних манипуляций и потенциальных ошибок.

Важно:

Не торопитесь отправлять работу на ревью. Прежде, чем это сделать убедитесь, что пункт **2** выполним с вашим набором систем и компонентов.