

Плюсы и минусы структур данных.

- Динамический массив — List, HashSet...

Обычный массив ограничен жёсткими рамками заданного размера, что не всегда удобно. Не всегда можно при задании или определении массива знать нужный ему размер.

Динамический массив в этом случае удобней, так как в него по мере необходимости можно добавлять новые элементы. Например, если в игре генерируется произвольное — в заданных рамках — количество врагов, и нужен массив врагов, то удобнее работать с динамическим массивом, чем пытаться предугадать нужный размер обычного массива с вероятностью вылететь за предугаданные пределы в случае каких-либо изменений.

Различные виды динамических массивов дают широкие возможности их использования.

С другой стороны из-за неопределённого размера массива, объём занимаемой им памяти больше, чем у обычного массива. Соответственно и операции с ним занимают больше ресурсов. Так же при пополнении динамического массива в определённых моментах добавления дополнительного элемента станет достаточно дорогостоящей операцией — когда исходный объём выделенной на массив памяти подойдёт к концу и будет расширяться.

- Очередь

Удобна при работе с объектами, где важен порядок, и работать нужно с первым элементом. Например, если разрабатывается игра с картами, то после перетасовывания колоды или сброс, взаимодействие идёт только с верхней картой. И удобно реализовывать массив через очередь. Для удобства и для уменьшения шансов ошибок.

Возможностей работы с элементами внутри очереди на порядки меньше чем у обычного массива.

- Стек

Позволяет создавать условную «стопку» объектов. Например, стек из авто сохранений. Так удобно пополнять его, чистить и сразу брать нужное сохранение, когда оно надо.

Минусы те же, что и с очередью.