Optimization.

What's optimization?

Split into 4 groups

POTATO PEELER MACHINE

Group 1:

- Design a potato peeler that **peels as many potatoes as possible in a short time.**

Group 2:

- Design a potato peeler that **produces as little waste as possible while peeling**

Group 3:

- Design a potato peeler that **is as cheap as possible to build and operate**

Group 4:

- **Figure out what on earth is the point of this exercise.**

- Descriptions should be single sentences, nothing too complicated.

- Looking for concepts, not finished designs!

- Go!

- What do we get if we swap optimization targets now?

- Throughput → less waste

- Less waste → cheap

- Cheap → throughput

- (etc)

- Speed (i.e, frame rate).
- Size:
  - Memory limits.
  - Storage media limits (also: download time).
  - Bandwidth limits.
- Fun.
- Other criteria too!

- Measure.

- Find bottlenecks.

- Do things more efficiently.

  - Better algorithms.

  - Hardware-friendly implementation (assembly).

  - Or most commonly, don't do things if possible.

FRAMES PER SECOND?

FRAMES PER SECOND?

- FPS is common, but confusing.
  - "I optimized this routine to be 2 FPS faster"

- mspf is better
  - "I optimized 5ms per frame from this routine"

- Routines to get current time in ms:
  - Win32:GetTickCount, SDL:SDL_GetTicks

- To optimize memory usage, you need to be able to measure it.
  - Commonly, this means writing your own memory manager.
    - ..here be dragons..
  - Alternatively, look at your code, figure out what takes space, and see what you can do about it.
- On desktop windows, not usually a problem (until your project grows..)
  - Task manager can show rough memory use

- Disk space, download size, etc.

- Easy to measure.

- Relatively easy to optimize.

  - Compress.

  - Generate.

  - Leave out.

- Again, measure.

- Some optimization tactics:

  - Leave data out (predict, extrapolate)

  - Encode more efficiently (arithmetics)

  - Compress

- Test.
- Test.
- Test.
- Let other people play, *listen to their comments.*
- Test.
- Test.
- Test.
- Test.
- Test.
- Test.

OTHER CRITERIA

- Tradeoffs.

    - Space vs time.

    - Everything vs development time.

- AI optimization.

    - How can me make Bob find Alice sooner?

- Usability optimization.

    - Too much text, confusing user interfaces..

- "Premature optimization is the root of all evil"
  - Donald Knuth

- When writing code, optimize for readability, not speed!
  - You'll end up spending most of your time re-reading your own code, so better make it understandable.