

Games AI

Lecture 6.1

Automated Game Testing

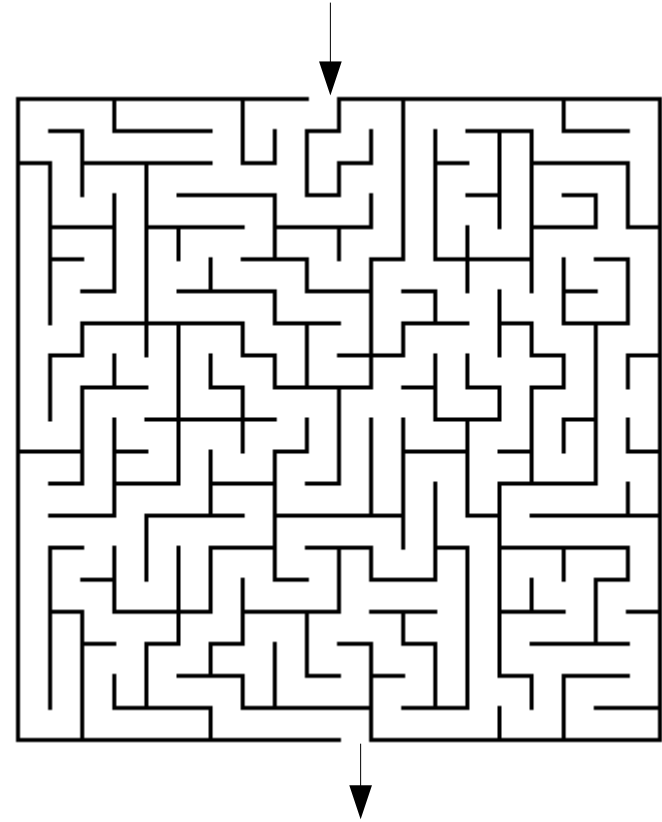
- Last lecture we talked about **Procedural Content Generation**
 - A PCG algorithm has a **generative space** of content that it can create
 - We want our content to be **functional** and **aesthetic**.
 - Among the other features we wanted from our generator were:
 - Reliability
 - Controllability
 - Diversity
 - Creativity

- **Iteration** is important in developing good PCG
 - But how do we iterate?
 - 1) Generate some content
 - 2) Look at it**
 - 3) Tweak the generator
 - 4) Repeat

- Look at it
 - But what if that takes too long?



- Look at it – **with AI**
 - Is this maze passable?
 - How could we check automatically?



- **More challenging...**
 - Is this level passable?
 - How could we check automatically?



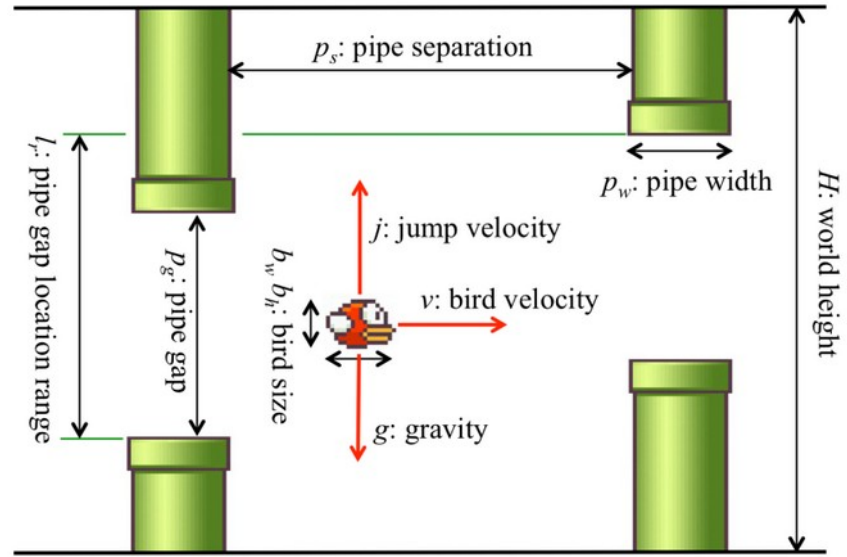
- **Iteration (2)**
 - 1) Generate some content
 - 2) AI gives it a score**
 - 3) Tweak the generator
 - 4) Repeat

- **AI gives it a score**
 - But what if we want a particular aesthetic?
 - How do we give this a score?



- **Iteration (3)**
 - 1) Generate some content
 - 2) Designer (efficiently) gives it a score**
 - 3) Tweak the generator
 - 4) Repeat

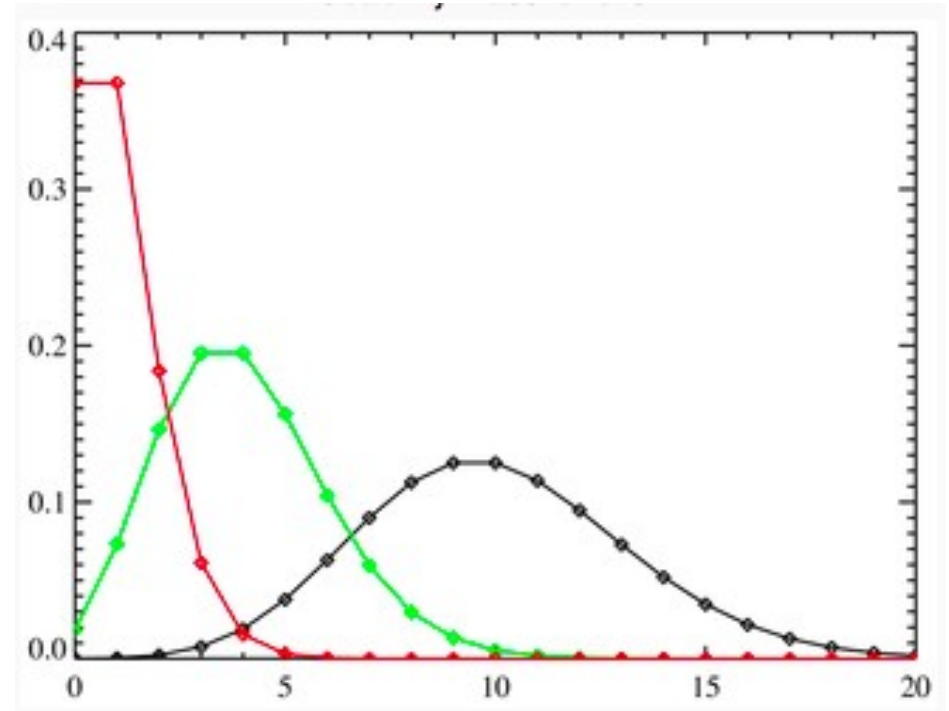
- **More difficult...**
 - But what if we want a particular aesthetic of play
 - e.g. how do we score difficulty?



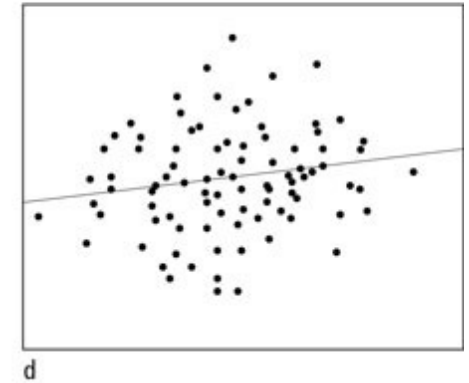
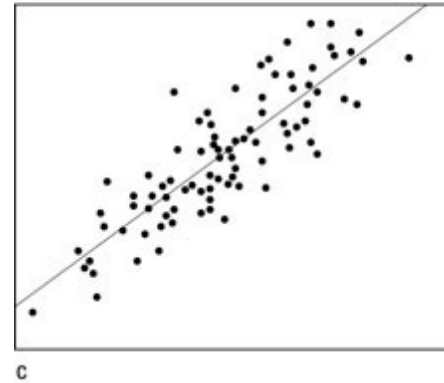
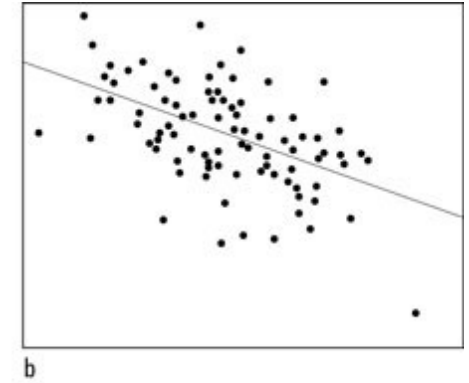
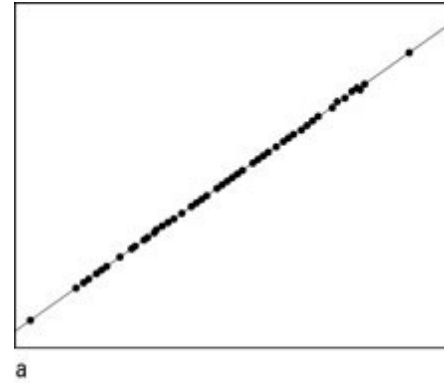
- **Iteration (4)**
 - 1) Generate some content / **Build a version of the game**
 - 2) AI or Designer (efficiently) gives it a score
 - 3) Tweak the generator / **develop the next version of the game**
 - 4) Repeat

- We also wanted our PCG to have:
 - Reliability
 - Controlability
 - Diversity
 - Creativity
- If we can **devise a statistical measure** for these, then we can evaluate them automatically

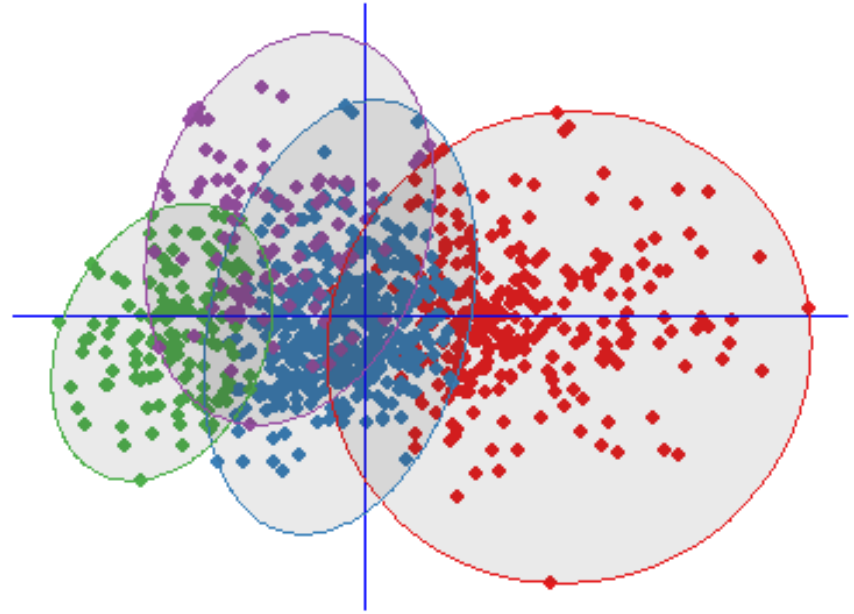
- How would we generate a statistic for:
 - **Reliability?**



- How would we generate a statistic for:
 - **Controllability?**



- How would we generate a statistic for:
 - **Diversity?**



- **Iteration (4)**

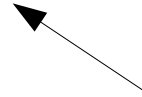
- 1) Generate some content

- 2) Score it

- 3) Tweak the generator **based on that score**

- 4) Repeat

Still requires human intervention



- To iterate fastest, we need the **AI to re-program** the generator
- We need the generator to... **evolve!**

