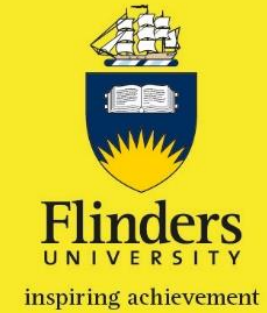
COMP3752 Computer Game Development



3D Tower Defence

Peter Mitchell

Topics Today

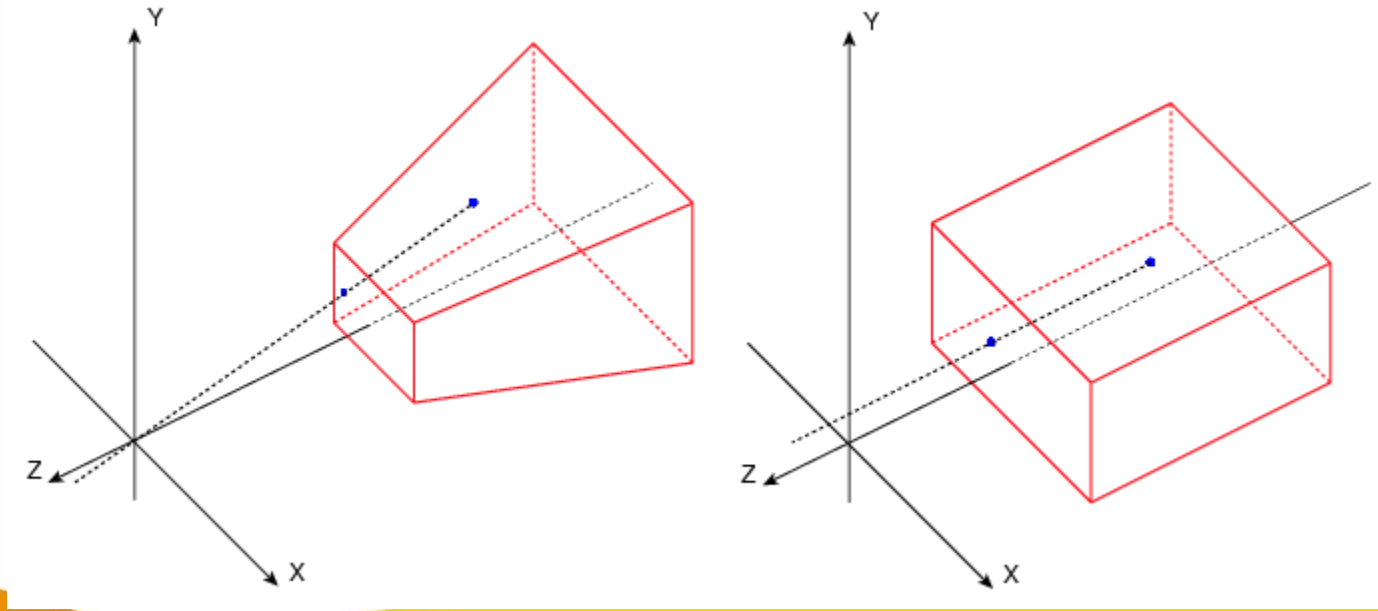
- Tower Defence Game
  - Structure of a Tower Defence, Camera Movement, State Machines, AI Behaviours, Object Management Suggestions.
- 3D Character Controller
- Events
- GameObject Composition (Single Responsibility Principle)
- Chad's Challenge (My 2011 XNA based Group Project)
- Procedural 3D Grid Generation

Structure of a Tower Defence

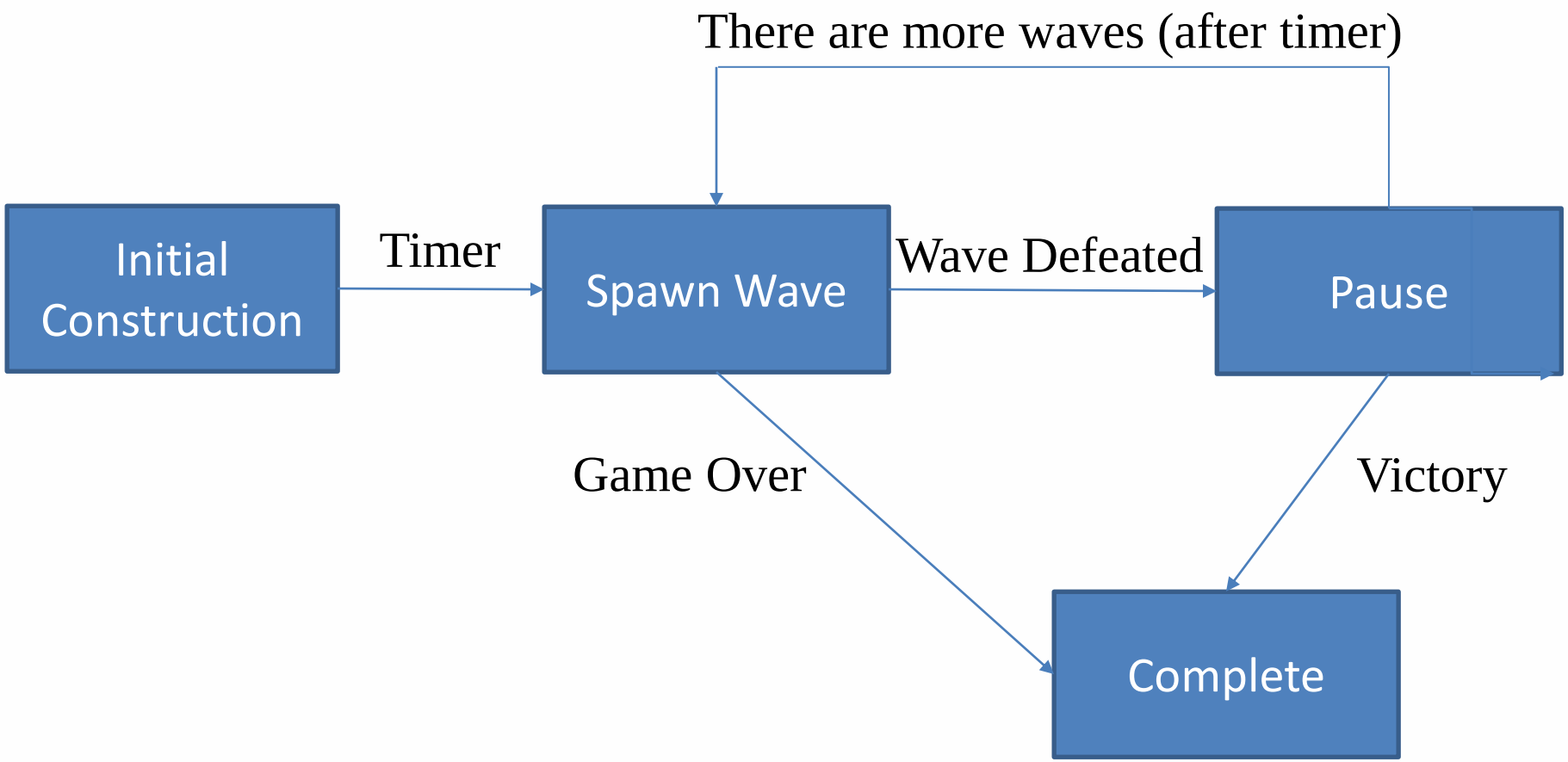
- Towers (Projectiles, Tower Placement)
  - Single target, rapid fire, area of effect, slowing
- Enemies (Wave Spawning, Following Path, Unit Death or Player Failure)
  - Normal enemy, fast enemy, dangerous enemy, mini boss, boss
- Game Flow
  - Initial Construction -> Spawn Wave -> Intermission -> Repeat Spawn Wave/s and Intermission -> Game End

Camera Movement

- transform.position (x,y,z movement)
- transform.localEulerAngles (x,y,z rotation)
- Restricting bounds can be applied with Mathf.Clamp(v, min, max)
- Perspective Camera (left) vs Orthographic (right)



State Machines



AI Behaviours

- Enemies: Follow Path
  - Follow nodes until teleporter or death
- Towers: Find targets in range and shoot

Object Management Suggestions

- Object Pooling (reuse of objects)
- Sensible Hierarchy Object Spawning (attach to a parent)
- Object Reference Caching

3D Character Controller

- CharacterController instead of Rigidbody Component

Events

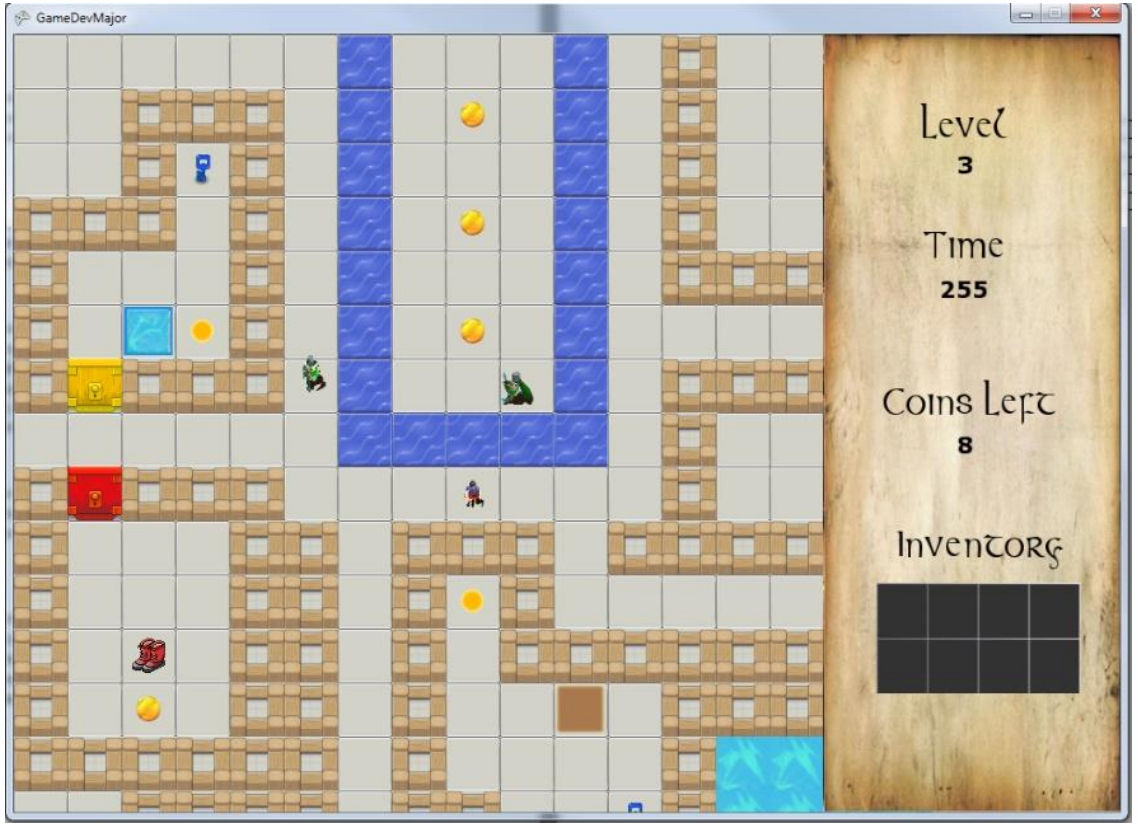
- C# and Unity variations
- Use Unity Events for Unity UI and for situations where you want to set events up in the inspector. Otherwise use the C# version.
- Events work by registering actions to an event. Then when something happens you tell the event to invoke causing any registered actions to be all executed.

GameObject Composition

- Single Responsibility Principle: Each component should have a distinct job that is focused on one job.
- Can use Events to decouple the requirements for components to pass important messages.

Chad's Challenge

- Game Dev project that I lead development on in 2011 version of the topic using XNA.
- A game mimicking the game Chip's Challenge.
- Features a level editor that was used to make all 10 levels.



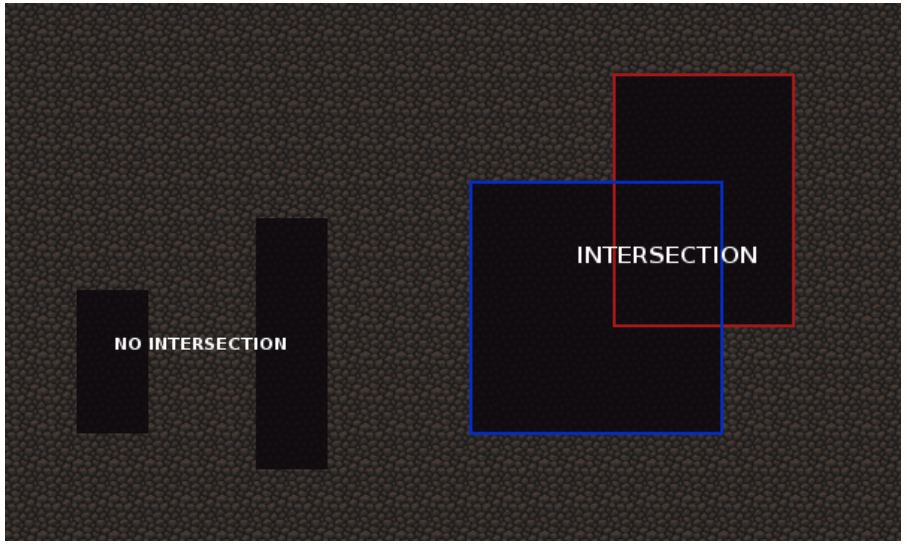
Procedural 3D Grid Generation

- Demo generates rooms that do not overlap and joins to other rooms by creating corridors.
- A Poisson Disk Sampling is then used to place objects over the grid evenly and each object’s position is validated to only spawn inside the rooms.

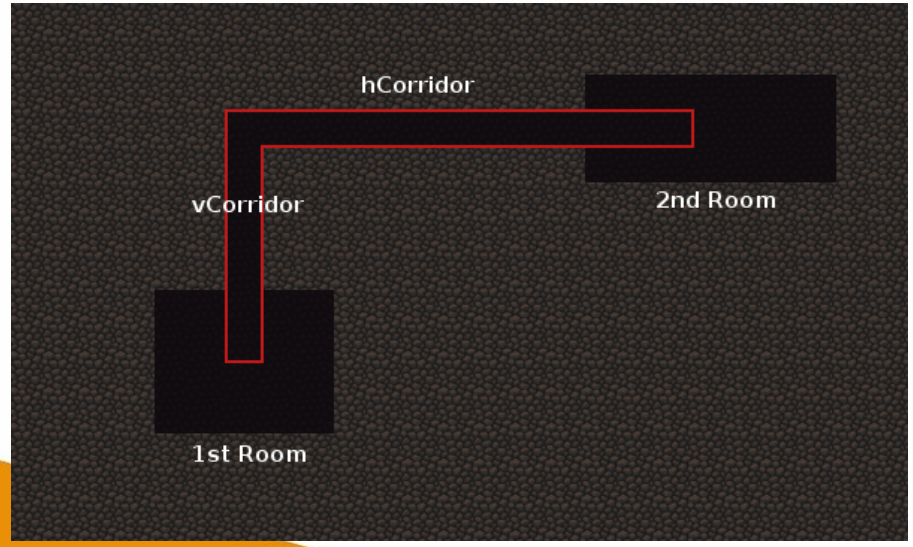
Map Generation with Rooms linked via Corridors

A

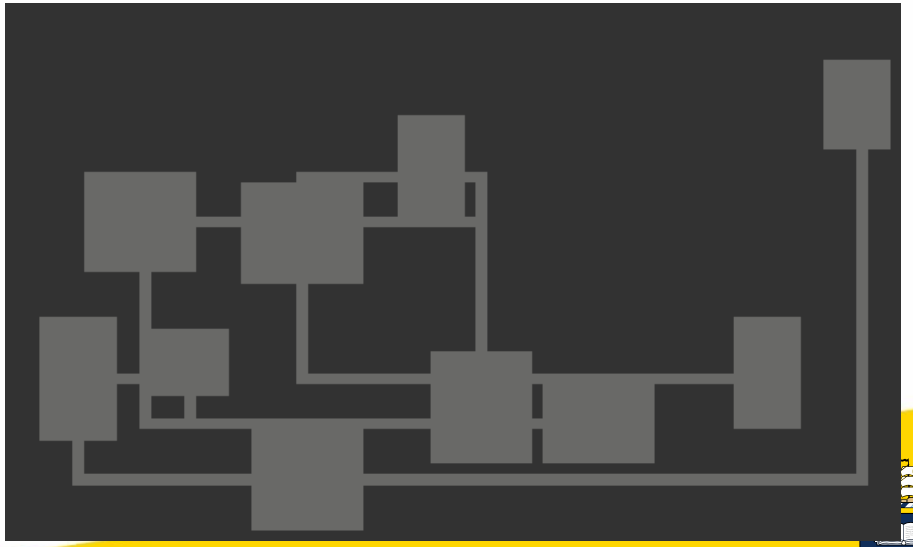
B



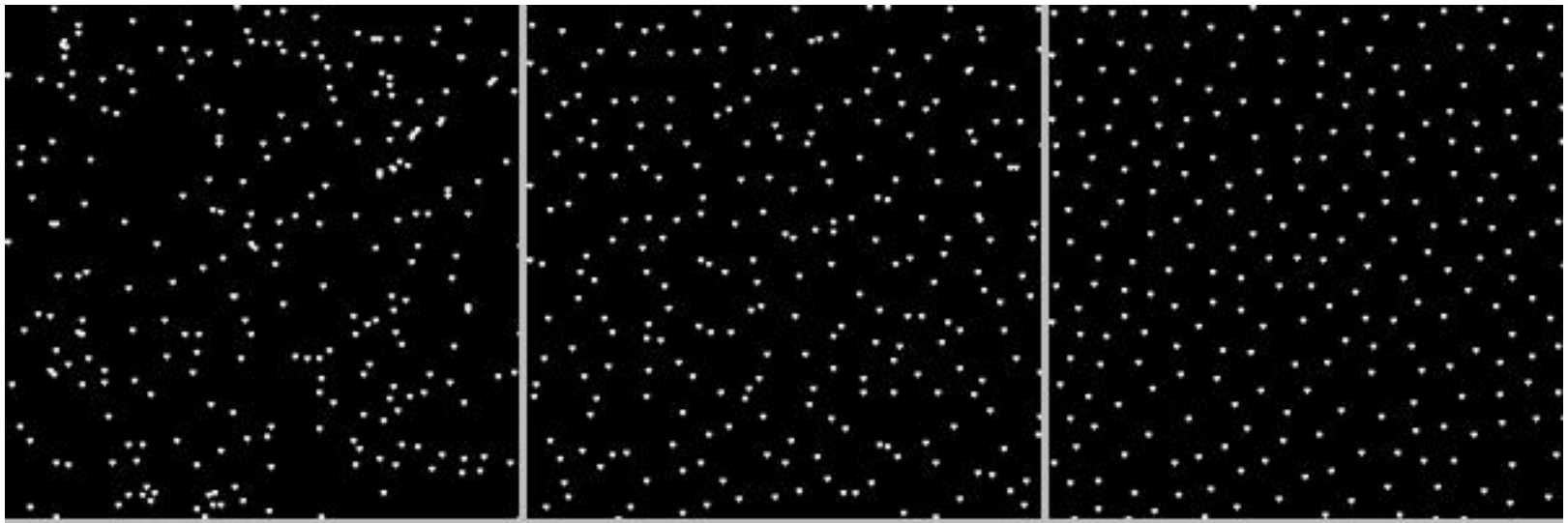
C



D



Random Distributions



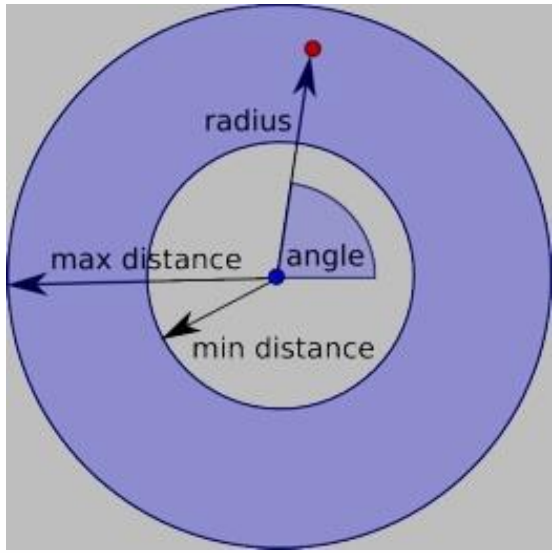
Uniform Random Points. The x and y coordinates of these points have been chosen randomly within the width of the image.

Jittered Grid. The image is divided into a grid, and one point is randomly selected from every cell in the grid.

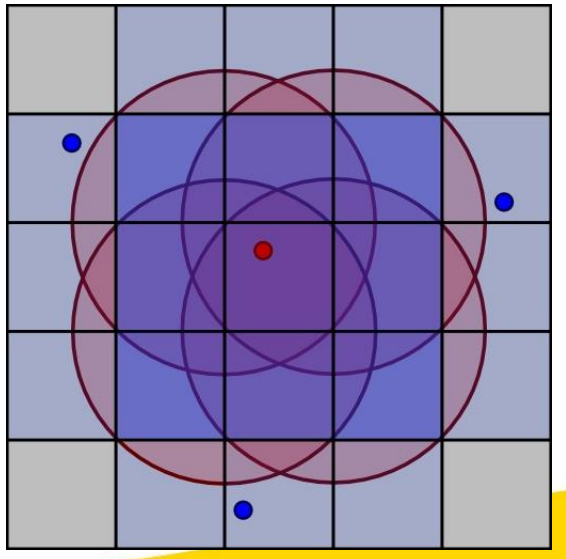
Poisson disc sample points

Poisson Disk Sampling

- Generate Sample Point:



- Check neighbours for conflicts:



Workshop Next Week

- Next week will be the last week I’m running the Unity workshops.
- If there are particular topics you would like some insight into let me know ASAP and I will try to cover them.