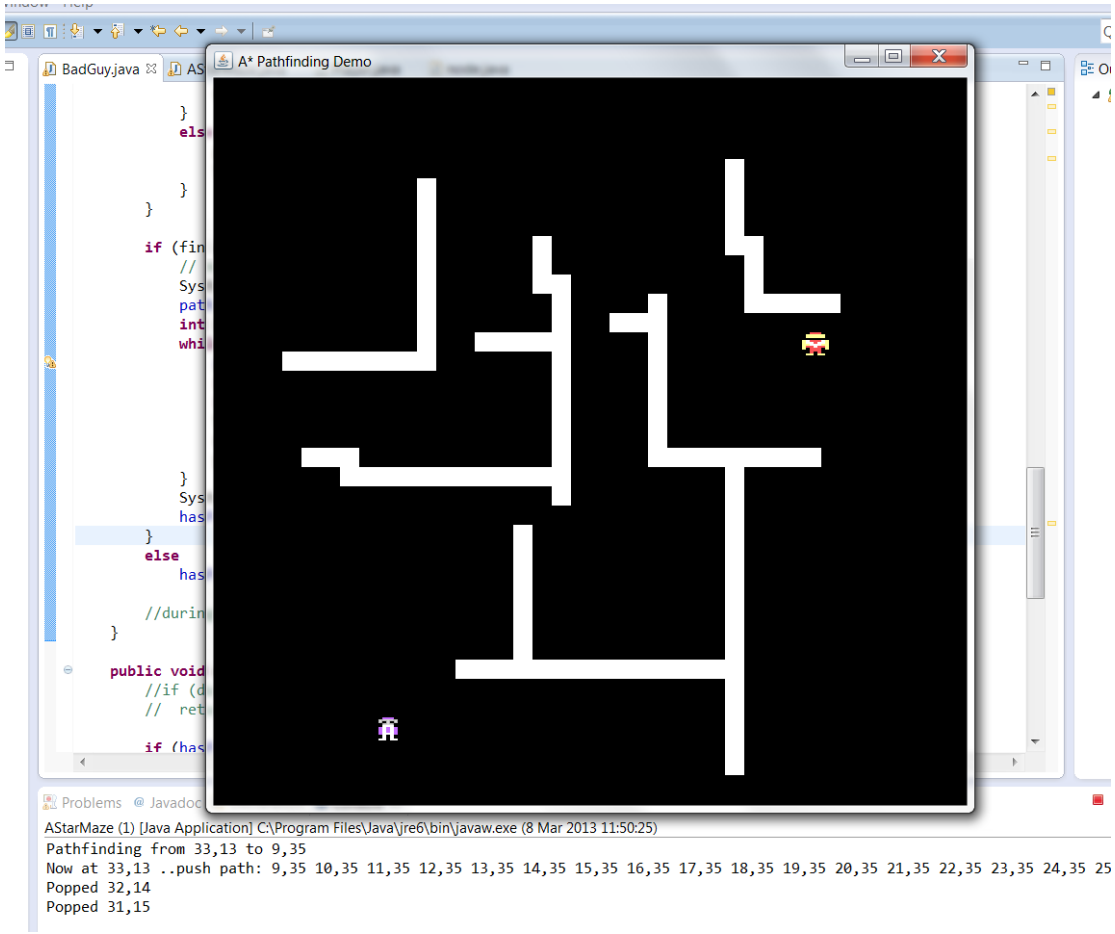
CT255
NGT2

Week 10
[2D Games in Java]

Dr. Sam Redfern
sam.redfern@universityofgalway.ie

Last Week's Assignment (A*)

- Download base code for 'badguy chases the player' game
- This provides maze drawing, loading, saving
- It also moves the player with arrow keys, and badguy moves according to a dumb 'straight line' chase path – stops at walls
- Your goal is to implement A* pathfinding to make the badguy chase more effectively
- The A* path should be recalculated whenever the player moves or the maze is modified



Base code:

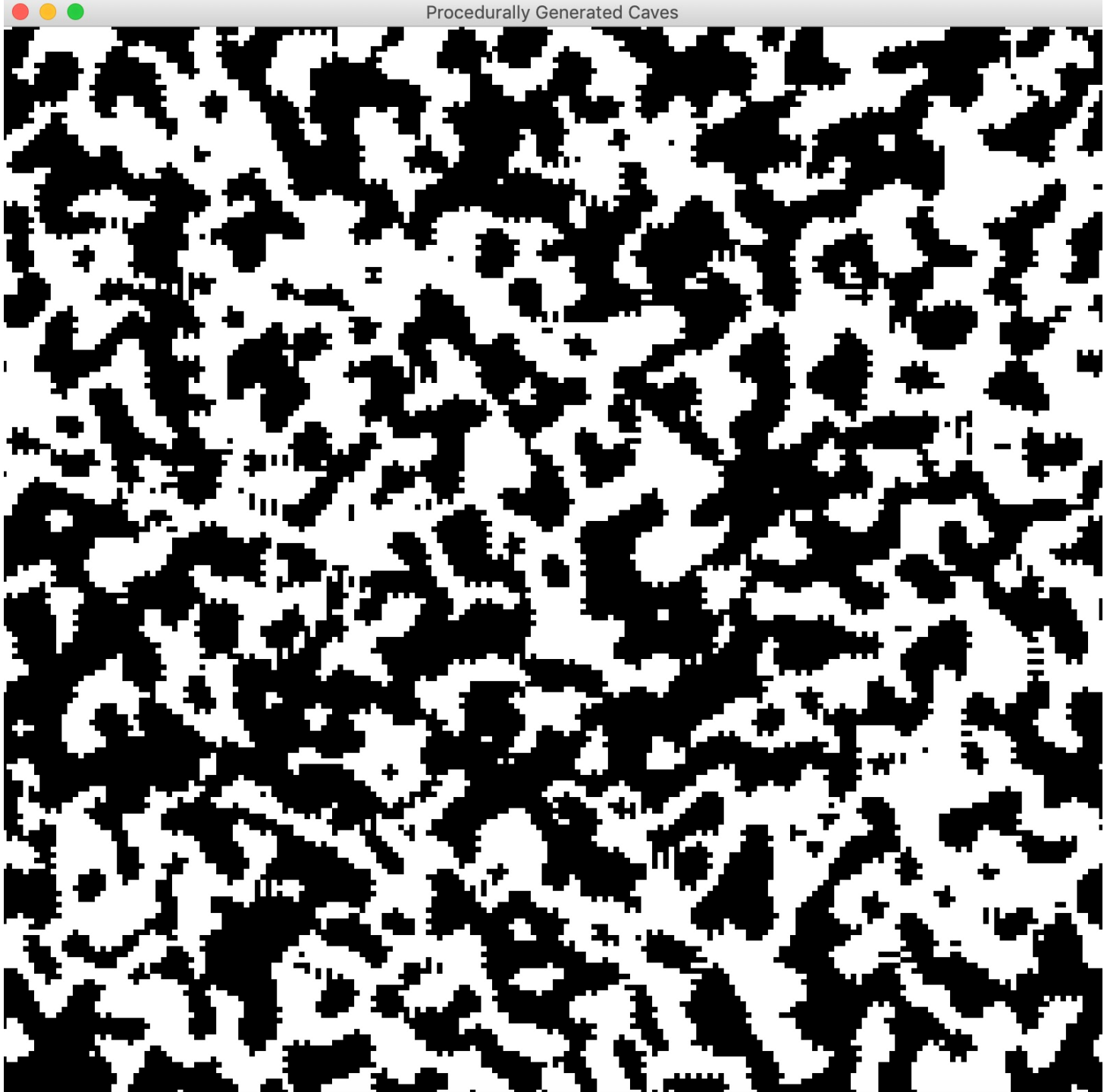
AStarDemoBaseCode.zip

(posted on Blackboard)

Another example of a Cellular Automata algorithm in use

- The image on the next slide is of an algorithmically-generated cave-like structure, for use in a 2D computer game. Each of the cells, laid out in a 200x200 grid, either has a wall (white) or a floor (black).

- The cellular automata algorithm which generated this output uses the following steps:
  - For each cell, randomly define it as: *wall* (60% chance) or *floor* (40% chance)
  - Perform the following procedure 4 times:
    - Calculate the number of wall neighbours of each cell, and define each cell which has at least 5 neighbouring wall cells, as a wall cell itself. Otherwise (i.e. if it has less than 5 wall neighbours) define it as a floor cell.


Procedurally Generated Caves

Assignment

- Write a Java program which implements the above algorithm
  - You'll need to deal with the 'edge wrapping' issue
  - Put a sizeable delay between each iteration of the algorithm, so that the user can see progress rather than just the end result
  - There's no need to use double buffering