Notes

Marks come out end of this week or next week – already marked they just need to do secondary marks

Assignment due 5th May

- Two states but the start/game over screen does not count as a state
- Each line should be commented in your code
- No set length for each level – choose difficulty just has to have 2 sets
- Make sure to quote/credit – mention inspiration – should be mentioned in report
- Games are harder – don't get extra marks for it
- No marks for website just need to be hosted on panel server
- Classes will be key and make the code easier

Classes:

- Like a blueprint – a template and with that template we can reuse the template instead of creating new
- To define class in a code you just use 'class'
- To use a class – make one, create a variable that is equal to the class and that creates a new instance (tbh you should watch the lecture recording it would make more sense) (around 45mins in)
- Arrays and classes link – an array can store all the variables you put for classes
- Always need a constructor function in class

```
class Bird{

    constructor(){

}

    show()

}
```

- Initialise all the variables that are going to be contained in our class
- Can add functions to classes for example, show() or up() – these functions are called methods when the functions belong to a class. The methods can be called using the dot operator. bird.show()

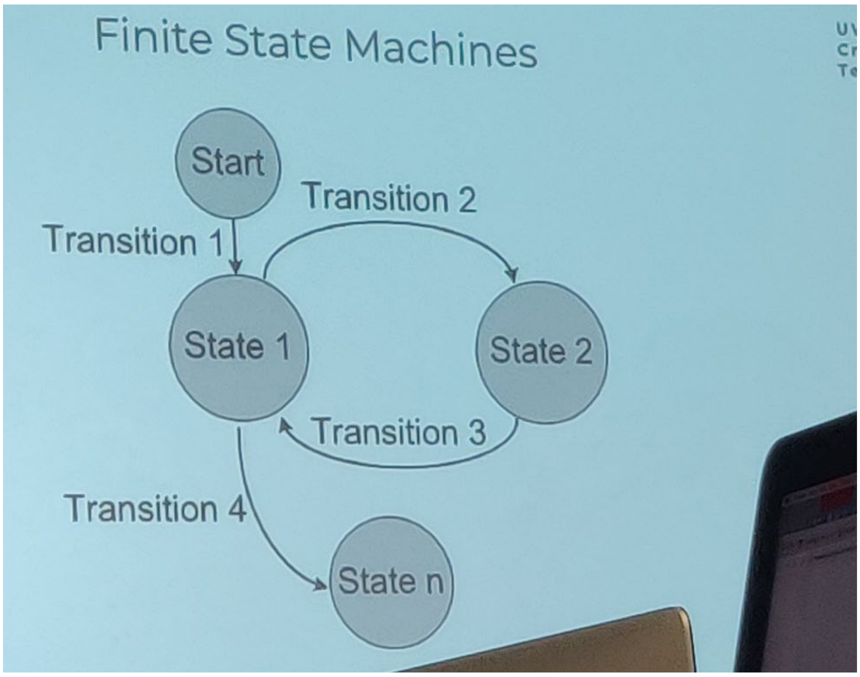
Separate js files:

- Might want to work in separate js files – especially for a game, to do this add them to the html

Finite state machine:

- Can only be in one **state** at one time
- Changes in **state** are called **transitions**
- Transitions occur based on some form of input or events

Notes



Finite State Machines

Example of a finite state machine – doesn’t have to follow this flowchart

- Switch case statement allows transitions to happen