CS4303: Video Games

Physics Practical: Ballista Command

Samurai Spirits, SNK

Shoot ‘em up

- A long-lived genre dating from the 1960s.
- Defeat enemies with the given weapons
- An early example is **Missile Command** ported to many platforms (1980):

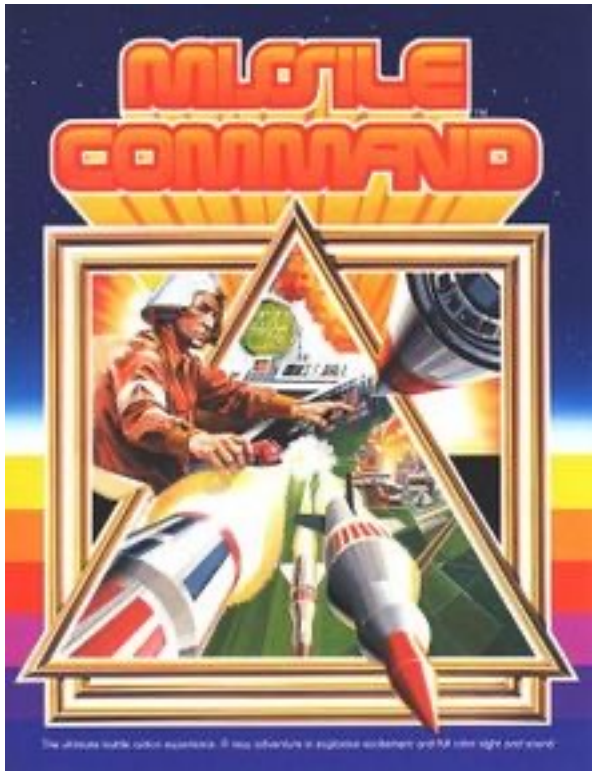


Image from Wikipedia

https://www.youtube.com/watch?v=nokIGklnBGY

Shoot ‘em up

- A long history, extending into modern gaming.

  https://en.wikipedia.org/wiki/Shoot_%27em_up

- More recent titles such as Geometry Wars or Jamestown

Image from Steam

Ballista Command

- This Practical comprises 20% of CS4303.
- It is due on Friday 12 February at 21:00.
- The Task:
  - Implement **in Processing** a variant of the classic Missile Command Game.
- Deliverables:
  - A report
  - A game guide
  - The Processing source code for the video game you will write.

Specification

- Single-player game
- Waves of meteorites fall from the sky
- Shoot at them to make them explode
- Points are scored for each one destroyed
- The game ends when no cities are intact

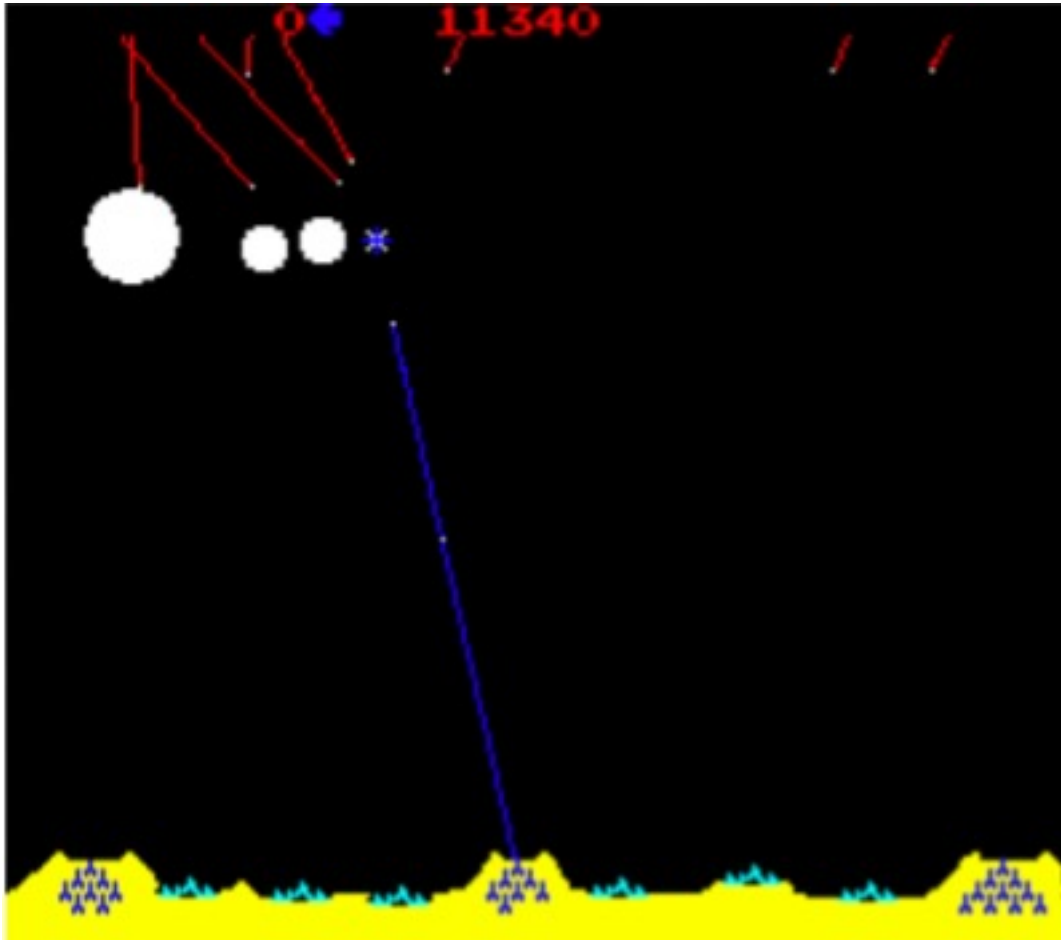


Image from Wikipedia

Twist

- All projectiles are affected by physics

https://www.youtube.com/watch?v=aiiQ8btusrs

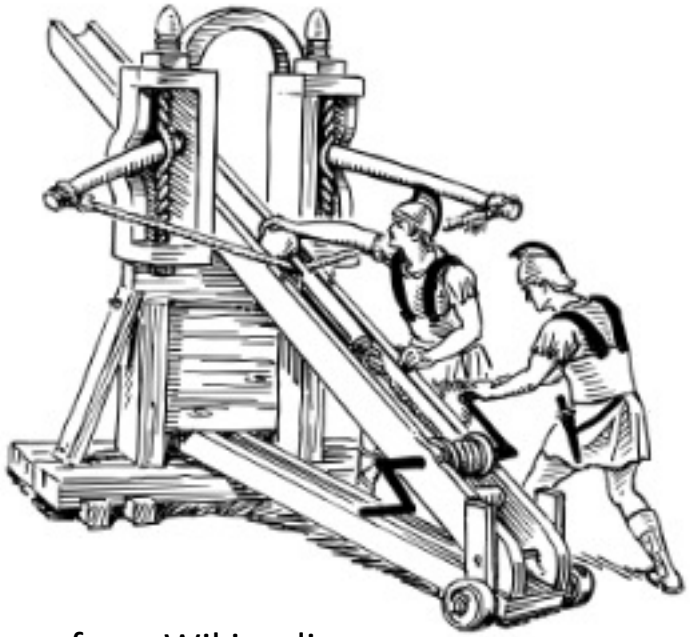


Image from Wikipedia

Similar behaviour to angry birds!

You Decide

- There are a number of places in the specification where you will need to make a decision about a design detail.
  - what are the controls?
  - Spawning angles/forces for meteorites?

- When making these decisions think about what effect it has on the game:
  - Does it make the game more or less fun to play?

The Play Area

- A 2d space with ground level near the bottom.
- There are three batteries: left, middle and right

The Play Area (2)

- Elements for you to decide (and document):
  - The dimensions and look of the play area, landscape, ballistas, rounds, meteorites ….
  - The strength of gravity and drag

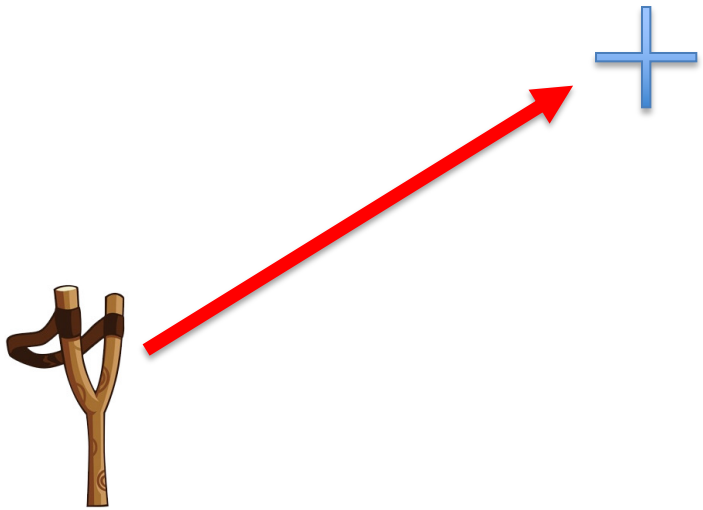
The meteorites

- Appear falling from the sky (the top of the play area).
- **Random** initial velocity
- Use the wave number to influence the quantity and initial velocity of each meteorite so as to increase **difficulty**
- The meteorites should be subject to:
  - The force of gravity,
  - The force of friction due to movement in the air

The Ballistae

- Each has 10 bombs to shoot
- Bombs affected by Initial force, then also affected by gravity and drag
- Explode on command!

- Initial vector:



The Ballistae

- Elements for you to decide (and document):
  - The implementation of the controls: mouse, keyboard, … ?
  - The Looks
  - Magnitudes
  - The dimensions and look of the bombs
  - Collision detection

Explosions

- Meteorite caught in the blast radius of an explosion should explode.
- This should allow you to fire various bombs and also chain explosions if timed properly.

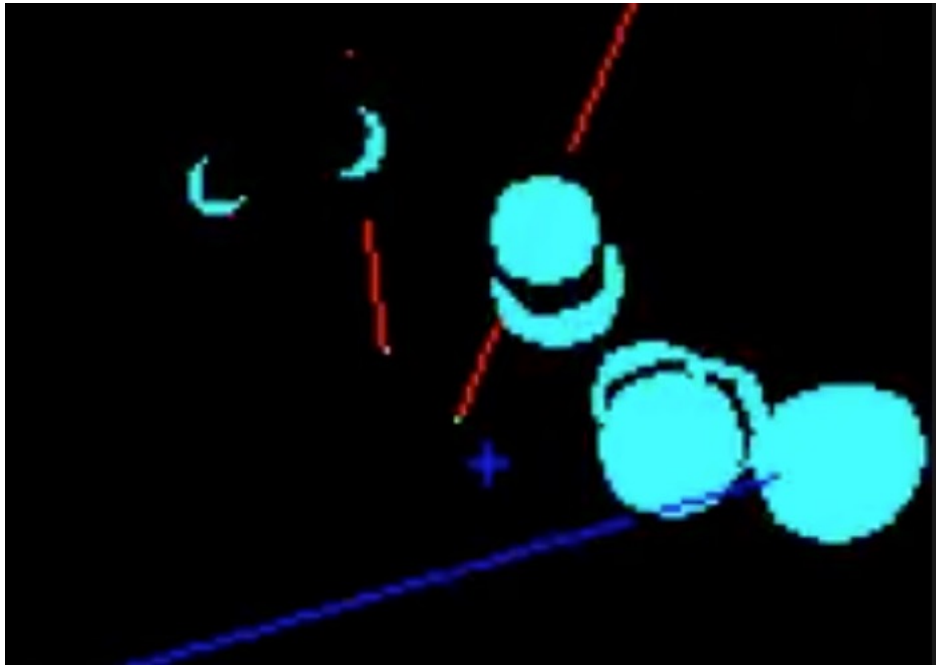
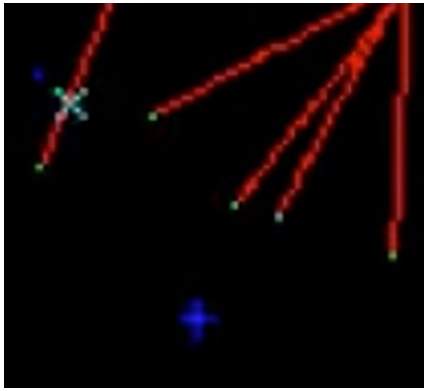


Image from youtube

Additional features

- Meteorites splitting mid-flight

- Bomber and satellite enemies firing additional meteorites to the cities



Images from youtube

User Interface

- At least, the following information should be displayed:
  - The score
  - Ammo and cities left
- Elements for you to decide (and document):
  - When the game ends will you provide the option of restarting the game?
  - Will you have an introductory screen giving instructions before the game starts?

Report

- Your report should document the design and implementation of your game.
- Include screenshots that show your game in operation and illustrate its features.
- Your report should include:
  - An **introduction** and **Conclusion** sections
  - discussion of all of the **design decisions**
  - an account of your implementation of the **physics involved**.

Marking

- The practical will be marked following the standard mark descriptors as given in the Student Handbook
- Further guidance as to what is expected:
- To achieve a mark of **7** or higher:
  - A bare bones implementation of the game, consisting of a single wave of falling particles. This implementation should be adequately described in an accompanying report and game guide

Marking

- To achieve a mark of **11** or higher:
  - In addition to the above,
  - the game should proceed in waves as specified, with an increasing level of difficulty.
  - This implementation should be well described in an accompanying report and game guide.

Marking

- To achieve a mark of **14** or higher:
  - In addition to the above,
  - some of the additional features should be implemented, together with a fully correct physics implementation. This implementation should be well described in an accompanying report and game guide.
- To achieve a mark of **17**:
  - the full basic specification above must be implemented, including both of the additional features. The report and game guide should be written to a high standard.

Marking

- To achieve a mark greater than **17**:
  - In addition to the requirements for a mark of 17, evidence of an exceptional achievement in terms of technical challenge, the original “smart bomb” enemies fully implemented and/or sound effects using the minim library for Processing