

School of Computer Science  
University of St Andrews  
2019-20  
CS4303  
Video Games  
Practical 1: Particle Command

This Practical comprises 20% of CS4303. It is due on **Friday 4th October** at 21:00.

The deliverables consist of:

- A report.
- A Player's Guide for your game.
- The Processing source code for the video game you will write.

### **Background**

This practical is intended to give you the opportunity to learn the Processing language, and implement some of the concepts from the Physics component of the module.

The task is to implement **in Processing** a variant of the classic video game Missile Command. If you are not familiar with this type of game, your first task is to read the following summary:

<https://www.technologyuk.net/computing/computer-gaming/gaming-landmarks-1960-1985/missile-command.shtml>

You can also play the game (requires Flash) here:

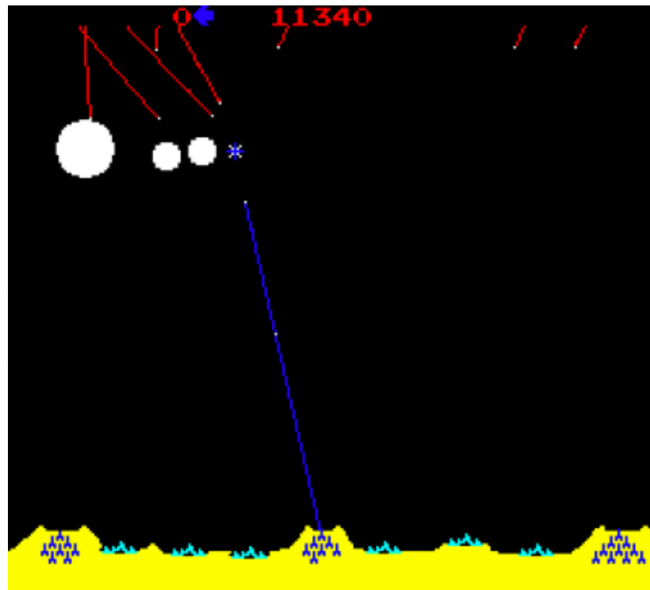
<https://my.ign.com/atari/missile-command>

### **Overview**

Particle Command is a single-player game in which the player controls three missile batteries, each with ten missiles. Using the missiles, the goal is to protect six cities from particles (think of these as rock projectiles rather than the missiles of the original game) falling from above.

### **The Play Area**

The play area is arranged as per Missile Command, as shown in the following screenshot:



Missile Command. Atari 1980.

The missile batteries occupy the left, right and centre positions on the ground, with two groups of three cities in between. This general arrangement should remain the same in your implementation of Particle Command, but the graphical appearance of these elements is left for you to decide and document.

### **Waves**

The game proceeds in waves. In each, a set number of particles fall towards the cities with the intent of destroying them. The number of particles should increase with the wave number so that the level of challenge in the game increases as the player progresses. The number of waves is unlimited.

The game is over when all cities have been destroyed.

### **Falling Particles**

The particles fall from the sky (the top of the play area). They should have a random initial velocity, but you should use the wave number to influence the initial velocity of each particle so as to increase difficulty in later waves. Particles must also be affected by both gravity and drag, as described in lectures.

If a particle hits a city, the city is destroyed. If a particle hits a missile battery, the missile battery is disabled for the current wave.

### **Missiles**

Each battery has a stock of ten missiles per wave. The number remaining in each battery should be clearly visible to the player.

The batteries are controlled independently by the player, who may choose which of the batteries fires a missile. Aiming is performed via a crosshair as per Missile Command. The control scheme for aiming and firing is left for you to decide.

In Particle Command, a missile is a particle with a thrust force propelling it in the direction of its orientation. Missiles should also be subject to gravity and drag. As in Missile Command, the missiles fired from the central missile battery should travel more quickly than those fired from the left or right batteries.

When a missile reaches its target position it explodes. Any particles caught in the blast radius of this explosion should themselves explode. The blast radius of these additional explosions may trigger further falling particles to explode, and so on.

### **Additional Features**

From wave 2 onwards, the following additional gameplay elements should appear:

- Some of the falling particles should split into two or more separate particles as they fall.
- Bomber and satellite enemies should appear intermittently and travel the play area horizontally, periodically dropping additional falling particles.

### **Scoring**

Basic scoring is as follows:

- The player should be awarded 25 points for each destroyed falling particle, and 100 for hitting the bomber or satellite enemies.
- At the conclusion of each wave, 100 bonus points are awarded for each surviving city, and 5 for each unused missile.

However, scoring is multiplied according to the wave following the same scheme as Missile Command:

- Waves 1 – 2: 1x
- Waves 3 – 4: 2x
- Waves 5 – 6: 3x
- Waves 7 – 8: 4x
- Waves 9 – 10: 5x
- Waves 11+: 6x

A city should be restored (if any have been destroyed) for every 10,000 points earned. Cities are restored at the end of the wave in which the required number of points is reached.

The current score should be depicted at the top of the screen, as in Missile Command.

### **Report**

Your report should document the design and implementation of your game in detail. In particular, it should contain an account of your implementation of the physics involved. Include screenshots that show your game in operation and illustrate its features.

### **Marking**

The practical will be marked following the standard mark descriptors as given in the Student Handbook (see link below). There follows 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.

- 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.
- To achieve a mark of 14 or higher: In addition to the above, some of the additional features should be implemented. This implementation should be well described in an accompanying report.
- To achieve a mark of 17: the full basic specification above must be implemented and the report written to a high standard.

### **Extensions**

There follow two possible extensions to your game. These are not required to gain a mark of 17, but at least one extension item (either from this list or of your own design) must be well implemented and documented to gain a mark above 17.

- In later levels of Missile Command a “smart bomb” enemy appears. Investigate and add this enemy to your implementation of Particle Command.
- Add sound effects! See the minim library for Processing.

### **Pointers**

Your attention is drawn to the following:

- **Mark Descriptors:**  
<https://info.cs.st-andrews.ac.uk/student-handbook/learning-teaching/feedback.html>
- **Lateness:**  
<https://info.cs.st-andrews.ac.uk/student-handbook/learning-teaching/assessment.html>
- **Good Academic Practice:**  
<https://info.cs.st-andrews.ac.uk/student-handbook/academic/gap.html>