

G54GAM Games

Introduction

Who am I?

- Contact Information
 - Dr. Martin Flintham
 - martin.flintham@nottingham.ac.uk
 - C2, Computer Science
 - Email or drop-in
- Course materials are on Moodle
- <http://moodle.nottingham.ac.uk/course/view.php?id=53754>

Module Delivery

- Theory Component – Lectures
 - 2x1 hour per week
 - Monday 16:00-17:00 A25 Business School South
 - Tuesday 12:00-13:00 LT2 Exchange
- Practical Component – Labs
 - 1x2 hour per week
 - Thursday 13:00-15:00 A32 Computer Science

Module Overview

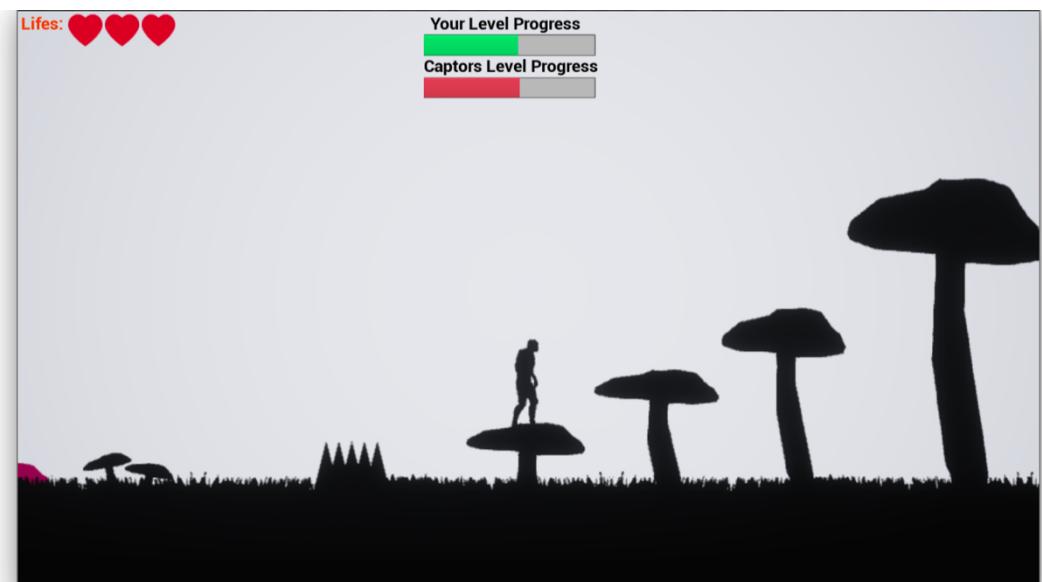
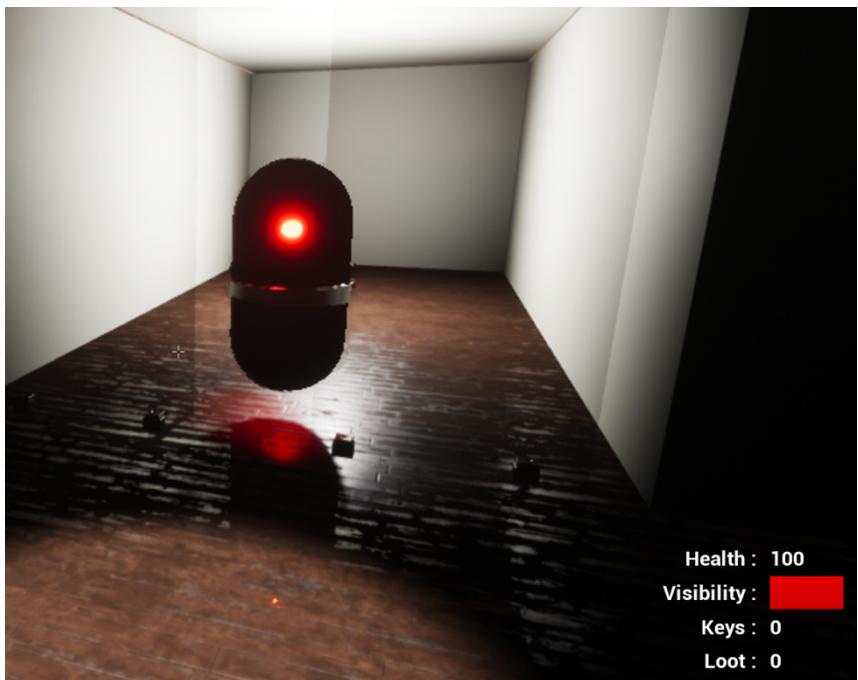
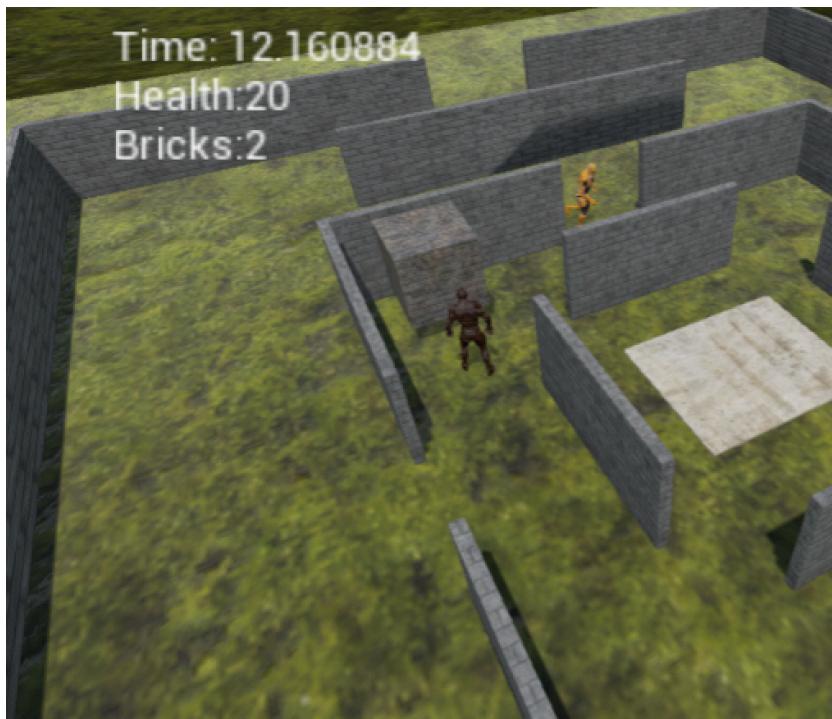
- This module covers the history, development and state-of-the art in computer games and technological entertainment.
- Students will gain an appreciation of the range of gaming applications available and be able to chart their emergence as a prevalent form of entertainment.
- Students will study the fundamental principles of theoretical game design, and how these can be applied to a variety of modern computer games.
- Students will study the development of games as complex software systems. Specific software design issues to be considered will include the software architecture of games, and the technical issues associated with networked and multiplayer games.
- Students will use appropriate software environments to individually develop a number of games to explore relevant theoretical design and practical implementation concepts.

Module Overview

- Education Aims
 - (To teach the history of games as an industry and a form of entertainment); to teach the principles of game design and implementation, and to give experience of designing and developing games.
- Learning Outcomes
 - Knowledge and Understanding
 - Understanding of the history, theory and practice of programming in the context of computer games.
 - Intellectual Skills
 - Understand and logically evaluate program requirements and specifications.
 - Understand the complex ideas of game programming solutions and relate them to particular problems.
 - Professional Skills
 - The ability to write programs.
 - The ability to transfer programming skills between classes of applications.
 - Transferable Skills
 - The ability to apply abstract frameworks to concrete examples.
 - The ability to solve problems using programming including mathematical problems, to schedule and present their work and to retrieve additional learning material.

Assessment

- Examination 30%
 - 1 hour written exam
 - Past papers available
 - All lectured material is examinable unless otherwise indicated
- Coursework 1 30%
 - 2000 word report
 - Study and analysis of a game design and player experience
 - Due ~March
- Coursework 2 40%
 - Portfolio of lab game designs + *significant* game prototype
 - Demonstrate use of the principles we will cover
 - “I heard this module is hard because you make people think of new things”
 - Due ~May



Expectations

- Why are you here?
- The games industry dislikes “games degrees” (apparently)
 - Somewhat of a misnomer to be able to “learn” “games”
 - (?)\$\$\$ billion worldwide revenue
 - Games are highly complex interactive systems
 - Art / design / writing ... coding / networks
- Love of games
 - Passion for games is helpful, but this is not an exercise in game *fandom*
 - Encyclopedic knowledge of games is not required – this module works by collectively sharing / discussing many examples
- A “hyper-masculine rhetoric”
 - Killing, weapons, violence, fighting, life and death, harm
- Prerequisites
 - “basic knowledge of programming”
 - Some mathematical competence 2d/3d cartesian coordinate systems

Expectations

- Our approach
 - Part of the Human-Computer Interaction theme
 - Theoretical and academic
 - Software engineering, technical challenges and the architecture of games
 - The application of existing theories / design principles / lenses / *strong concepts*
 - More abstracted than particular instances, without aspiring to be at the scope of generalized theories. We propose an intermediate design knowledge form that we name strong concepts that has the following properties: is generative and carries a core design idea, cutting across particular use situations and even application domains; concerned with interactive behavior, not static appearance; is a design element and a part of an artifact and, at the same time, speaks of a use practice and behavior over time; and finally, resides on an abstraction level above particular instances. Hook et al.
 - I.e. what are the features of a *first person shooter*? Develop a new game by experimenting with these features as design principles

Expectations

- 20 credits = 200 hours
 - ~40 hours lectures and labs
 - ~160 hours self study
- Turn up to lectures
 - Attendance is mandatory
 - **Take notes**
 - Engage in discussions
- Attend the lab sessions
 - Do the lab exercises
 - **Do more than the lab exercises.** Get other people to play your games. Make them fun.
- Do self-directed learning / reading / watching / *playing*

Topics (1)

- A brief history of games
- Categorising games
 - Properties of games and play
 - Design philosophies
 - Simulation, narratology, ludology
 - Core mechanics
 - Genre classifications
- Experiencing games
 - Meaningful play, interaction and choice
 - Mechanics, dynamics and aesthetics
 - Flow, immersion, presence

Topics (2)

- Elements of Game design
 - Formal and dramatic elements of play
 - Creating challenge
 - Progression and Balance
 - Narrative and storytelling

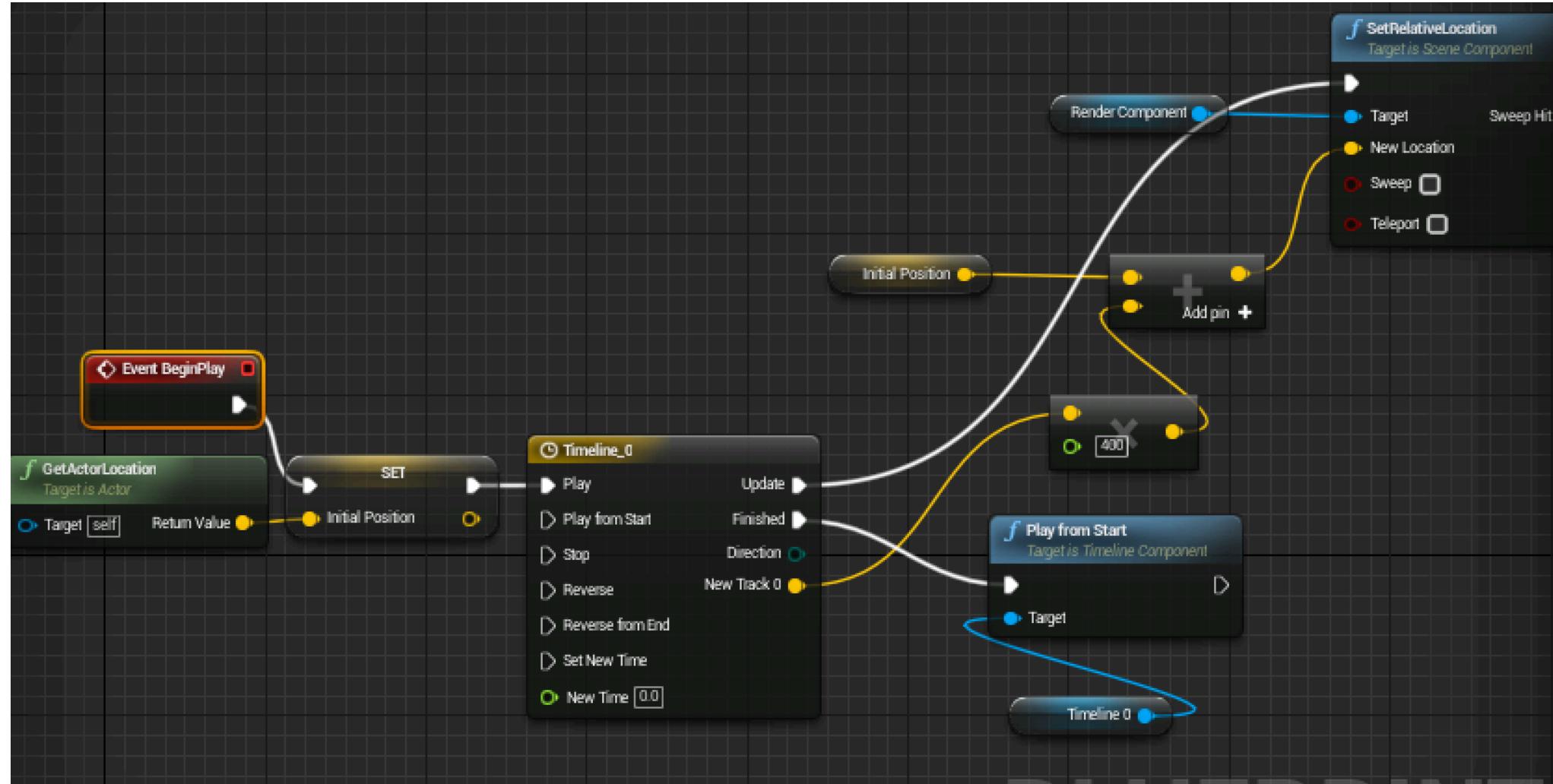
Topics (3)

- Game software architectures
 - Separation of work, software design patterns
 - Engine components, game loops
 - Multiplayer and network architectures
 - Physics and motion
- Serious games, Pervasive games

Labs

- Thursdays 13:00-15:00 A32
- Making games with Epic's Unreal Engine
 - <https://www.unrealengine.com/>
 - Free download, installed in A32
 - Why not Unity or ...?
 - Mostly IDE use / scripting
 - Can use C++ if you really want to
- Weekly exercises
 - Making simple games that explore different mechanics
 - Groundwork for coursework 2





Recommended Reading

- Rules of Play: Game Design Fundamentals. Katie Salen and Eric Zimmerman (2003)
- Game Design Workshop: A Playcentric Approach to Creating Innovative Games. Tracy Fullerton (2004)
- Andrew Rollings and Ernest Adams on Game Design. Andrew Rollings and Ernest Adams (2003)
- Game Engine Architecture. Jason Gregory (2014)
- Patterns in Game Design. Staffan Bjork and Jussi Holopainen (2004)
- Ian Bogost, Jane McGonigal
- **Youtube, <https://itch.io/>**

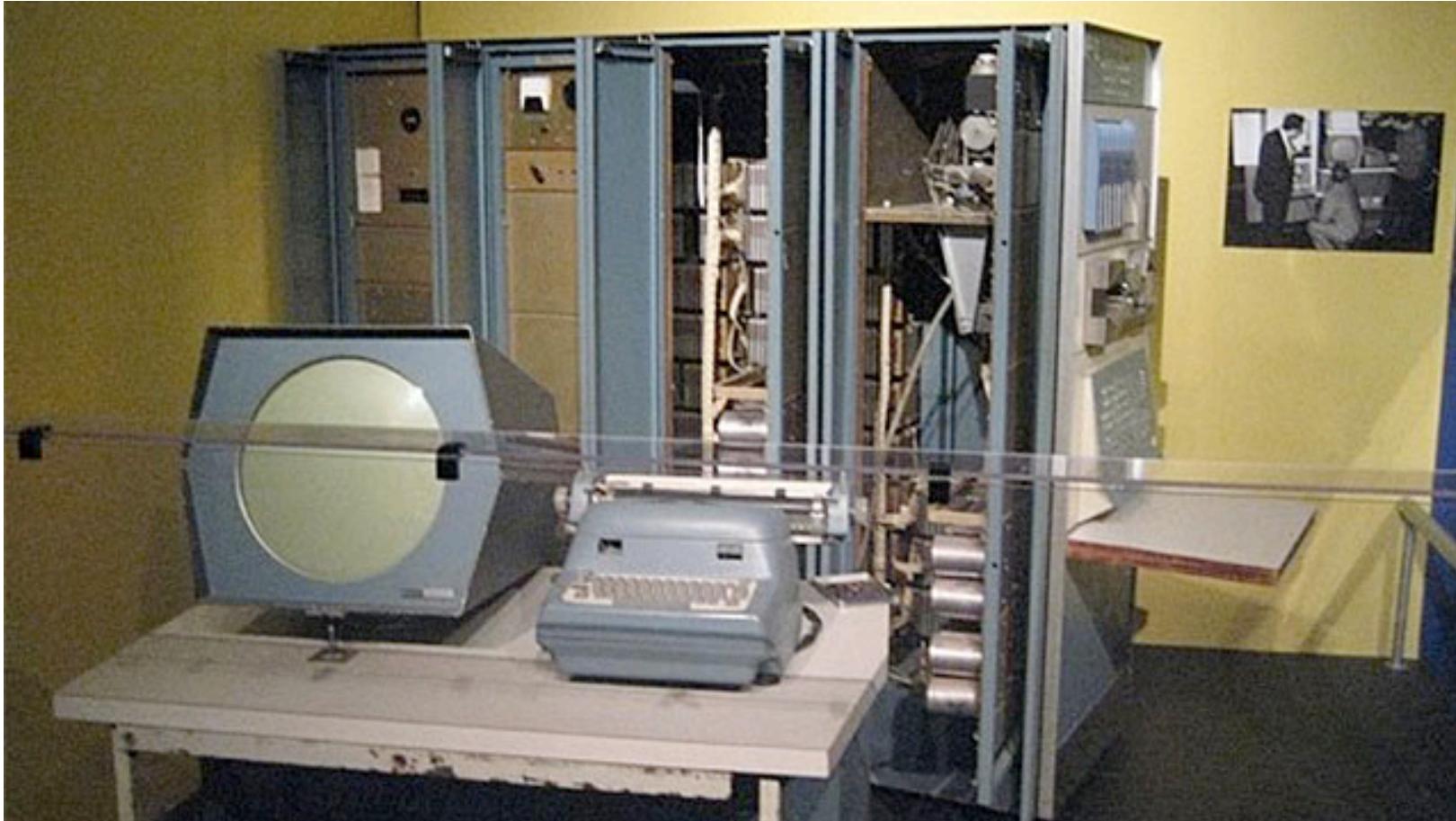
Questions on module structure?

A (Very) Brief History of Games

- Significant *eras*
- 8 generations of console
- Steady increase in sophistication of the home PC
- Incremental development
- What was the first video game?



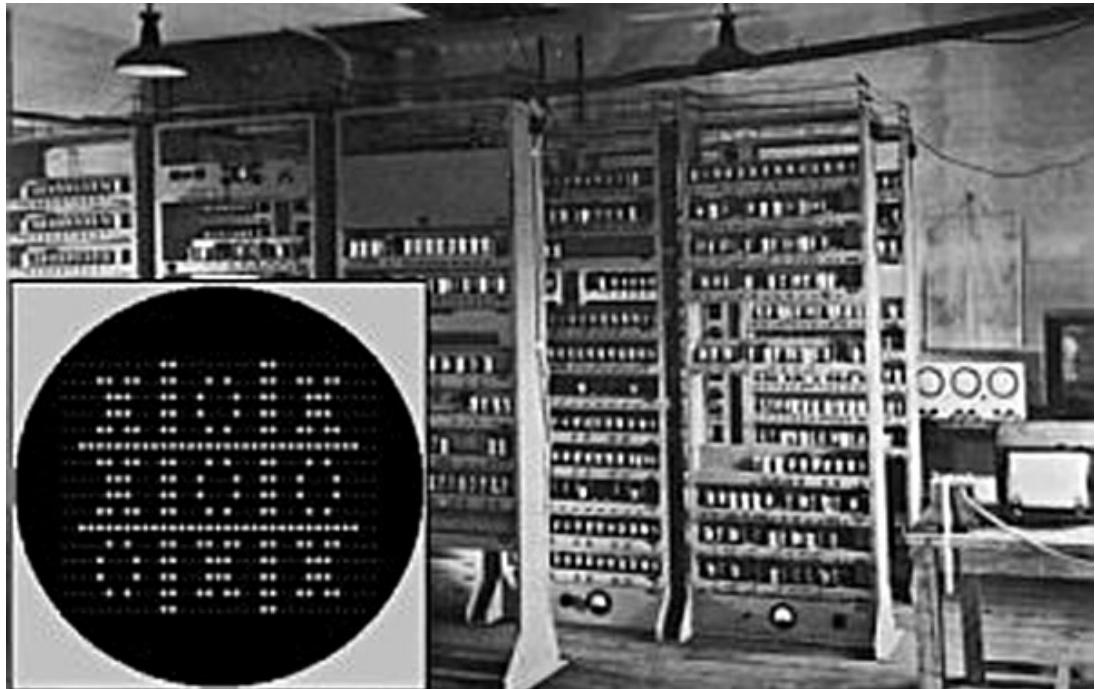
Cathode-Ray Tube Amusement Device (1947)



The first patent for an electronic game

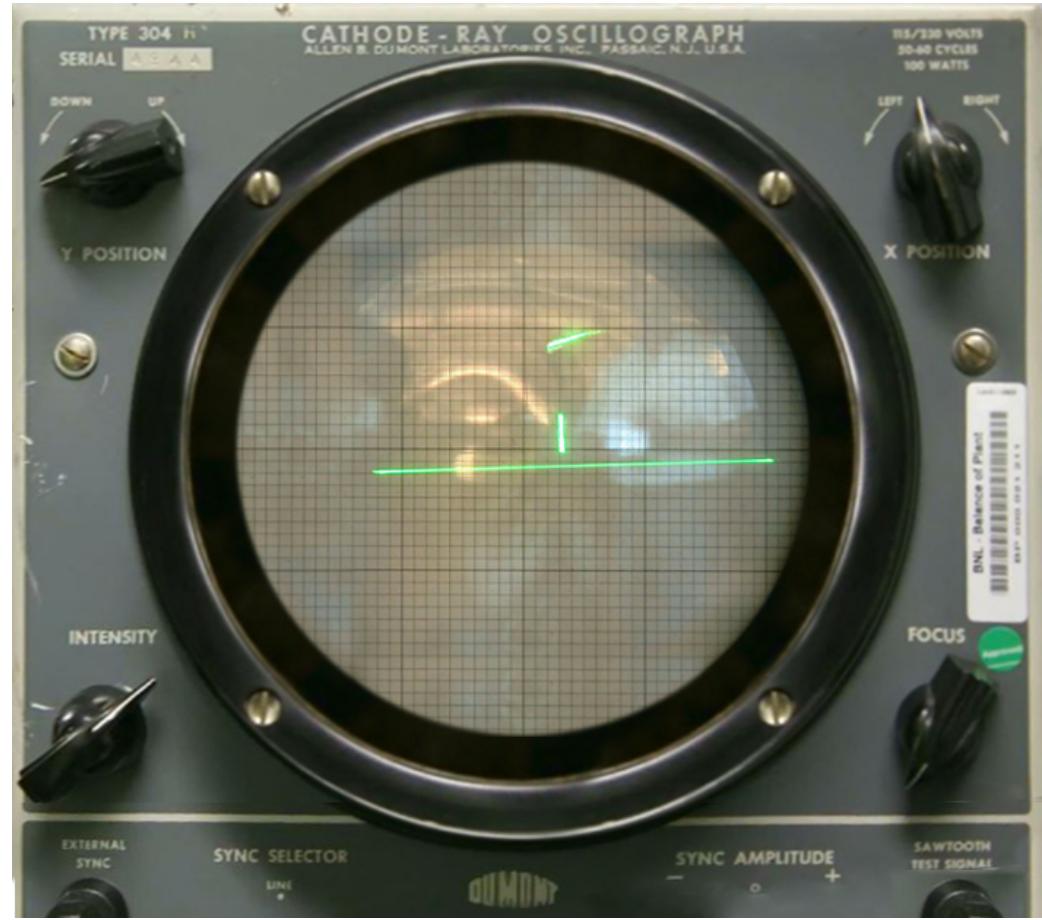
OXO (1952)

- Graphical tic-tac-toe
- Developed on the EDSAC
- First digital display
- Compete against rudimentary AI using a rotary dial
- <http://www.pong-story.com/1952.htm>
- Oldest *graphical* computer game



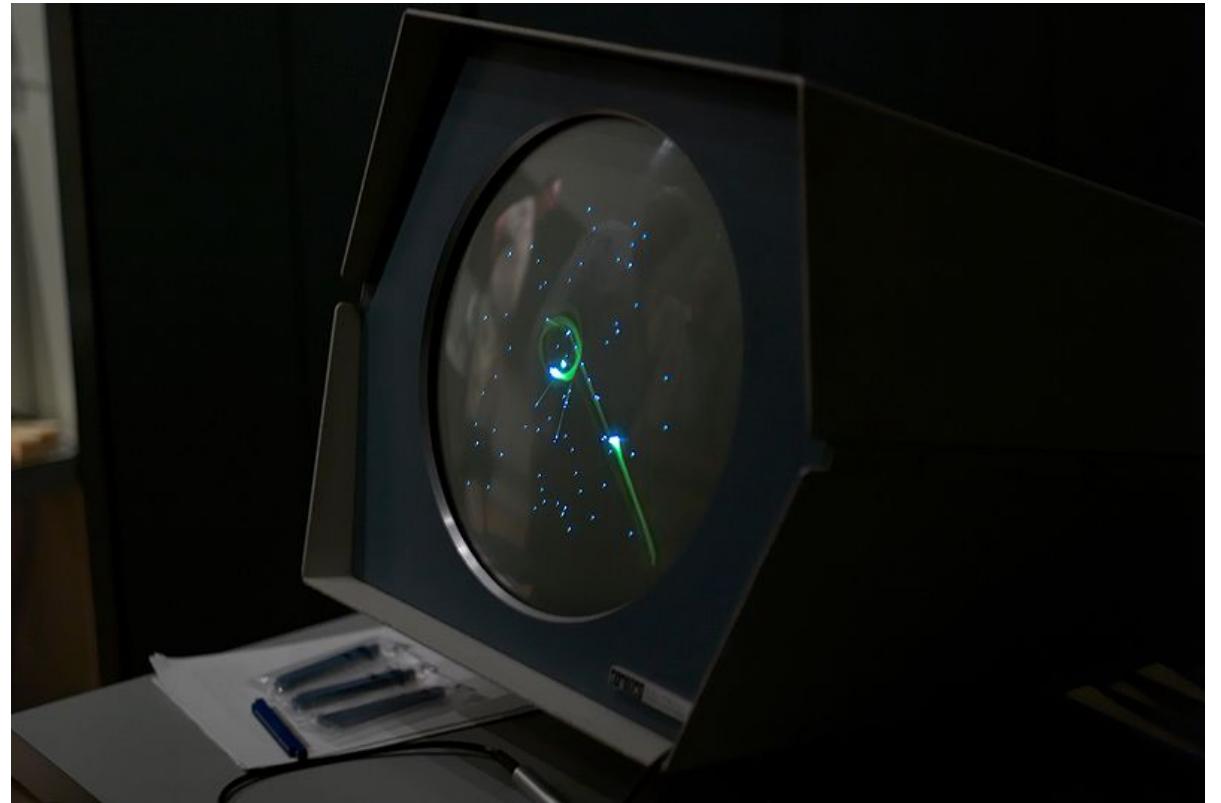
Tennis for Two (1958)

- Oscilloscope display
 - Analogue computer
- Two player
 - Players take turns to change trajectory of the “ball”
 - Simulates trajectories, wind effects, bounces, net



SpaceWar! (1961)

- DEC PDP-1
- Two players controlling ships
- Firing missiles at one another
- Manoeuvring in the gravity well of a star
- First widely available and *ported* computer game



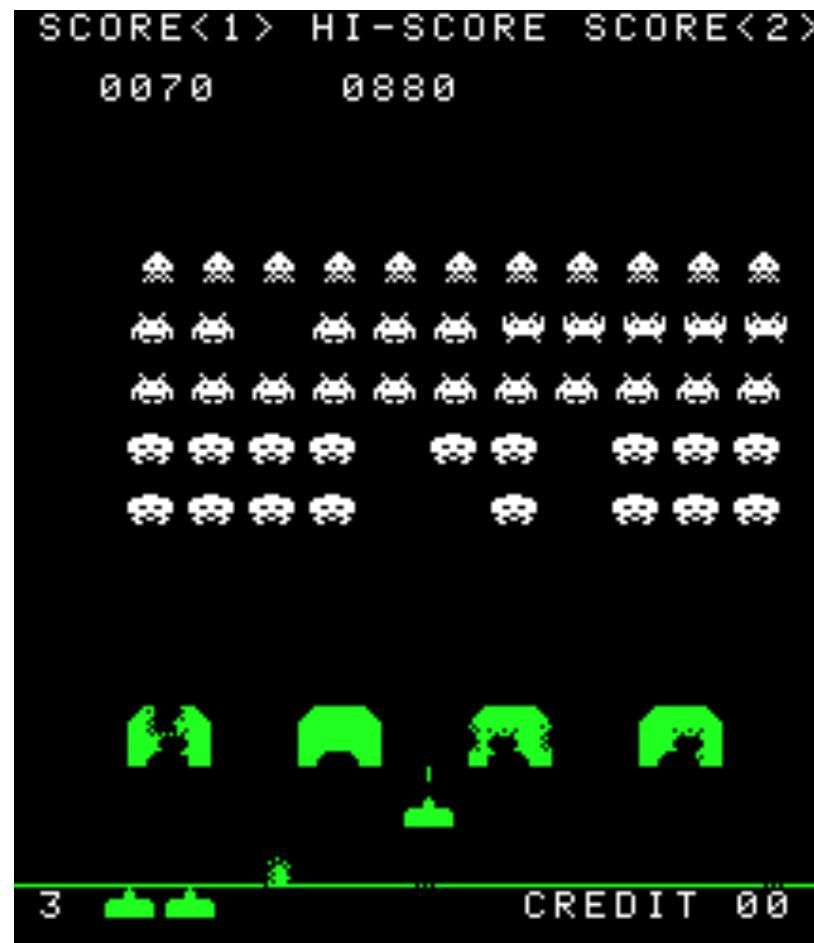
Arcade Games: Galaxy Game / Computer Space (1971)



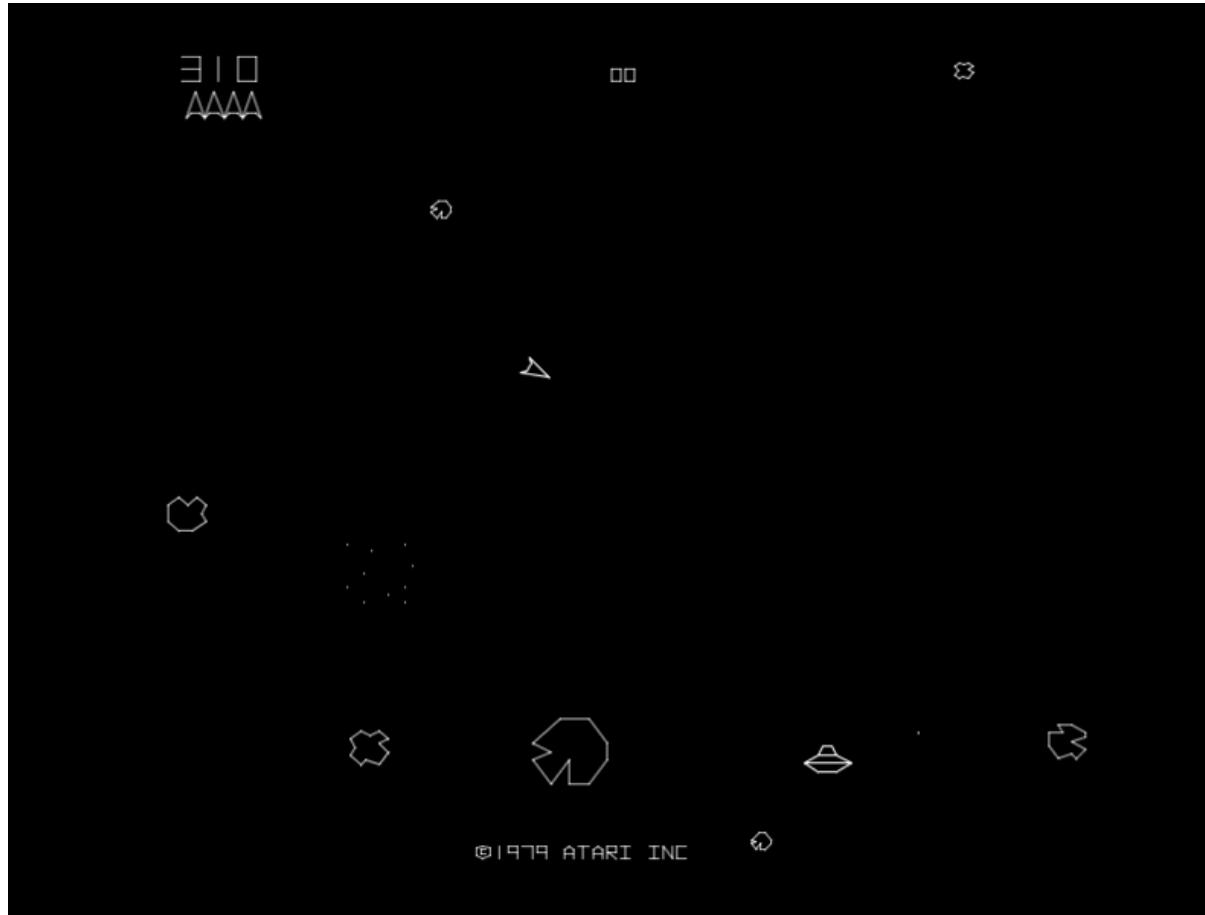
Arcade Games: Pong (1972)



Arcade Games: Space Invades (1978)



Arcade Games: Asteroids (1979)



Arcade Games: Pac-Man (1980)



University Mainframes: Adventure / Colossal Cave (1975)

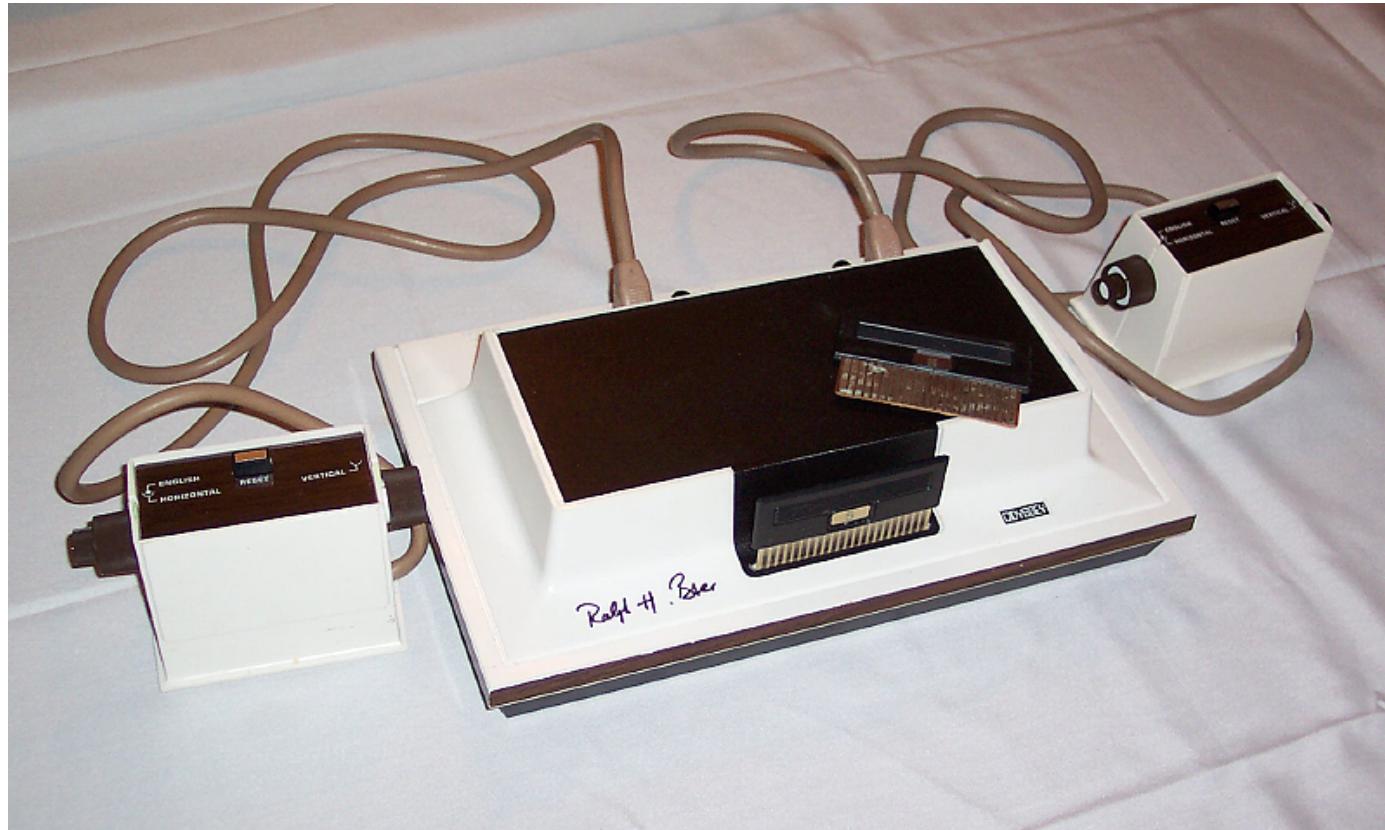
```
PAUSE INIT DONE statement executed
To resume execution, type go. Other input will terminate the job.
go
Execution resumes after PAUSE.
WELCOME TO ADVENTURE!! WOULD YOU LIKE INSTRUCTIONS?

y
SOMEWHERE NEARBY IS COLOSSAL CAVE, WHERE OTHERS HAVE FOUND
FORTUNES IN TREASURE AND GOLD, THOUGH IT IS RUMORED
THAT SOME WHO ENTER ARE NEVER SEEN AGAIN. MAGIC IS SAID
TO WORK IN THE CAVE. I WILL BE YOUR EYES AND HANDS. DIRECT
ME WITH COMMANDS OF 1 OR 2 WORDS.
(ERRORS, SUGGESTIONS, COMPLAINTS TO CROWTHER)
(IF STUCK TYPE HELP FOR SOME HINTS)

YOU ARE STANDING AT THE END OF A ROAD BEFORE A SMALL BRICK
BUILDING . AROUND YOU IS A FOREST. A SMALL
STREAM FLOWS OUT OF THE BUILDING AND DOWN A GULLY.
```

Also Hunt the Wumpus (1972), Adventure (1975), Zork (1977)

1st Gen Consoles: Magnavox Odyssey (1972)



Games pre-configured in console, “cartridges” input parameters



1st Gen Consoles: Home Pong (1975)



2nd Gen Consoles: Atari 2600 (1977)



Games burnt in ROM in removable cartridges

2nd Gen Consoles: Pitfall (1982)



2nd Gen Consoles: Pac-Man (1982)



2nd Gen Consoles: Intellivision (1980)



2nd Gen Consoles: Emerson Arcadia 2001 (1982)



2nd Gen Consoles: ColecoVision (1982)



2nd Gen Consoles: Atari 5200 (1982)



2nd Gen Consoles: E.T. the Extra-Terrestrial (1982)



Famously considered responsible for the “video game crash of 1983”. Documentary *Atari: Game Over*

8 bit era: BBC Micro



8 bit era: Commodore 64



8 bit era: ZX Spectrum



8 bit era: Atari ST



8 bit era: Commodore Amiga



8 bit era: Manic Miner (1983)



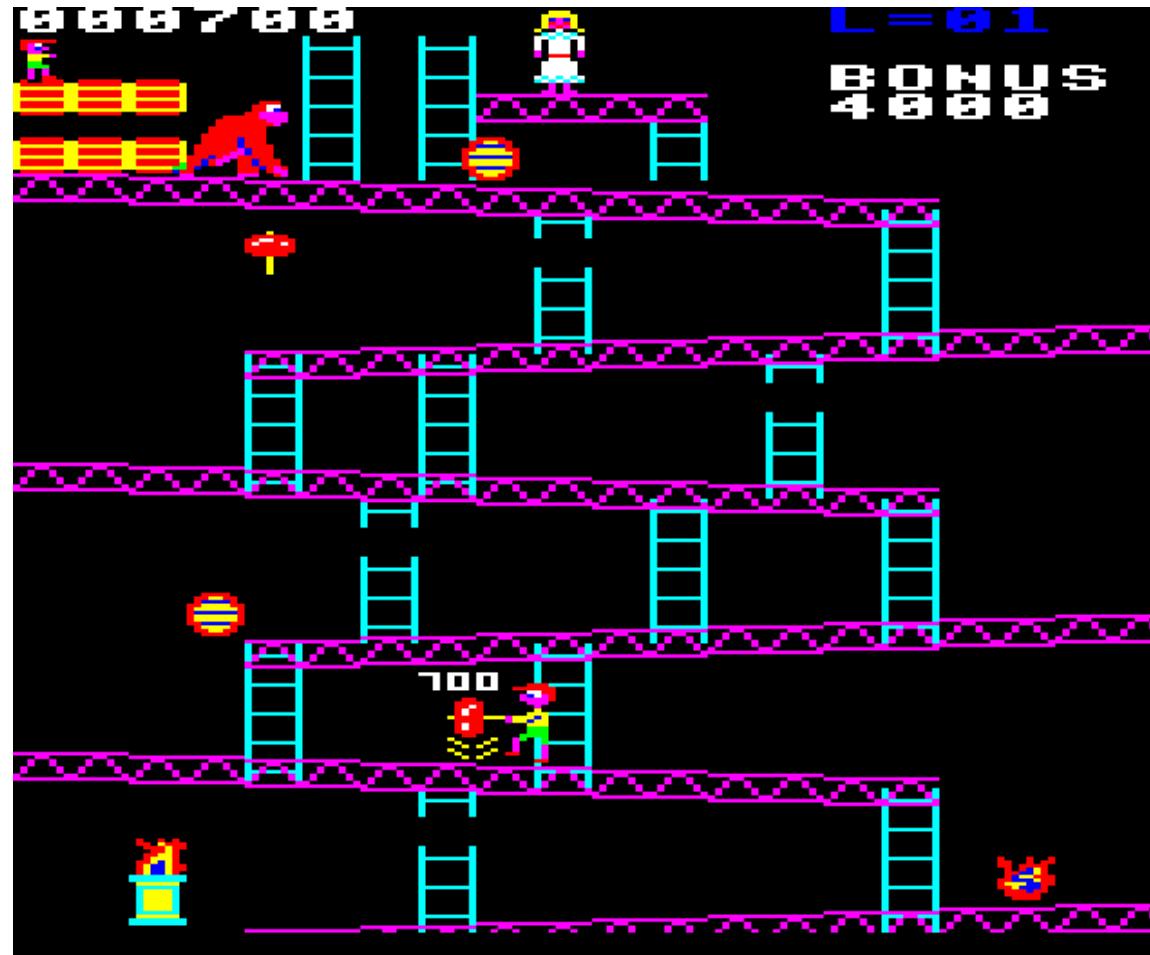
8 bit era: Jet Set Willy (1984)



8 bit era: Elite (1984)



8 bit era: Killer Gorilla (1984)



3rd Gen Consoles: Nintendo Entertainment System (1985)



8 bit, game pads, emergence of long-term franchises

3rd Gen Consoles: Sega Master System (1986)



3rd Gen Consoles: The Legend of Zelda (1986)



3rd Gen Consoles: Final Fantasy (1987)



4th Gen Consoles: Sega Mega Drive (1989)



Parallax scrolling, large sprites, stereo audio

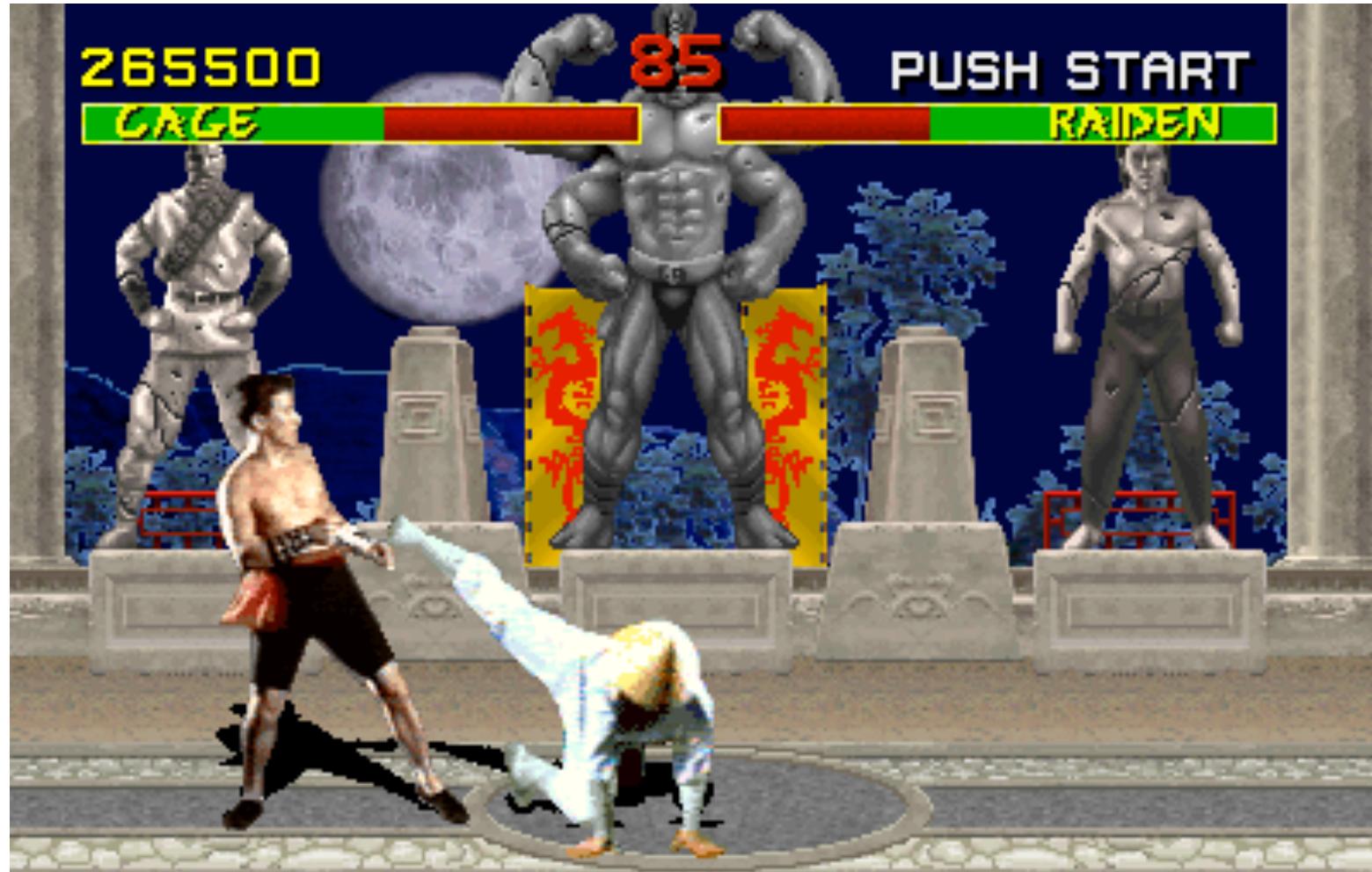
4th Gen Consoles: Super NES (1991)



4th Gen Consoles: Sonic the Hedgehog (1991)



4th Gen Consoles: Mortal Combat (1992)



90s PC: Wolfenstein 3D (1992)



Genre defining 32bit PC games, 1st 3D graphics accelerator cards

90s PC: Dune II (1992)



90s PC: Myst (1993)



90s PC: Doom (1993)



90s PC: Ultima Online (1997)



90s PC: Half-Life (1998)



5th Gen Consoles: Sony PlayStation (1994)

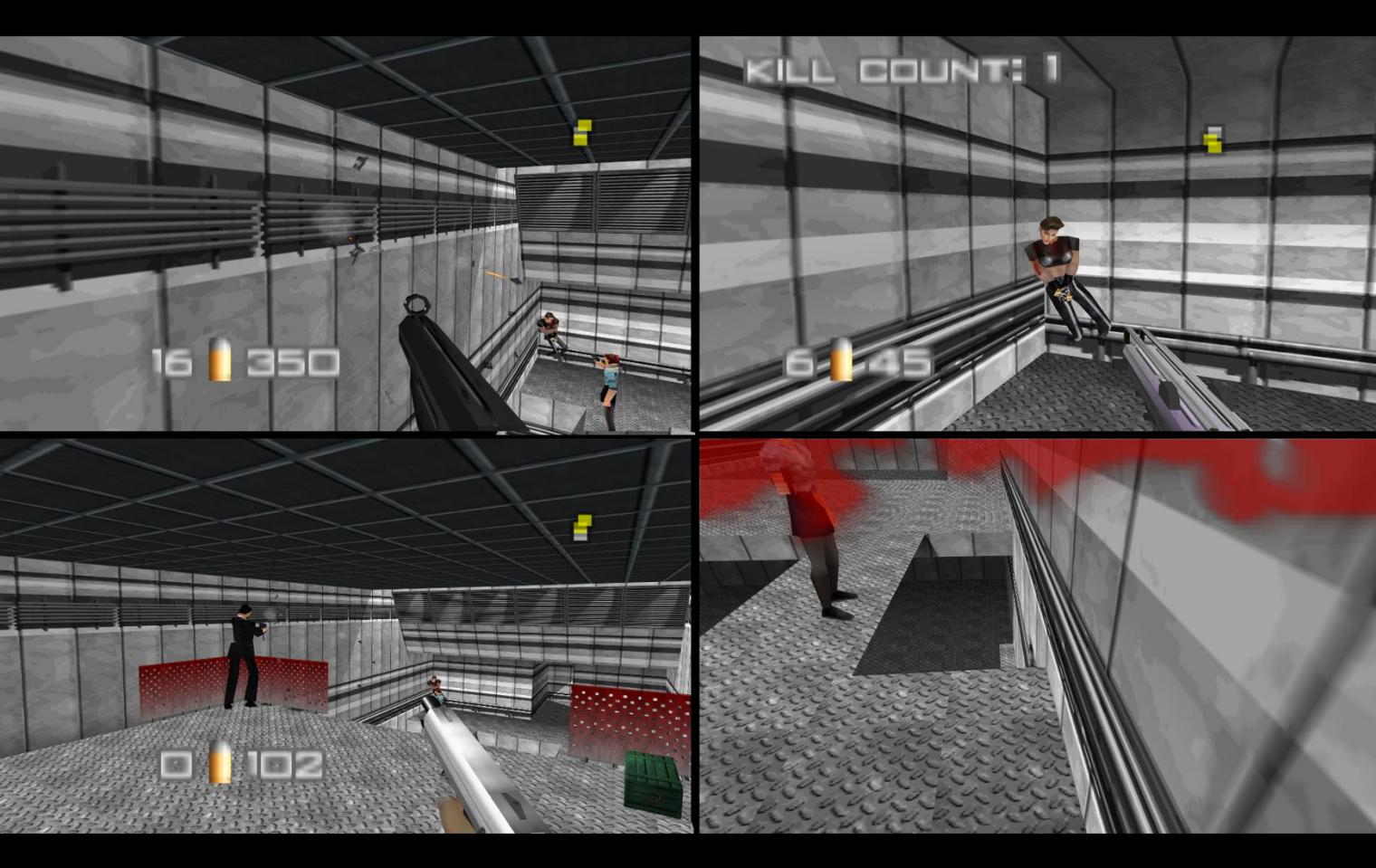


32 bit, 3D polygonal rendering, 24 bit colour, optical storage

5th Gen Consoles: Nintendo 64 (1996)



5th Gen Consoles: GoldenEye 007 (1997)



5th Gen Consoles: Legend of Zelda: Ocarina of Time (1998)



6th Gen Consoles: Sega Dreamcast (1998)



128 bit, some online support

6th Gen Consoles: Sony PlayStation 2 (2000)



6th Gen Consoles: Nintendo GameCube (2001)



6th Gen Consoles: Microsoft Xbox (2001)



6th Gen Consoles: Grand Theft Auto 3 (2001)



6th Gen Consoles: Halo (2001)



Online Gaming: Eve Online (2003)



Ubiquitous broadband access leads to growth of online PC gaming

Online Gaming: World of Warcraft (2004)



7th Gen Consoles: Xbox 360 (2005)



"Breakthrough technologies" – hi-definition rendering, HD movie playback, motion-based interaction

7th Gen Consoles: Sony PlayStation 3 (2006)



7th Gen Consoles: Nintendo Wii (2007)



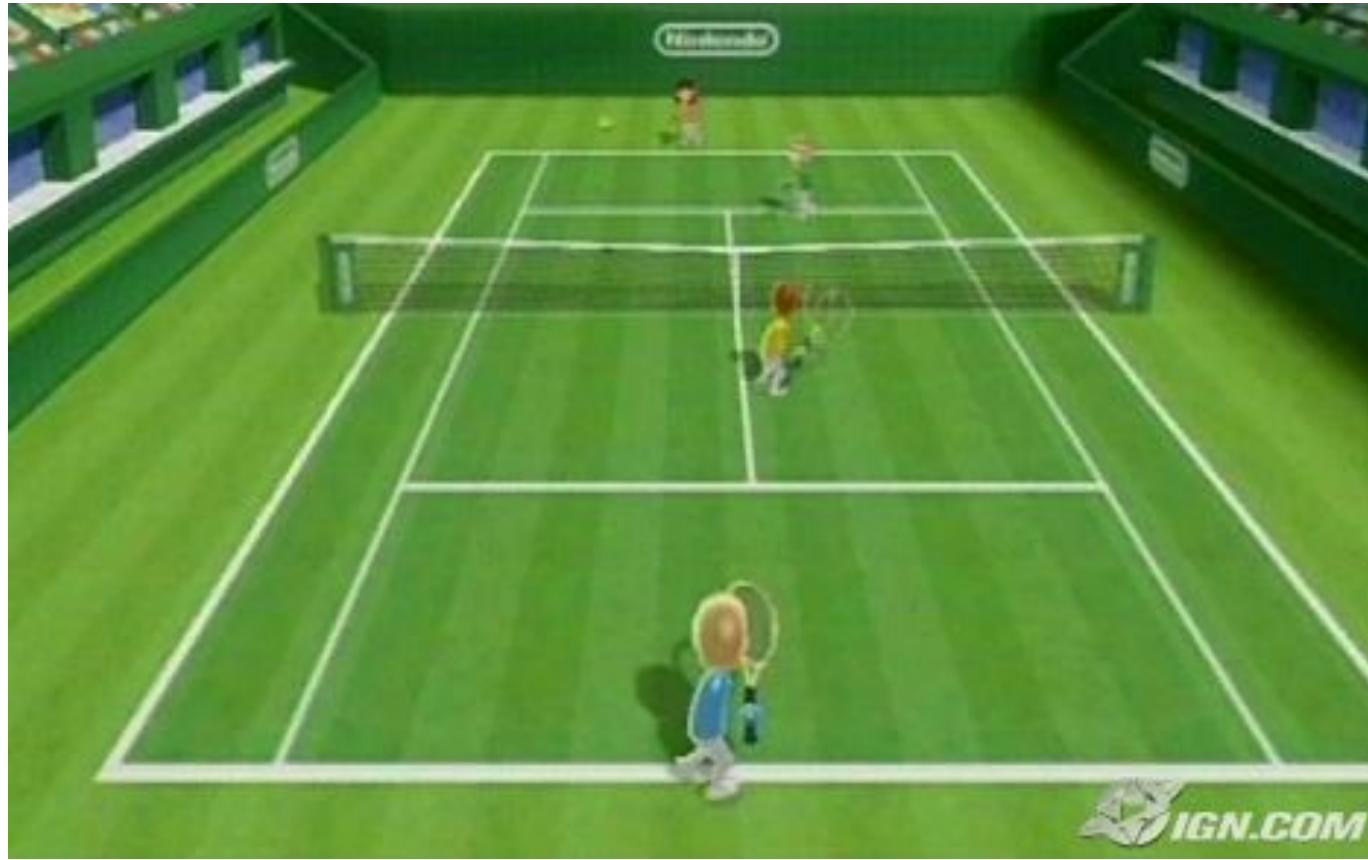
7th Gen Consoles: Grand Theft Auto IV (2008)



7th Gen Consoles: Call of Duty: Black Ops (2010)



Physical Interaction: Wii Sports (2006)



Physical Interaction: Guitar Hero (2005)



Physical Interaction: Microsoft Kinect (2010)

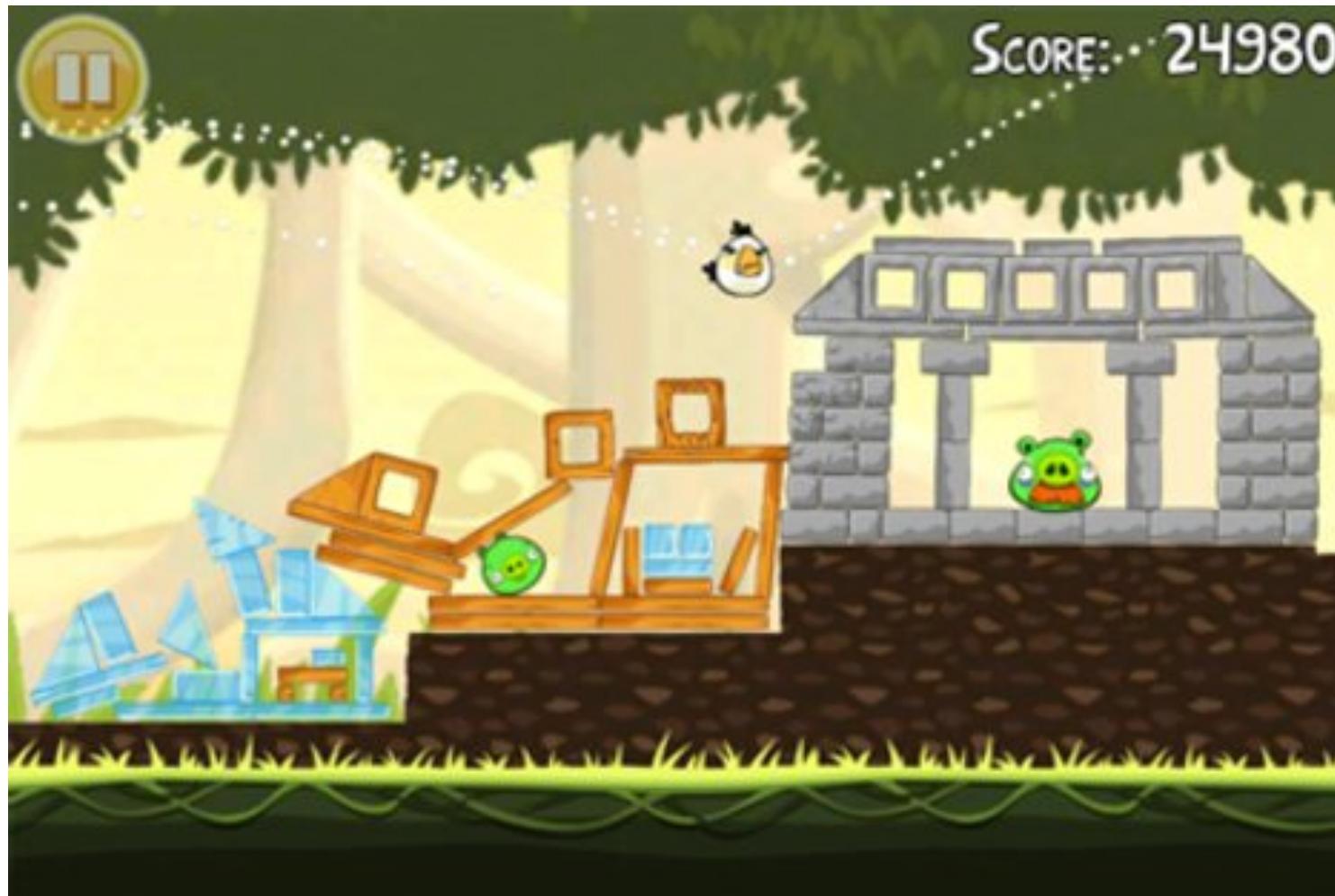


Casual: Bejewelled Blitz (2008)



Mobile phone, browser, social network-based games

Casual: Angry Birds (2009)



Casual: Farmville (2008)



Indie Mainstream: Braid (2008)



Indie Mainstream: Minecraft (2009)



8th Gen Consoles: PS4 / Xbox One (2013)



Wii U (2012)

8th Gen Consoles: Destiny (2014)



New Paradigms: No Man's Sky (2016)



Procedurally generated content

New Paradigms: Oculus Rift (2016)



New Old Paradigms: Idiot (2001)

