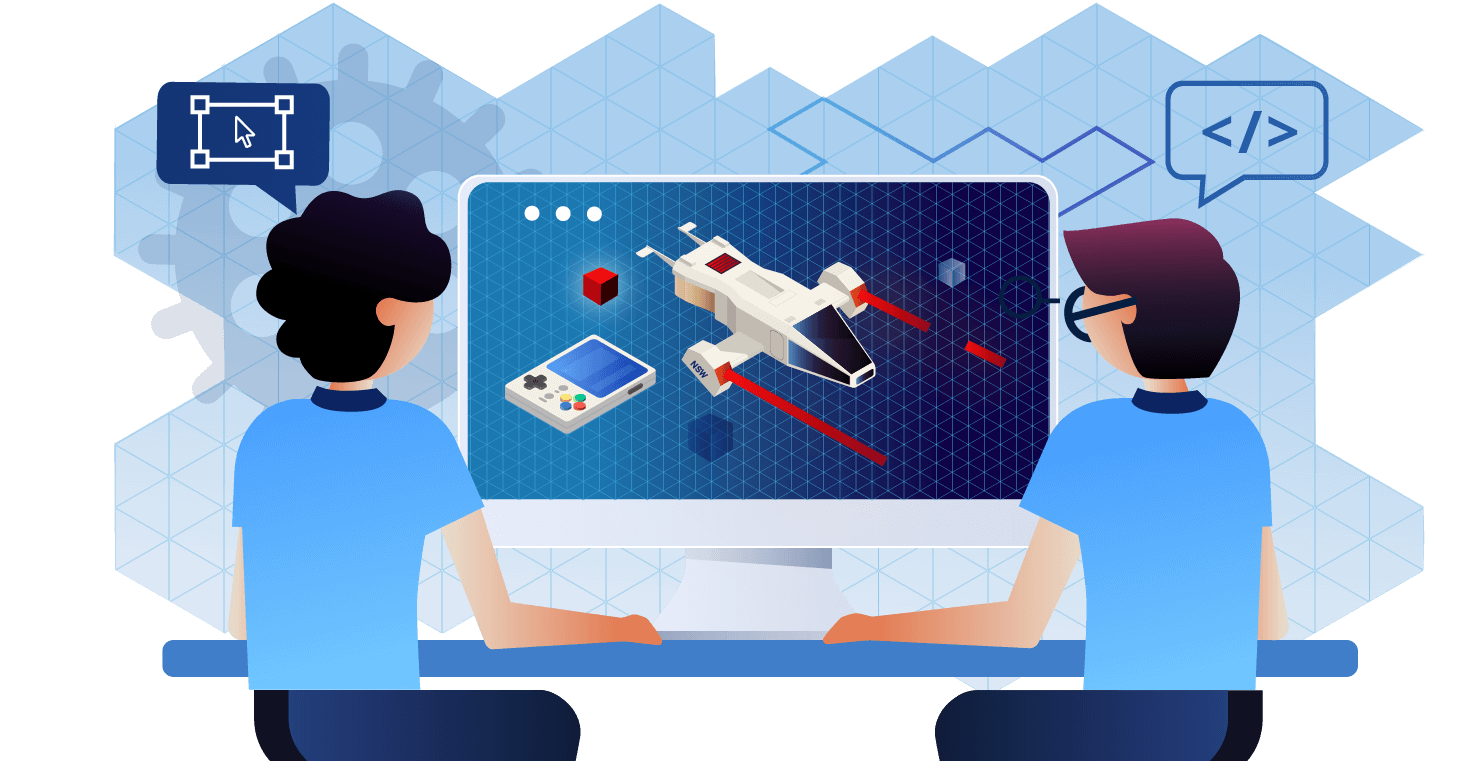
# Computing Technology Stage 5 (Year 10) – teacher support resource

**Software development** **– creating games and simulations**

****

## Teacher support resource

**Teacher note:** this resource has been designed to facilitate the ready conversion into a student booklet by removing the answers within the response windows. Teacher notes can be deleted before distributing to students.

Student name:

Class:

Teacher:

Contents

[Teacher support resource 1](#_Toc146194966)

[Unit overview 4](#_Toc146194967)

[Assessment task 1 overview 5](#_Toc146194968)

[Steps to success 6](#_Toc146194969)

[What is the teacher looking for? 7](#_Toc146194970)

[Assessment task 2 overview 8](#_Toc146194971)

[Steps to success 9](#_Toc146194972)

[What is the teacher looking for? 10](#_Toc146194973)

[Glossary 11](#_Toc146194974)

[The design and production process 16](#_Toc146194975)

[Identifying and defining 17](#_Toc146194976)

[Games and simulations 17](#_Toc146194977)

[Industry impact of games and simulations 18](#_Toc146194978)

[Perspectives of diverse groups in games and simulations 19](#_Toc146194979)

[The social impacts, and ethical and legal responsibilities in games or simulations 22](#_Toc146194980)

[Understanding challenges through game play 28](#_Toc146194981)

[Understanding the main features of a system 32](#_Toc146194982)

[Object oriented programming 35](#_Toc146194983)

[Features of object-oriented programming language activity 36](#_Toc146194984)

[Representation of logic when designing games and simulations 38](#_Toc146194985)

[Practical application – learning through coding 41](#_Toc146194986)

[Functional and non-functional requirements 42](#_Toc146194987)

[Functional and non-functional requirements activity 43](#_Toc146194988)

[Use case diagrams 44](#_Toc146194989)

[Predefined algorithms and artificial intelligence 46](#_Toc146194990)

[Design principles relevant to game design 47](#_Toc146194991)

[Data privacy and cybersecurity in game design 50](#_Toc146194992)

[Data ownership 54](#_Toc146194993)

[Data protection and cybersecurity 58](#_Toc146194994)

[Representing data 64](#_Toc146194995)

[Producing and implementing flowcharts 65](#_Toc146194996)

[Practical application – learning through coding 66](#_Toc146194997)

[Desk checking 67](#_Toc146194998)

[Error type in programming 70](#_Toc146194999)

[Practical application – learning through coding 74](#_Toc146195000)

[Gantt chart 74](#_Toc146195001)

[Record of project development 76](#_Toc146195002)

[Testing and evaluating 78](#_Toc146195003)

[Careers 79](#_Toc146195004)

[References 82](#_Toc146195005)

## Unit overview

In this unit students will develop a fundamental understanding of creating games and simulations. Students will investigate different game and simulation systems and research one in depth. Students will be guided through the design production process and use an iterative approach in creating a game or simulation as a solution to a user’s needs.

During Weeks 1 to 8 of the learning sequence, students will gain an understanding of the computational, design and systems thinking used in creating games and simulations. A range of games and simulations will be investigated that allows students to understand how innovation, enterprise and automation have inspired the evolution of computing technology.

During Weeks 9 to 18 of the learning sequence, students will design and test a system, creating a game or simulation which is coded and iterative in design. To develop their coding skills, students work to design, produce and evaluate algorithms and implement them in an object-oriented programming language. Students manage, document and explain individual work practices.

During Weeks 19 to 20 of the learning sequence, students showcase their project to the class and seek self and peer review. Students also investigate careers in the game and simulations industries.

## Assessment task 1 overview

**Type of task:** creating games and simulations research task and prototype

**Outcomes being assessed:**

A student:

* understands how innovation, enterprise and automation have inspired the evolution of computing technology **CT5-EVL-01**
* applies computational, design and systems thinking to the development of computing solutions **CT5-THI-01**

[Computing Technology 7–10 Syllabus](https://curriculum.nsw.edu.au/syllabuses/computing-technology-7-10-2022) © NSW Education Standards Authority (NESA) for and on behalf of the Crown in the State of New South Wales, 2022.

**Suggested weighting: 15%**

Students will research the game *Pac-Man* and investigate its structure and impact on society, communicating their findings through a presentation. Students will create a user interface design for an arcade style game considering the inputs and outputs, including a menu screen and playable area.

Students are to discuss the history of the *Pac-Man* game, created in the 1980s and its impact on the game industry at the time of release. Understanding *Pac-Man* gives the opportunity to look at mazes and data structure and control structures to navigate and score.

Students demonstrate an understanding of the hardware available at the time and how its limitations affected the processing involved in the ghosts.

Students will then discuss links between these early games and our current gaming environment.

### Steps to success

Table 1 – assessment preparation schedule

|  |  |
| --- | --- |
| Steps | What I need to do/when I need to do it |
| Compare old hardware to current hardware  Explore how the changing needs of society have influenced the development of games and simulations, including the impact of simulations and games on a range of industries. | * Produce a presentation outlining the differences between hardware from the 1980s and now and how it has affected games available then to now. * Elaborate on how this has affected the general audience for games over time. * Specify the non-functional requirements of a game or simulation. * Examples include * age suitability * motivation or immersion * visual appeal of a game world. |
| Understand Ghost Artificial Intelligence (AI) | * Using flowcharts, explain the logic that the 4 Ghosts in *Pac-Man* followed. |
| Discuss the AI effect | * Write a brief report outlining how these predefined algorithms created a simple AI and the resulting experience for the player. |
| Elements of an arcade style game | * Discuss the elements that make up an arcade style game and how the elements have been implemented into *Pac-Man*. * Some examples include * core gameplay loop * levels * self-contained progression * simple controls * lives * high score tracking * over the top gameplay * increasing difficulty. |
| Graphical User Interface (GUI) screen designs | * Provide 3 different screen designs for an arcade style game. The screen designs must include a menu screen and game area. |

### What is the teacher looking for?

The teacher is looking for a demonstration of understanding on how the tools available to game designers have changed over time and how the games offered have also changed as a result.

The teacher is looking for an appreciation of how creativity can overcome limitations when designing games and provide engaging experiences. Overcoming limitations is outlined and then utilised in the production of screen designs for your own game.

## Assessment task 2 overview

**Type of task:** creating games and simulations project and documentation

**Outcomes being assessed:**

A student:

* selects and applies safe, secure and responsible practices in the ethical use of data and computing technology **CT5-SAF-01**
* applies iterative processes to define problems and plan, design, develop and evaluate computing solutions **CT5-DPM-01**
* manages, documents and explains individual and collaborative work practices **CT5-COL-01**
* communicates ideas, processes and solutions using appropriate media **CT5-COM-01**
* designs, produces and evaluates algorithms and implements them in a general-purpose and/or object-oriented programming language **CT5-OPL-01**
* designs and creates user interfaces and the user experience **CT5-DES-01**

[Computing Technology 7–10 Syllabus](https://curriculum.nsw.edu.au/syllabuses/computing-technology-7-10-2022) © NSW Education Standards Authority (NESA) for and on behalf of the Crown in the State of New South Wales, 2022.

**Suggested weighting: 35%**

Students will utilise their programming knowledge and skills to create an arcade style game using Python and an appropriate framework (Pygame Zero). Students will work through and demonstrate the producing and implementing stages of developing an arcade-style game by submission of a portfolio and working game.

### Steps to success

Table 2 – assessment preparation schedule

|  |  |
| --- | --- |
| Steps | What I need to do/when I need to do it |
| Create a portfolio | * Create a document that is presented professionally using techniques learnt earlier in the semester and which has a heading for each of the steps listed in this table. |
| Produce a Gantt chart | * Create a spreadsheet and implement a Gantt chart to show intended work over the term. |
| Identifying and defining | * Identify the need for the game. * Define the problem and requirements. * Propose a solution. |
| Researching and planning | * Research 2 existing arcade games focusing on game elements that are included in these games that can be implemented into your game. * Detail the logic for part of your game using either a flowchart or pseudocode. |
| Implementing | * Produce an arcade style-game using Python. The game must have a menu, clear objective, score tracker and appropriate screens. * Produce internal and intrinsic documentation. |
| Testing | * Produce evidence, in the form of screenshots, to show that you have checked your product for errors, utilised error correction and are confident that it functions correctly under different circumstances. |
| Evaluate your game | * Gather feedback from 3 peers, using the rubric in Assessment task 2, to gather feedback on your game. * Evaluate your final product and how well you believe it meets the requirements for your product as set out in the identifying and defining section of your portfolio. |

### What is the teacher looking for?

The teacher is looking for a game that could be placed into an arcade that is playable and enjoyable.

Some features of an arcade game include:

* ensuring multiple design principles are incorporated into the game design
* allowing for multiple games or attempts, keeping track of the number of games each player has won
* implementing high score tracker
* allowing for the game progress to be saved and then resumed at a later time
* recording the history of moves so that the game may be replayed
* main menu
* multiplayer capabilities
* implementing a basic AI that the player may play against.

The quality of the code will also be considered as well as the presentation of your supporting material documenting the development of the game product.

## Glossary

Many of the following words will gather more meaning to you as you work through this booklet.

Each time you see an unfamiliar word in bold throughout this workbook you can add its definition in the table below in case you need to refer back later.

|  |  |
| --- | --- |
| Word | Definition |
| Aesthetics | Aesthetics is a core design principle that defines a design's pleasing qualities. In visual terms, aesthetics includes factors such as balance, colour, movement, pattern, scale, shape and visual weight. |
| Algorithm | An algorithm is a procedure used for solving or performing a computation. |
| Artificial Intelligence (AI) | The theory and development of computer systems able to perform tasks normally requiring human intelligence, such as visual perception, speech recognition, decision-making, and translation between languages. |
| Augmented Reality (AR) | A technology that replicates, enhances or overlays extra information about the real-world environment, using computer-generated data such as global positioning systems (GPS), sound, videos and images. Common examples include a car windshield with a heads-up display (HUD) that projects 3-dimensional navigation information and virtual lanes, or a swimming telecast using a line to indicate the position of the record holder in relation to the actual swimmers in the race. |
| Branching | An instruction in a computer program or algorithm that causes different actions to be performed depending on specified conditions. |
| Cyber safety | A term that refers to behaviour and the precautions that may be exercised when providing personal information in an online or digital environment. |
| Cybersecurity | Cybersecurity is the protection of internet-connected systems such as hardware, software and data from cyberthreats. |
| Data | A discrete representation of information using number codes. Data may include characters (for example alphabetic letters, numbers and symbols), images, sounds and/or instructions that, when represented by number codes, can be manipulated, stored and communicated by digital systems. For example, characters may be represented using ASCII code or images may be represented by a bitmap of numbers representing each ‘dot’ or pixel. |
| Desk checking | Desk checking is the process of manually reviewing the source code of a program. It involves reading through the functions within the code and manually testing them, often with multiple input values. |
| Event-driven programming | Event-driven programming is a programming paradigm in which the flow of the program is determined by events such as user actions (mouse clicks, key presses), sensor outputs, or message passing from other programs or threads. |
| Flowchart | A graphical representation of the sequence of operations in an information system or program. Different symbols are used to draw each type of flowchart. |
| Function | A term used in programming to describe a self-contained sequence of instructions that performs a specific task or tasks and is designed to be able to be reused throughout the program. Functions often accept some kind of input, perform some process on that input, and return a result that can be used by other parts of a program. Most programming languages allow for user-defined functions but will also provide pre-defined functions. |
| Functional requirement | Functional requirements define the software's goals, meaning that the software will not work if these requirements are not met. |
| Game | A game is a form of interactive entertainment that is typically characterised by rules, a goal, and a feedback system. The goal of a game is often to win or achieve a high score, and the feedback system is often used to inform the player of their progress towards this goal. |
| General purpose programming language | A coding or programming language used to write computer software. It uses letters, numbers and symbols arranged in a prescribed format (language) to instruct a computer how to carry out specific tasks. Also known as text-based programming. |
| Iteration | A repetition of a process in computer programming where each repeated cycle builds towards a desired result. |
| Logic | A system or set of principles underlying the arrangements of elements in a computer or electronic device so as to perform a specified task. |
| Logical operators | Logical operators are generally used for combining 2 or more relational statements. An operator used to compare logical expressions that returns a result of true or false. Common logical operators include AND, OR, and NOT. |
| Mixed reality (MR) | Mixed Reality (MR) is a term used to describe a continuum between the real and virtual worlds, where physical and digital objects co-exist and interact in real-time. It is a combination of Augmented Reality (AR) and Virtual Reality (VR) and it creates a hybrid environment where the user can interact with both the real and virtual world. |
| Non-functional requirement | A non-functional requirement is a [requirement](https://en.wikipedia.org/wiki/Requirement) that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours. They are contrasted with [functional requirements](https://en.wikipedia.org/wiki/Functional_requirement) that define specific behaviour or functions. |
| Object-oriented programming (OOP) language | A type of programming language that organises code around data, or objects, rather than functions and logic. An object is a data field that has unique attributes and behaviour. |
| Prototype | A trial product or model built to test an idea or process to inform further design development. Its purpose is to see if and how well the design works and is tested by users and systems analysts. A prototype can be both a physical object or exist in digital form. |
| Pseudocode | Writing in plain English line by line (step by step) what you want the computer program to do. |
| Relational operators | A relational operator is a programming language construct or operator that tests or defines some kind of relation between 2 entities. |
| Simulations | A simulation is a model that mimics the operation of an existing or proposed system, providing evidence for decision-making by being able to test different scenarios or process changes. |
| Test case | A test case is a set of actions performed on a system to determine if it satisfies software requirements and functions correctly. |
| Use case | A use case is a description of the ways in which a user interacts with a system or product. |
| Virtual reality (VR) | Virtual reality (VR) refers to a computer-generated simulation of a 3-dimensional environment that can be interacted with using specialised hardware, such as a head-mounted display (HMD) or gloves fitted with sensors. The user is immersed in this environment and can interact with it as if it were real. |

**Teacher note:** for students with an EALD background, a completed glossary can be provided so that they have additional time to understand the key terms with bilingual dictionaries. The glossary can be provided to students in their preferred communication mode.

## The design and production process

Throughout your study of Computing Technology, you will learn about design processes and how to apply them. You will explore different types of design processes and learn how to apply them in your design project.

The design and production process:

* involves a sequence of organised steps which provide a solution to design needs and opportunities
* may take a few seconds or minutes, such as when you select what clothes to wear, or may take years as in the case with the design of a motor vehicle
* may involve one person or may involve many people
* may be simple or complex, depending on the task
* involves questioning (or evaluating) throughout the iterative process.

Figure 1 – flowchart of design and production process

Design and production process diagram
A flowchart labelled 'Ongoing evaluation' with a two-headed arrow indicating both directions. 
The first part of the flowchart is called '1. Identifying and defining'. It says 'identify and define the needs, opportunities and wants of a computing challenge, practise the technical skills, develop evaluation criteria.' There is an arrow pointing to the next section, which is labelled '2. researching and planning'. It says 'research, generate and practise ideas, be creative and propose new approaches to problems, explore new design opportunities.' An arrow points to the next section, labelled '3. producing and implementing', it says 'build and implement ideas, apply a variety of skills and techniques to create products that meet set criteria, modify and iterate solutions'. The arrow points to the next section, labelled '4. testing and evaluating'. It says 'test and evaluate solutions/products, evaluate quality and effectiveness against the criteria, make judgements throughout the solution and use these to refine the product.'
After testing and evaluating is a big arrow called 'Review if required to improve' and it goes all the way back up to the first part of the flowchart, indicating a cycle.

## Identifying and defining

### Games and simulations

Define the following concepts in the space below.

What is a game?

|  |
| --- |
| **Sample answer:**  A game is a form of interactive entertainment that is typically characterised by rules, a goal, and a feedback system. The goal of a game is often to win or achieve a high score, and the feedback system is often used to inform the player of their progress towards this goal. |

What is a simulation?

|  |
| --- |
| **Sample answer:**  A simulation is a model of a real-world system that is used to study and predict the behaviour of that system. Simulations are often used in fields such as engineering, physics, and economics to test hypotheses and make predictions. |

What is the difference between a game and a simulation?

|  |
| --- |
| **Sample answer:**  The main difference between games and simulations is that games are designed for entertainment, while simulations are designed for study and prediction.  Games have a goal and a feedback system to inform the player of their progress towards that goal, while simulations are used to model and understand real-world systems. |

### Industry impact of games and simulations

Choose an industry and investigate the impact that gaming and simulation technology has had on that industry and how it is used in the real world.

How do these companies use augmented reality (AR), mixed reality (MR) and virtual reality (VR) to create immersive experiences?

You will need to research and gather information on the specific ways in which gaming and simulation technology is used within the industry, as well as any benefits or challenges that have arisen because of these technologies.

Consider the social, ethical, and legal responsibilities of using gaming and simulation within the chosen industry. Provide your findings in the space below.

|  |
| --- |
| **Sample answers:**   * Aerospace and defence: the use of simulations in training pilots and military personnel has been crucial in this industry for decades. * Construction: virtual reality (VR) simulations are increasingly being used to visualise and plan construction projects, allowing for more efficient and accurate planning. * Automotive: automotive companies use simulations to test the aerodynamics and performance of new car designs before they are built. * Healthcare: medical simulations are used to train surgeons and other medical professionals in a safe, controlled environment. * Energy: energy companies use simulations to model and optimise the performance of power plants, wind farms, and other energy systems. * Finance: financial institutions use simulations to model and test investment strategies, and to predict market trends. * Education: gaming and simulations are increasingly being used in education to make learning more engaging and interactive for students. * Advertising: VR and augmented reality (AR) are being used to create immersive advertising experiences for consumers. * Retail: VR and AR simulations are being used to create virtual stores, allowing customers to shop from the comfort of their own homes. |

### Perspectives of diverse groups in games and simulations

**Case Study: Blizzard Entertainment**

Overview: Blizzard Entertainment, a renowned game developer and publisher, is best known for developing *World of Warcraft*, *Diablo*, *Overwatch* and *StarCraft*. The company has made efforts to explore and incorporate the perspectives of diverse groups in their games and simulations. Below is how they addresses the criteria of inclusivity among Aboriginal and Torres Strait Islander peoples, culturally and linguistically diverse people, people of different ages and genders, and people with disabilities.

Research the following inclusivity initiatives relating to the company Blizzard, finding explicit examples and using the discussion questions to assess understanding perspectives of diverse groups in games and simulations.

|  |  |  |
| --- | --- | --- |
| Inclusivity Initiative | Example | Discussion questions |
| Cultural representation | *Overwatch* features characters from various cultural backgrounds, including characters like Hanzo and Genji Shimada, who are of Japanese descent, and Pharah and Baptiste who are of Egyptian and Haitian descent.  *World of Warcraft* introduced the Zandalari Trolls, a race inspired by various cultures, including elements that can be associated with Indigenous cultures. While not a direct representation of Aboriginal or Torres Strait Islander peoples, this example shows Blizzard's willingness to draw from a diverse range of cultural influences. | * How does Blizzard incorporate the perspectives of culturally diverse groups in games like *Overwatch*? * What impact does this have on player experiences? * How can game developers like Blizzard respectfully incorporate Indigenous perspectives into their games? * What steps can be taken to ensure cultural sensitivity and authenticity? |
| Gender diversity | Blizzard's character lineup includes diverse genders, such as the character Tracer, who identifies as LGBTQ+.  Sylvanas Windrunner from *World of Warcraft* and Kerrigan from *StarCraft* are strong, multifaceted female characters. | * How does Blizzard promote gender diversity and LGBTQ+ representation in their games? * What role do these diverse perspectives play in storytelling? |
| Accessibility features | Blizzard has implemented accessibility features in games like *World of Warcraft* to accommodate players with disabilities. These include colourblind mode and customisable UI elements. | * How does Blizzard address the needs of players with disabilities? * What are the benefits of incorporating accessibility features? |
| Language localisation | Blizzard localises their games in multiple languages, allowing players from linguistically diverse backgrounds to enjoy the gaming experience. | * How does language localisation contribute to the inclusivity of Blizzard's games? * What challenges might arise in ensuring effective localisation? |
| Representation of disabilities | Some Blizzard games include characters with disabilities. For instance, Mei in *Overwatch* uses a mobility-assist robot. | * How can game developers sensitively represent characters with disabilities in games? * What positive impact can this representation have on players? |

**Teacher note:** this activity could be a group task whereby teams are allocated the inclusivity initiative and then provided the discussion questions to create an infographic or presentation to communicate their understanding using examples of the case study company Blizzard Entertainment. The use of YouTube videos of the games could assist students with their understanding if the games/company are not known.

## The social impacts, and ethical and legal responsibilities in games or simulations

Outline how social impact, and ethical and legal responsibilities relate to games or simulations. Think about both the positive and negative for each, how the game should be and what the developer will need to consider.

**Teacher note:** sample answers are provided under the ‘Definition’ and ‘Explanation’ headings in the table below. These can be deleted when distributing to students as a resource.

|  |  |  |
| --- | --- | --- |
| Aspect | Definition | Explanation |
| Social impact | The effects that games can have on individuals and society, including how they influence behaviour, relationships, and communities. | Games can shape how people interact, communicate, and collaborate.  They can promote positive behaviours like teamwork and problem-solving, but they can also lead to issues like addiction and social isolation. |
| Ethical responsibility | Involves making moral choices and decisions while developing, playing, or engaging with games. | Game developers have a responsibility to create games that respect diverse cultures, avoid harmful stereotypes, and promote fair and inclusive gameplay.  Players also have ethical responsibilities, such as treating others with respect in online communities. |
| Legal responsibility | Rules and regulations set by governments or authorities that must be followed during game development, distribution, and play. | Games must adhere to laws related to age restrictions, content ratings, copyright, and data protection.  Breaking these laws can lead to legal consequences for both game developers and players. |

**Activity 1**: in groups, students read and discuss the gamified **ethical** dilemma scenarios. Students are required to look at both choices and as a group list the consequences for both choices from an ethical standpoint.

**Gamified ethical dilemma 1: The Lost Artifact**

**Scenario:** in an adventure game, the player's character stumbles upon a hidden, ancient artifact in a remote jungle. This artifact is rumoured to possess incredible powers and is of significant historical and cultural importance to the local Indigenous community. However, a powerful and unscrupulous artifact collector offers a substantial amount of in-game currency for it.

**Choices:**

1. Sell the artifact to the collector and gain a considerable amount of in-game wealth.
2. Return the artifact to the local Indigenous community and potentially receive their gratitude and assistance later in the game.

|  |
| --- |
| **Sample answer:**  **Consequences**   1. Selling the artifact to the collector may lead to wealth but also to negative consequences, such as a damaged reputation and potential in-game environmental repercussions. 2. Returning the artifact to the community may not provide immediate rewards but can lead to positive alliances and a sense of ethical accomplishment. |

**Gamified ethical dilemma 2: The Virtual Bullying Dilemma**

**Scenario:** in a virtual world-building game, players can interact with others and build their own virtual towns. The player's character witnesses another player repeatedly bullying and harassing a younger player, making the game environment uncomfortable and unwelcoming for the younger player.

**Choices:**

1. Ignore the situation and continue playing, not wanting to get involved.
2. Report the bully to the game’s moderators and stand up for the younger player, even if it means potential in-game conflict.

|  |
| --- |
| **Sample answer:**  **Consequences**   1. Ignoring the situation may lead to continued bullying and a hostile in-game environment. 2. Reporting the bully may result in the moderators taking action, potentially improving the game environment and fostering a sense of responsibility for others' well-being. |

**Activity 2**: in groups, students are assigned the role of game developers and they are employed to create a hypothetical game project while navigating ethical dilemmas that may arise during development.

Students are to read the scenario, consider the ethical dilemmas and answer the guiding questions as a team. Once completed report back to the class.

**Scenario 1: virtual environmental restoration**

**Game project**

Students are tasked with developing a simulation game in which players must restore a virtual environment that has been severely polluted. They must make ethical decisions about resource allocation, clean-up methods, and community involvement.

|  |  |
| --- | --- |
| Ethical dilemmas | Guiding questions |
| * Should players prioritise economic growth over environmental protection? * How should players balance short-term gains with long-term environmental sustainability? * What role should the virtual government and regulations play in the game? | 1. What are the ethical implications of prioritising economic growth over environmental protection? 2. How can the game encourage players to consider the consequences of their environmental decisions? 3. How can in-game consequences reflect the real-world impact of environmental choices? |

**Scenario 2: virtual health crisis management**

**Game project**

Students are challenged to create a game that simulates a health crisis, such as a pandemic or outbreak. Players must make ethical decisions regarding resource distribution, public health measures, and vaccine distribution.

|  |  |
| --- | --- |
| Ethical dilemmas | Guiding questions |
| * How should players balance individual rights with public health measures? * Is it ethical to prioritise certain groups, such as frontline workers, for limited resources like vaccines? * What role should misinformation and public panic play in the game? | 1. What ethical principles should guide the allocation of limited resources in a health crisis? 2. How can the game encourage players to think critically about the consequences of their decisions on public health? 3. How can the game address the challenges of misinformation and panic during a crisis? |

**Scenario 3: AI ethics in a virtual workplace**

**Game project**

Students are tasked with creating a simulation game set in a virtual workplace where players interact with AI colleagues. They must make ethical decisions regarding AI rights, job displacement, and AI-human collaboration.

|  |  |
| --- | --- |
| Ethical dilemmas | Guiding questions |
| * Should AI have rights and protections similar to human workers? * How should players address the potential job displacement caused by advanced AI? * What ethical considerations should govern the use of AI in the workplace? | 1. What ethical principles should guide interactions between humans and AI in a virtual workplace? 2. How can the game foster discussions about the ethical implications of AI in real-world workplaces? 3. How can the game explore the potential benefits and risks of AI-human collaboration? |

Research different countries’ regulations and rating systems for video games. Students are to use reliable sources such as Government websites or gaming industry associations.

The countries include the United States (ESRB), Canada (ESRB), Europe (PEGI), Australia (ACB) and Japan (CERO).

For each country in the table below, provide legal requirements, age ratings, and any censorship guidelines.

|  |  |  |  |
| --- | --- | --- | --- |
| Country | Legal requirements | Age ratings | Censorship guidelines |
| United States (ESRB) |  |  |  |
| Canada (ESRB) |  |  |  |
| Europe (PEGI) |  |  |  |
| Australia (ACB) |  |  |  |
| Japan (CERO) |  |  |  |

What are the consequences for game developers who do not adhere to these regulations in the specific countries?

|  |
| --- |
| **Sample answer:** |

How do these regulations impact players in terms of access to games and content?

|  |
| --- |
| **Sample answer:** |

## Understanding challenges through game play

By providing a safe and interactive learning environment, games and simulations can help players understand and address various challenges in a fun and engaging way.

Games and simulations can address various challenges in several ways listed in the table below.

|  |  |
| --- | --- |
| Challenge | Reason |
| Environmental challenges | Games and simulations can raise awareness about environmental issues and promote sustainable practices. For example, educational games can educate players about renewable energy sources and the consequences of carbon emissions. |
| Lifestyle challenges | Games can be used to promote healthy lifestyles and habits, such as exercise, healthy eating, and stress management. |
| Societal challenges | Games and simulations can be used to simulate social scenarios and decision-making processes, allowing players to experience the consequences of their choices in a safe and controlled environment. |
| Economic challenges | Economic simulations can be used to teach players about basic economic principles, such as supply and demand, and the consequences of macroeconomic policies. |

****As a class, students watch [Two Point Campus (1:38)](https://www.youtube.com/watch?v=_14i0Zxsfos).

*Two Point Campus* is a simulation game that focuses on managing a university campus. While it primarily centres around the management of an academic institution, it also incorporates various challenges including environmental, lifestyle, societal and economic challenges.

In groups, students research examples of how the game *Two Point Campus* addresses the challenges.

|  |  |  |
| --- | --- | --- |
| Challenge | Examples from the game | Discussion questions |
| Environmental challenges | **Sustainability initiatives**  Players can introduce eco-friendly campus features like solar panels and recycling centres. Balancing environmental concerns with the university's growth is a challenge.  **Landscaping and ecosystem preservation**  Decisions about campus layout can affect local ecosystems. Managing green spaces while expanding the campus can be an environmental challenge. | 1. How can incorporating sustainability initiatives into the campus design benefit both the virtual university and the real world? 2. What trade-offs exist between expanding the campus and preserving local ecosystems, and how can these be addressed? 3. What can we learn from games like *Two Point Campus* about the importance and impact of sustainability in real-world campus planning? |
| Lifestyle challenges | **Student well-being**  Maintaining students' mental and physical health is crucial. Balancing academic demands with recreational activities and ensuring students have access to health services is a lifestyle challenge.  **Time management**  Students and staff have schedules to adhere to. Ensuring that everyone's time is optimised while minimising stress is a lifestyle challenge | 1. How does the game balance the well-being of students and staff with the demands of running a successful university? 2. In what ways can virtual campus life mimic the challenges and benefits of real campus life? 3. What strategies can be employed in the game to optimise time management for both students and faculty? |
| Societal challenges | **Diversity and inclusion**  The game touches on societal challenges by incorporating diverse student and staff demographics. Promoting inclusivity and preventing discrimination can be part of gameplay. | 1. How does *Two Point Campus* promote diversity and inclusion within the virtual academic community, and why is this important in a gaming context? 2. What can game developers do to ensure that their games are representative of diverse backgrounds and experiences? 3. How might the societal challenges presented in the game relate to real-world issues of diversity and inclusion on university campuses? |
| Economic challenges | **Budget management**  Players must manage the university's budget. Balancing expenditures on faculty salaries, construction, and student services with income from tuition and research grants poses an economic challenge.  **Tuition and accessibility**  Setting tuition fees and scholarships can impact the accessibility of education. Striking a balance between affordability and financial sustainability is an economic challenge. | 1. How does managing the university's budget in the game reflect the financial considerations of real-world educational institutions? 2. What ethical considerations come into play when setting tuition fees and scholarships, and how do they impact access to education? 3. In what ways can games like *Two Point Campus* help players develop financial management skills that are applicable in real life? |

**Teacher note:** this activity links to the game and university campus, however, it could be modified to focus on school as the educational institution by rephrasing the question slightly. If using the game as a case study does not meet your school context, other real-world examples could be used.

## Understanding the main features of a system

The concept of inputs, storage, transmission, processes, and outputs in games and simulations refers to the fundamental elements that govern how these interactive experiences function.

* Inputs represent user actions or external data.
* Storage maintains the state and history of the system.
* Transmission deals with data exchange.
* Processes include computations and interactions within the system.
* Outputs manifest as visible or audible feedback to users.

Using your understanding of the game chess and a flight simulator, describe the inputs, storage, transmission, processes, and outputs that make up these products.

**Teacher note:** sample answers are provided under the ‘Description’ heading in the table below. These can be deleted when distributing to students as a resource.

**Chess**

|  |  |
| --- | --- |
| Main feature | Description |
| Input | Player moves: when playing chess online, players input their moves by interacting with the graphical user interface, selecting a piece, and specifying its destination square.  User login data: input includes the player's login credentials used to access the online chess platform. |
| Storage | Chessboard state: the current position of pieces on the chessboard is stored in a server database.  Player profiles: user data such as usernames, ratings, and game history are stored on the platform's server. |
| Transmission | Moves and data are transmitted between the player's device and the online chess server in real-time to maintain synchronisation between both players' boards.  Player challenges and invitations are transmitted to other players to initiate games. |
| Processes | Legal move validation: the online chess platform processes each move to ensure it adheres to the rules of chess and updates the board accordingly.  Matchmaking: the platform may employ algorithms to match players of similar skill levels for fair gameplay.  Chat and notifications: real-time messaging processes allow players to communicate and receive game-related notifications. |
| Output | Updated game state: the online chess platform visually displays the current position of pieces on the virtual chessboard for both players.  Notifications: messages or notifications are displayed for events like check, checkmate, or game completion.  Post-game analysis: after a game, players can access the outcome, a record of moves made, and performance statistics. |

**Flight Simulator**

|  |  |
| --- | --- |
| Main feature | Description |
| Input | Pilot controls: the inputs come from the pilot, who operates the control surfaces, throttle, and other cockpit elements.  Environmental data: Inputs include weather conditions, terrain data, and air traffic information. |
| Storage | Aircraft state: the simulator stores the state of the virtual aircraft, including its position, speed, altitude, and system statuses.  Flight history: data on the pilot's actions and the flight's progression may be stored for later review. |
| Transmission | Data may be transmitted to a central server for performance analysis or multiplayer interactions. |
| Processes | Physics simulation: the simulator processes the laws of physics to calculate how the aircraft responds to control inputs and environmental conditions.  Visual rendering: graphics processes render the virtual world, including the aircraft, terrain, weather effects, and more.  System modelling: the simulator models the behaviour of various aircraft systems, such as engines, avionics, and hydraulics. |
| Output | Visual feedback: the simulator provides a realistic visual representation of the flight, including the view from the cockpit.  Flight data: information such as airspeed, altitude, heading, and attitude is displayed to the pilot.  Training feedback: in training scenarios, feedback is provided to the pilot regarding their performance, errors, and areas for improvement. |

## Object oriented programming

Explain object-oriented programming.

|  |
| --- |
| **Sample answer:**  Object-Oriented Programming (OOP) is a way of organising and structuring code. It is based on the idea of objects, which are individual units of data and behaviour. Each object is an instance of a class, which is like a blueprint or template for creating objects |

List some examples of an object-oriented programming language in the space below.

|  |
| --- |
| **Sample answers include:**   * C++ * Java * C# * Python * Ruby * Swift. |

**Teacher note:** if you are unfamiliar with object-oriented programming principles in Python, utilise this resource – [Object Oriented Programming Getting Started for Australian Teachers.](https://sites.google.com/dltv.vic.edu.au/object-oriented) There are videos and activities that can be used to assist in explaining and for students to demonstrate their understanding.

Complete the table with features of an object-oriented programming language.

**Teacher note:** sample answers are provided under the ‘Use in object-oriented programming language’ heading in the table below. These can be deleted when distributing to students as a resource.

|  |  |
| --- | --- |
| Feature | Use in object-oriented programming language |
| Classes | The core concept of object-oriented programming is the class, which is a blueprint or template for creating objects. |
| Objects | Objects are instances of a class and have their own properties and behaviour. |
| Abstraction | Remove some characteristics from a class to have only necessary/essential properties and behaviours. |
| Encapsulation | An object-oriented language allows for encapsulation, which means that the data and behaviour of an object can be hidden from the rest of the program and only exposed through a set of defined interfaces. |
| Inheritance | Object-oriented languages support inheritance, which allows for the creation of new classes that inherit the properties and behaviour of existing classes. This allows for the reuse of existing code and the creation of a hierarchical class structure. |
| Polymorphism | Object-oriented languages support polymorphism, which allows objects of different classes to be treated as objects of a common superclass. This allows for more flexible and reusable code. |

### Features of object-oriented programming language activity

Using the game *Space Invaders* as reference, create class designs for the main game play. Think about how the parts of the game function, for example the player has lives and can move left and right.

|  |  |
| --- | --- |
| Space Invaders classes | Properties and behaviours |
| Player | Properties   * image/design * lives * firing speed   Behaviours   * move * fire * die |
| Aliens | Properties   * image/design * lives   Behaviours   * move * shoot |
| Barrier | Properties   * image/design * lives/hit points   Behaviours   * barrier break down |
| Bullet | Properties   * design/colour * speed/power   Behaviours   * movement |

### Representation of logic when designing games and simulations

Logic plays a crucial role in the design of games and simulations. It is used to control the flow of the game or simulation, determine the outcome of certain actions, and create realistic and engaging experiences.

****Teacher introduces the concepts of [Sequences, Selections and Loops (2:26)](https://www.youtube.com/watch?v=eSYeHlwDCNA). While watching the video, complete the definitions and questions for standard control structures, including examples where possible.

**Teacher note:** sample answers are provided under the ‘Definition’ heading in the table below. These can be deleted when distributing to students as a resource.

|  |  |
| --- | --- |
| Logic | Definition |
| Sequences | Sequences are a set of instructions that are executed in a specific order, one after the other. |
| Selection (branching) | Branching allows for the creation of different paths or outcomes based on conditions. |
| Iteration (loop) | Iteration is the repeated execution of a certain set of instructions, which can be used to create loops or cycles in the game or simulation. |

What is the benefit of the sequence control structure when designing games and simulations? List some restrictions of using sequences.

|  |
| --- |
| **Sample answer:**  Benefits: allows for the creation of a series of events that unfold over time as a progression, such as a character's journey through a level in a game or a simulation of a chemical reaction, easier to create, set and achieve specific game objectives.  Limitations: lack of flexibility and the game becoming too predictable and challenges in creating different level difficulties. |

What is the benefit of the selection control structure (branching) when designing games and simulations? List some restrictions of using selection.

|  |
| --- |
| **Sample answer:**  Benefits: empowers the player to make decisions giving them a sense of control over the outcome of the game, a more adaptive game play is provided, increase in diverse narrative and storytelling.  Limitations: developers need to consider the player choice impact on their programming and choices leading to complexities to manage and test. Limitations affect the length of the game which needs to be considered when pacing and player engagement extends over time. |

What is the benefit of the iteration control structure (loops) when designing games and simulations? List some restrictions of using loops.

|  |
| --- |
| **Sample answer:**  Benefits: allows for actions to be repeated multiple times like character movement or enemy behaviour, increases dynamic and interactive gameplay and provides continuous updates of the game world.  Limitations: could lead to bugs or errors if not designed correctly, increase in complexity of coding for beginners, for example nested loops; if not programmed correctly could lead to performance concerns. |

What are logical operators?

|  |
| --- |
| **Sample answer:**  Logical operators are used to combine multiple conditions and create more complex conditions. They are used to evaluate the truthfulness of certain statements and determine the outcome of certain conditions.  The 3 most common logical operators are:  AND (&& or and): The AND operator returns true if both of the conditions being compared are true.  OR (|| or or): The OR operator returns true if at least one of the conditions being compared is true.  NOT (! or not): The NOT operator negates the truthfulness of a statement.  These are the basic logical operators that are used in most programming languages. Basic logical operators allow us to join simple conditions and create more complex ones, which is important for creating realistic and engaging experiences in game and simulations. |

What are relational operators?

|  |
| --- |
| **Sample answer:**  Relational operators are used to compare values and determine the outcome of certain conditions. They are used to evaluate the relationship between 2 values and return a Boolean value (either true or false). The most common relational operators are:  Greater than (>)  The ‘greater than’ operator returns true if the value on the left side of the operator is greater than the value on the right side.  Less than (<)  The ‘less than’ operator returns true if the value on the left side of the operator is less than the value on the right side.  Greater than or equal to (>=)  The ‘greater than or equal to’ operator returns true if the value on the left side of the operator is greater than or equal to the value on the right side.  Less than or equal to (<=)  The ‘less than or equal to’ operator returns true if the value on the left side of the operator is less than or equal to the value on the right side.  Equal to (==)  The ‘equal to’ operator returns true if the value on the left side of the operator is equal to the value on the right side.  Not equal to (!= or <>)  The ‘not equal to’ operator returns true if the value on the left side of the operator is not equal to the value on the right side.  These relational operators are essential in making decisions in games and simulations, and are commonly used in control structures such as if-else statements and loops to determine the flow of the code. |

### Practical application – learning through coding

Students are to use the tutorial to learn to build [Tetris](https://simplegametutorials.github.io/pygamezero/blocks/) using IDLE & PyGame Zero which can be teacher led or student driven. The focus should be on students understanding why applying control structures and functions are important.

PyGame Zero is a beginner version to Pygame and allows from a smoother transition to programming. This can be run through IDLE if pgzero is pip installed. A [Flappy Bird](https://simplegametutorials.github.io/pygamezero/bird/) style game could also be used.

Examples for understanding logic in Tetris.

|  |  |
| --- | --- |
| Logic | Example from Tetris code |
| Sequences | def new_sequence():     global sequence     sequence = list(range(len(piece_structure)))    random.shuffle(sequence)  new_sequence() |
| Selection (branching) | if key == keys.X:   piece_rotation += 1   if piece_rotation > len(piece_structures[piece_type]) - 1:    piece_rotation = 0  elif key == keys.Z:    piece_roation -= 1    if piece_rotation < 0:         piece_rotation = len(piece_structures[piece_type]) - 1 |
| Iteration (loop) | for y in range(18):  for x in range(10):       block_size = 20          block_draw_size = block_size - 1 |

## Functional and non-functional requirements

Teacher introduces the concept of [functional vs non-functional requirements (1:36)](https://www.youtube.com/watch?v=zCX-N1H8Vps) in software design.

What are functional requirements in terms of software design?

|  |
| --- |
| **Sample answer:**  Functional requirements define what a software system should do and what features or capabilities it should have. |

What are non–functional requirements in terms of software design?

|  |
| --- |
| **Sample answer:**  Non-functional requirements define the quality attributes and performance standards for a software system. They describe how the software should operate, rather than what it should do. These requirements specify the constraints and conditions under which the software must function, such as response time, security, reliability, scalability, compatibility, and usability.  Non-functional requirements can impact the design of the software and often have trade-offs, as satisfying one requirement may negatively impact another. Examples of non-functional requirements include a maximum response time for a web page, the level of security required for sensitive data, or the need for the software to be scalable to handle increasing numbers of users. |

## Functional and non-functional requirements activity

Choose a game you would like to design and complete the table below to list its requirements. For each requirement, add a short explanation of why it is important for the game.

Description of chosen game:

|  |
| --- |
| **Sample answer:**  Racing game |

|  |  |
| --- | --- |
| Functional requirements | Non–functional requirements |
| Multiple cars | Smooth and responsive controls |
| tracks |  |
|  |  |
|  |  |
|  |  |

## Use case diagrams

Teacher introduces the concept of [use case diagrams (4:21)](https://www.youtube.com/watch?v=Omp4RbHbB0s) in software engineering and their purpose.

Answer the following questions in the space provided.

What are use case diagrams?

|  |
| --- |
| **Sample answer:**  Use case diagrams are a graphical representation of what a system must do.  Use case diagrams depict the interactions between the system, actors, and the different scenarios in which the system is used. A use case diagram provides a visual representation of the functionality offered by the system, and the different ways in which that functionality is used by the actors, or user roles, that interact with the system.  In a use case diagram, the system is represented as an oval, the actors are represented as stick figures, and the interactions between the actors and the system are represented as lines connecting the 2. Each use case is depicted as a rounded rectangle and represents a specific scenario or task that the system can perform. |

What is the purpose of use case diagrams?

|  |
| --- |
| **Sample answer:**  Use case diagrams are used in software development to help identify and understand the functional requirements of a system, and to ensure that the system provides the desired functionality to the actors who will use it. They are also useful for communicating the requirements to stakeholders and developers, and for testing and validation. |

Teacher introduces 2 examples of use case diagrams:

* [Vending Machine](https://creately.com/diagram/example/igdgb2ii1/use-case-diagram-of-a-vending-machine)
* [Classic Snake Game.](https://creately.com/diagram/example/ig93s3du/snake-game-use-case-diagram-classic)

Considering a game you are familiar with, create a use case diagram.

Outline the actor and key interactions that allow for the game to function. Some games could include *Snake*, *Space Invaders* or *Flappy Bird*.

|  |
| --- |
| **Sample answer:**  Basic information for a use case diagram for ***Space Invaders***:  **Actors:** Player – represents the person playing the game.  **Use Cases:**   * Start game – the player initiates a new game. * Move spaceship – the player can move their spaceship left and right to avoid enemy bullets and shoot at invaders. * Shoot bullets – the player can shoot bullets to destroy the invading aliens. * Destroy invaders – invaders can be destroyed by the player's bullets. * Avoid alien bullets – the player must avoid contact with bullets fired by the invaders. * Game over – the game ends when the player's spaceship is destroyed by an invader or when the player successfully defeats all invaders. * Pause game – the player can pause and resume the game.   Basic information for a use case diagram for ***Snake*:**  **Actors:** Player – represents the person playing the game.  **Use Cases:**   * Start game – the player initiates a new game. * Control snake – the player can control the direction of the snake using arrow keys or a controller. * Eat food – the snake can eat food items to grow longer. * Avoid collision – the snake must avoid colliding with the walls of the game board and its own body. * Score points – points are awarded to the player for each food item consumed. * Game over – the game ends when the snake collides with a wall or itself. * Pause game – the player can pause and resume the game. |

## Predefined algorithms and artificial intelligence

Play the game [Pac-Man](https://freepacman.org/) and become familiar with its features.

Teacher introduces [Pac-Man Ghost AI Explained (19:33)](https://www.youtube.com/watch?v=ataGotQ7ir8).

What are the 4 states of the ghosts in *Pac-Man*?

|  |
| --- |
| **Sample answer:**  Chase  Scatter  Eaten  Frightened |

When is the timer that controls phases reset?

|  |
| --- |
| **Sample answer:**  When the level is cleared or when *Pac-Man* dies. |

During the scatter and chase states, the ghosts use a targeting system to determine which direction to travel. Describe how a ghost decides to travel.

|  |
| --- |
| **Sample answer:**  As soon as they step onto a tile they will immediately choose a tile to move into from this point. The decision is between 3 tiles. The one directly in front of it, 90 degrees clockwise or 90 degrees counterclockwise. Turning around backwards 180 degrees is not an option. The one that is closest to the target tile linearly is the one the ghost will start moving to. |

## Design principles relevant to game design

Design principles in games are very important as they assist games in being created, played and impact on the player’s experience.

Design principles are used in combination depending on the type of game and the designer’s goal ensuring for an engaging and enjoyable gaming experience for all.

Use the space below to research the most common game design principles. Describe what the principle means in relation to gaming.

**Teacher note:** sample answers are provided under the ‘Description’ heading in the table below. These can be deleted when distributing to students as a resource.

|  |  |
| --- | --- |
| Game design principle | Description |
| Balance | Ensuring that gameplay is fair and that no single strategy or player has a significant advantage over others. It’s the unique balance of difficulty and fairness. |
| Pacing | Controlling the rhythm and tempo of the game to maintain player engagement and prevent boredom or frustration. |
| Player engagement | Keeping players interested and invested in the game by offering challenges, rewards, and a sense of progression. |
| Feedback/feedback loops | Providing clear and immediate feedback to players regarding their actions, progress, and the game's state that keep players engaged and motivated. |
| Immersion | Creating a game world or experience that draws players in and makes them feel like they are part of the game's universe. |
| Narrative and storytelling | Crafting a compelling narrative or story that drives the game's plot and motivates players to continue playing. |
| Flow | Achieving a state of flow where players are fully absorbed in the game, losing track of time and distractions. |
| Replayability | Designing games that encourage players to return and replay the game multiple times. |
| Reward Systems | Designing intuitive and user-friendly interfaces that make it easy for players to interact with the game. |
| Challenge | Gradually increasing the difficulty of the game to provide a sense of achievement and mastery. |
| Player Choice | Allowing players to make meaningful decisions that impact the game's outcome and story. |
| Simplicity | Striving for simplicity in game mechanics and rules to make the game easy to learn and understand. |
| Progression | Offering a sense of progression or advancement through the game, often through leveling up or unlocking new content. |

For each game, analyse how the game design principles can be applied to each. Research the relevant discussion points, which will help with some design principles but not all of them.

|  |  |  |
| --- | --- | --- |
| Game | Game design principles | Discussion points |
| Tetris | Balance, Pacing, Player Engagement, Feedback/Feedback loops, Simplicity, Challenge | 1. How does *Tetris* maintain a balance between difficulty and player engagement? 2. How does the game provide feedback to players, and how does its simplicity contribute to its appeal? |
| The Legend of Zelda: Breath of the Wild | Immersion, Player Engagement, Narrative and Storytelling, Player Choice | 1. How does the open world of *Breath of the Wild* contribute to player immersion? 2. How does the game's narrative encourage player exploration and choice? |
| Super Mario Bros | Pacing, Player Engagement, Flow, Reward Systems, Challenge, Feedback/Feedback Loops | 1. How does *Super Mario Bros* maintain a sense of flow and engagement for players? 2. How are rewards and challenges strategically placed to keep players motivated? |
| Portal | Pacing, Player Engagement, Simplicity, Challenge, Feedback/Feedback Loops, User Interface Design | 1. How does *Portal* maintain a balance between challenging puzzles and player engagement? 2. How does the game provide feedback to players, and how does its simplicity contribute to its appeal? |
| Minecraft | Player Choice, Progression, Replayability, Immersion | 1. How does the game's progression system motivate long-term play? 2. How does *Minecraft* empower players to make meaningful choices which can lead to replayability? |

**Teacher note:** this activity could be completed in pairs, with the teacher assigning a certain game to groups, with groups then presenting their findings to the discussion point questions. Additionally, other games could be chosen depending on the demographic of the class.

## Data privacy and cybersecurity in game design

[Brainstorm](https://app.education.nsw.gov.au/digital-learning-selector/LearningActivity/Card/542?clearCache=3d32c820-442f-7d68-d3ca-b2add7a710e8) why it is important to have data privacy and cybersecurity in today’s digital world. Students should think about their concerns about sharing personal information online.

|  |
| --- |
| **Sample discuss points:**   * Protection of personal information: safeguarding personal data such as names, addresses, and financial details is essential to prevent identity theft and fraud. * Preservation of individual rights: data privacy ensures that individuals have control over their personal information, respecting their right to privacy. * Prevention of unauthorised access: it helps in preventing unauthorised access to sensitive data, reducing the risk of data breaches. * Trust and reputation: businesses and organisations that prioritise data privacy earn the trust of customers and stakeholders, enhancing their reputation. * Protection against cyber threats: cybersecurity measures defend against a wide range of threats, including malware, ransomware, and phishing attacks. * Preservation of data integrity: it ensures the accuracy and reliability of data, preventing unauthorised changes or deletions. * Protection of intellectual property: it safeguards intellectual property, trade secrets, and proprietary information from theft and espionage. * Global connectivity: in an interconnected world, cybersecurity is vital for maintaining secure communications and collaborations across borders. * Personal safety: in the era of smart devices, cybersecurity is essential for preventing unauthorised access to connected vehicles, medical devices, and home automation systems, ensuring personal safety. * Ethical and legal obligations: cybersecurity compliance is required by various laws and regulations to protect individuals' and organisations' digital assets. |

It is important to note that not all apps and programs collect the same data, and users should carefully review an app's privacy policy or terms of service before downloading and using it to understand what data is being collected and how it will be used.

How do games/developers collect data?

|  |
| --- |
| **Sample answer:**  **User Profiles**   * **Username and email:** when players create accounts in online games, they provide their usernames and email addresses. This information is often used for authentication and communication. * **Player statistics:** games may collect data on player statistics, such as the number of wins and losses, achievements, and playtime. This data helps developers understand player preferences and skill levels.   **In-Game behaviours**   * **Game progress:** games often track a player's progress through levels or quests. This data helps developers identify where players might get stuck or lose interest. * **In-game purchases:** when players make in-game purchases, data related to the transactions, such as the items bought and the purchase amounts, is collected for financial and customisation purposes. * **Behavioural analytics:** games use analytics tools to collect data on how players interact with the game. This includes tracking which features are used most frequently and where players tend to drop off or abandon the game.   **Device information**   * **Device type and OS:** games collect data about the device being used to play, including the type of device (for example smartphone, PC, console) and the operating system (for example iOS, Android, Windows). This information helps optimise game performance for different platforms. * **Location data:** some games request access to a player's location for location-based features. For example, mobile games like *Pokémon GO* use GPS data to place virtual creatures in real-world locations. * **Hardware specifications:** game developers may collect data on the hardware specifications of the player's device, such as the CPU, GPU, and RAM. This helps in optimising graphics settings for the best performance.   **Social media integration**   * **Social media connections:** games often allow players to connect their social media profiles (for example Facebook, Twitter) for social interactions within the game. This integration can lead to the collection of additional user data and friend lists.   **Chat and communication data**   * **In-game chat:** if a game has chat features, the content of messages exchanged between players may be logged for moderation and security purposes.   **Feedback and surveys**   * **Player feedback:** games may prompt players to provide feedback or complete surveys. This data helps developers gather player opinions and improve the game. |

****Teacher introduces [Why this Data Expert Says TikTok Isn’t Safe (8:03)](https://www.youtube.com/watch?v=hZNHJNcO7BM).

Outline concerns that users should be aware of when using different apps?

|  |
| --- |
|  |

For each of the reasons below, explain why game developers would collect data from users when playing.

**Teacher note:** sample answers are provided under the ‘Explanation’ heading in the table below. These can be deleted when distributing to students as a resource.

|  |  |
| --- | --- |
| Reason | Explanation |
| Game improvement | Data collected from player interactions and behaviours within the game can provide valuable insights into how the game is played. Developers can use this information to identify areas that need improvement, such as levels that are too difficult or features that are rarely used. |
| Game balancing | Balancing a game to make it fair and enjoyable is a complex task. Data on player performance helps developers adjust game mechanics, difficulty levels, and character abilities to ensure a balanced and challenging experience. |
| Bug detection and fixing  performance optimisation | Data can reveal bugs, glitches, or technical issues that players encounter. Developers use this information to identify and address these problems, leading to a smoother and more bug-free gaming experience.  Data about device types, operating systems, and hardware specifications can guide developers in optimising game performance for different platforms. This ensures that the game runs smoothly on a wide range of devices. |
| Content customisation | Understanding player preferences through data allows developers to offer personalised content. This might include suggesting in-game items, characters, or challenges that align with a player's interests or play style. |
| Player engagement | Data analysis can uncover patterns related to player engagement, such as when players are most active, how long they play, and what motivates them to return to the game. Developers use this information to design events, updates, or rewards that keep players engaged. |
| Monetisation strategies (in-game purchases) | Data helps developers identify how players interact with in-game purchases or advertisements. This insight informs the design of monetisation strategies, ensuring they align with player preferences without compromising the gaming experience. |
| Community building (social media) | Games often foster online communities. Data can be used to understand player interactions within these communities, allowing developers to create forums, social features, or events that encourage positive engagement and camaraderie among players. |
| Research and innovation | Game data can contribute to research and innovation in game design, helping developers explore new gameplay mechanics, genres, or technologies based on player feedback and preferences. |

### Data ownership

When players engage with games, they generate various forms of data. This includes in-game actions, choices, progress, and even communication with other players. These interactions lead to the creation of valuable digital information. Game companies collect data from players. This data can range from player statistics and preferences to technical information about devices used for gaming. Game companies gather this data to improve their games, enhance player experiences, and inform business decisions.

[Brainstorm](https://app.education.nsw.gov.au/digital-learning-selector/LearningActivity/Card/542?clearCache=3d32c820-442f-7d68-d3ca-b2add7a710e8) who owns the data that is generated by players?

|  |
| --- |
| **Sample answer:**  **Shared ownership:** data generated during gameplay often resides in a shared space. Players contribute their actions and choices, while game companies collect and store this information. This shared ownership complicates the determination of sole ownership.  **Players’ rights:** players have a legitimate claim to data that directly reflects their in-game actions and choices. This data may include player profiles, progress, and achievements. Players often expect control over their personal gameplay data.  **Game company's rights:** game companies have a stake in the data they collect. They invest in the infrastructure and resources needed to maintain the game and its data, and they use this data to improve game quality, design updates, and personalise experiences for players. |

In the context of data ownership and data privacy in the gaming industry, ‘rights’ and ‘responsibilities’ are terms that define the roles and expectations of both players (users) and game companies.

For each of the scenarios below, in groups or as a class answer the discussion points about rights, responsibilities and ethical ways to handle data.

|  |  |
| --- | --- |
| Scenario | Discussion points |
| You're playing a popular online game, and it asks for your email address to create an account. The game developer says they will use it to send you updates and offers. Do you give them your email? | 1. What are the potential benefits of providing your email in this situation? 2. Potential benefits might include receiving updates about new game features, patches, and special offers. 3. What are the concerns associated with sharing your email address? 4. Concerns could involve receiving spam or unsolicited marketing emails, potential data breaches, or misuse of the email address. 5. How important is transparency in this scenario? 6. Transparency is crucial for informing players about how their email will be used and giving them the choice to opt-in or opt-out. 7. What would be an ethical way for the game developer to handle email data? 8. An ethical approach would involve obtaining informed consent, securing email data, and allowing players to easily unsubscribe from emails. |
| In a game, you notice it is tracking your in-game purchases and the items you collect. The developer says this helps them make the game more enjoyable for you. How do you feel about this? | 1. What potential advantages could tracking in-game purchases and collected items offer to players? 2. Advantages might include personalised game experiences, tailored rewards, and improved gameplay balance. 3. What privacy concerns might players have regarding the tracking of in-game data? 4. Concerns could involve data security, intrusive data collection, and potential misuse of player data. 5. How can game companies ensure players' trust when tracking in-game data? 6. Ensuring transparency, obtaining consent, and providing clear data protection measures can help build trust. 7. Should players have the option to opt out of data tracking? 8. Many players may prefer the option to opt out as it respects their autonomy and privacy. |
| You're playing a mobile game, and it asks for your location data to provide location-based features. It claims this will make the game more interactive. Do you allow it? | 1. What potential benefits can location-based features bring to gameplay? 2. Benefits might include real-world exploration, augmented reality experiences, and unique in-game events based on location. 3. What privacy and security concerns might arise when sharing location data? 4. Concerns could involve potential tracking of a player's real-world movements, data security, and the risk of data misuse. 5. How can game developers address the concerns of players when requesting location data? 6. Developers can address concerns through clear explanations of data usage, obtaining informed consent, and ensuring secure data handling. 7. Should players have the option to control when and how their location data is used in the game? 8. Allowing players to have control over location data sharing respects their privacy and preferences. |

### Data protection and cybersecurity

****Teacher introduces [Cyber Security in 7 Minutes (0:00-5:43).](https://www.youtube.com/watch?v=inWWhr5tnEA)

List the different attacks mentioned in the video that companies and individuals need to be aware of when it comes to cybersecurity.

|  |
| --- |
| **Sample answers:**   * Malware attack * Phishing attack * Man-in-the-middle attack * Password attack * Cyber security practices * Impact of a cyber attack * Advanced persistent threat (APT) * Denial of service attack & DDoS * SQL injection attack |

**Key areas of data protection and cybersecurity include:**

**Encryption –** the process of converting data into a code to prevent unauthorised access. It acts as a secure envelope around sensitive data, ensuring that even if it's intercepted, it remains confidential. In gaming, encryption is crucial for safeguarding sensitive player data, such as login credentials, payment information, and personal details. This ensures that even if a cybercriminal gains access to the data, they cannot decipher it without the encryption key.

**Authentication –** the verification process to ensure that users are who they claim to be. This includes strong passwords and multi-factor authentication which reduces the impact of a data breach and increases a user’s defence against phishing.

**Secure Storage –** storing data in a way that safeguards it from unauthorised access or data breaches. Some secure data storage practices and technologies include data encryption, access control, regular backups, data retention policies and employee training.

**The importance of safeguarding user data against breaches and unauthorised access:**

* Protecting user data builds trust and player confidence in the game and the game company.
* Data breaches can result in financial losses, damage to reputation, and legal consequences for game companies.
* Unauthorised access to personal information can lead to identity theft and other privacy violations for players.

Research the following real-world data breaches in the gaming industry. Provide an overview of what happened and discuss the consequences for both the user and the developer/company.

**Teacher note:** sample answers are provided under the ‘Overview’ and ‘Consequence’ headings in the table below. These can be deleted when distributing to students as a resource.

|  |  |  |
| --- | --- | --- |
| Breach | Overview | Consequence |
| Sony PlayStation Network Data Breach (2011) | Hackers accessed and stole personal data, including usernames, passwords, email addresses, and credit card information.  Users faced the risk of identity theft and fraudulent credit card charges. Sony experienced significant financial losses, including compensation to affected users and enhanced cybersecurity costs.  Severe damage to Sony's reputation, necessitating time and effort to rebuild trust.  Legal actions and regulatory investigations resulted from the breach. | **User consequences:**   * stolen personal data: hackers gained unauthorised access to Sony's PlayStation Network (PSN) and stole personal data, including usernames, passwords, email addresses, and credit card information * identity theft: users faced the risk of identity theft and fraudulent credit card charges * service disruption: the breach led to a 24-day service outage, preventing millions of gamers from accessing online services and playing games * loss of trust: users lost trust in Sony and its ability to protect their data.   **Developer/company consequences:**   * financial impact: Sony faced significant financial losses, including the cost of compensating affected users and enhancing cybersecurity measures * reputation damage: Sony's reputation was severely damaged, and it took time to rebuild trust among its user base * legal consequences: Sony faced legal actions and regulatory investigations due to the breach * increased security investments: the company had to invest heavily in improving its security infrastructure and practices. |
| Epic Games Data Breach (2021) | A security vulnerability in Epic Games' *Fortnite* led to potential exposure of user data.  Some users reported unauthorised access to their accounts, resulting in the theft of in-game items and virtual currency.  Epic Games responded promptly by addressing the breach, patching the vulnerability, and communicating with affected users. | **User Consequences:**   * stolen user data: in January 2021, a security vulnerability in Epic Games' *Fortnite* allowed hackers to access user accounts, potentially exposing personal information * account compromise: some users reported unauthorised access to their accounts, leading to theft of in-game items and virtual currency * security concerns: users became concerned about the security of their gaming accounts.   **Developer/company consequences:**   * response measures: Epic Games promptly addressed the breach, patched the vulnerability, and communicated with affected users * reputation impact: while not as severe as other breaches, the incident had some impact on Epic Games' reputation * security improvements: the company took steps to enhance its security practices to prevent similar incidents in the future. |

In groups, students will represent a game development company responsible for a popular online game. Provide each group with a scenario card. The group must research cybersecurity best practices and strategies and devise a plan to address the scenario effectively. Each group presents their scenario and response explaining their chosen strategies and the outcome.

|  |  |
| --- | --- |
| Scenario Cards | Scenario Cards |
| Incident 1: Server breach | **Incident 2: Phishing attack** |
| Description: a group of hackers has gained unauthorised access to your game's server. They claim to have accessed player data, including usernames and email addresses.  Potential impact: player data privacy compromised, potential for identity theft.  Timeline: incident detected within the last 24 hours. | Description: players are reporting receiving suspicious emails that appear to be from the game company, asking for their login credentials.  Potential impact: risk of account compromise, potential financial loss.  Timeline: reports of phishing emails have increased over the past week. |
| Incident 3: Distributed Denial of Service (DDoS) Attack | **Incident 4: In-Game cheat detection** |
| Description: your game's servers are under a sustained DDoS attack, causing service disruptions and lag for players.  Potential impact: game unplayable, loss of player trust.  Timeline: DDoS attack began 12 hours ago and continues. | Description: suspicious activity suggests that players are using cheats and hacks to gain an unfair advantage in your online game.  Potential impact: game integrity compromised, player frustration.  Timeline: reports of cheating have been increasing steadily over the past week. |
| Incident 5: Insider threat | **Incident 6: Ransomware attack** |
| Description: an employee with access to sensitive game code has resigned and taken a copy of the game's source code with them.  Potential impact: risk of intellectual property theft, possible game clones.  Timeline: employee left the company 2 days ago. | Description: a ransomware attack has encrypted critical game development files, and the attackers are demanding a ransom to decrypt them.  Potential impact: delayed game development, financial loss from ransom.  Timeline: attack occurred overnight, and the ransom demand was received this morning. |

## Representing data

In programming, variables are used to store and manipulate data, such as numbers, text, or other types of information. It represents a symbolic name or label used to store and manage data within a program or equation.

Variables have names (identifiers) that programmers assign to them, making it easier to reference and modify the data they contain. Variables can store a range of data types including Boolean, characters, strings, integers and floats.

****Teacher introduces [Why TRUE + TRUE = 2: Data Types (8:08)](https://www.youtube.com/watch?v=6otW6OXjR8c) and [What are data types (3:52)](https://www.youtube.com/watch?v=A37-3lflh8I).

Complete the table below by defining and outlining benefits and limitations of each listed data type.

|  |  |  |  |
| --- | --- | --- | --- |
| Data type | Definition | Benefits | Limitations |
| Boolean |  |  |  |
| Character |  |  |  |
| String |  |  |  |
| Integer |  |  |  |
| Float |  |  |  |

## Producing and implementing flowcharts

Examine the [various elements of a flow chart](https://www.tutorialspoint.com/programming_methodologies/programming_methodologies_flowchart_elements.htm) and their uses.

Complete the table below.

|  |  |  |
| --- | --- | --- |
| Symbol | Symbol name | Purpose |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

### Practical application – learning through coding

For a project currently being worked on, either [Tetris](https://simplegametutorials.github.io/pygamezero/blocks/), [Flappy Bird](https://simplegametutorials.github.io/pygamezero/bird/) or [Snake](https://simplegametutorials.github.io/pygamezero/snake/), students are to create flowcharts that represent a section or multiple sections of their code using the symbols in the previous activity.

Start with a sequence, then move through the different control structures and sections of code that will demonstrate students’ knowledge.

*PyGame Zero is a beginner version to Pygame and allows from a smoother transition to programming. This can be run through IDLE if Pgzero is pip installed.*

## Desk checking

As a class, walk through the below algorithms demonstrating how to conduct a desk check.

**Algorithm 1: simple addition – sequence**

# Start
# Input A and B (two numbers)
A = int(input("Enter the first number (A): "))
B = int(input("Enter the first number (B): "))

# Calculate the sum of A and B (C = A + B)
C = A + B

#Output C
Print ("The sum of A and B is:", C)

#Stop


|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sample answer:**  A=3; B=5   |  |  |  |  | | --- | --- | --- | --- | | **A** | **B** | **C** | **OUTPUT** | | 3 | 5 | 8 | The Sum of A and B is: 8 | |

**Algorithm 2: finding the maximum of 3 numbers – selection (branching)**

# Start
# Input three numbers: A, B, C 
A = int(input("Enter the first number (A): "))
B = int(input("Enter the first number (B): "))
C = int(input("Enter the first number (C): "))

# Check for the maximum
If A >= B and A>= C:
# Output A as the maximum
Print ("The maximum is A:", A)
Elif B >= A and B>= C:
# Output B as the maximum 
Print (The maximum is B:", B)
Else:
# Output C as the maximum 
Print (The maximum is C:", C)

# Stop

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sample answer:**  A=7; B=12; C=5   |  |  |  |  | | --- | --- | --- | --- | | **A** | **B** | **C** | **OUTPUT** | | 7 | 12 | 5 | The maximum is: 12 | |

**Algorithm 3: While loop to find the sum of N numbers – iteration (loop)**

# Start
# Input N (a positive integer)
N = int(input("Enter a positive integer (N): "))

# Initialise variables
sum_of_numbers = 0
count = 1

# While loop to find the sum of the first N numbers
while count <= N:
sum_of_numbers += count
Count += 1

# Output the sum
print ("The sum of the first", N, "number is:", sum_of_numbers)

# Stop

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sample answer:**  N=5; sum\_of\_numbers=0; count=1   |  |  |  |  | | --- | --- | --- | --- | | **N** | **sum\_of\_numbers** | **count** | **OUTPUT** | | 5 | 0 | 1 |  | |  | 1 | 2 |  | |  | 3 | 3 |  | |  | 6 | 4 |  | |  | 10 | 5 |  | |  | 15 | 6 | The sum of the first 5 numbers is: 15 | |

For all algorithms above, create a flowchart depiction.

## Error type in programming

* **Syntax errors** are detected by the compiler or interpreter during code compilation and prevent the program from running.
* **Logic errors** result from flawed program logic and lead to incorrect behaviour but don't cause program crashes.
* **Runtime errors** occur during program execution and can crash the program or produce error messages that need to be handled.

Complete the table below with definitions, examples, and the impact that error has on running code.

**Teacher note:** sample answers are provided under the ‘Description’, ‘Example’ and ‘Impact’ headings in the table below. These can be deleted when distributing to students as a resource.

|  |  |  |  |
| --- | --- | --- | --- |
| Errors | Description | Example | Impact |
| Syntax | Syntax errors, also known as compilation errors, occur when the code violates the rules of the programming language's syntax. These errors prevent the program from being compiled or interpreted. | * Missing semicolons or parentheses * Using undeclared variables * Incorrect capitalisation of function or variable names | Syntax errors are detected by the compiler or interpreter during the compilation or parsing phase. They prevent the program from being executed until fixed. Programs with syntax errors won't run at all. |
| Logic | Logic errors occur when the program's logic is flawed, resulting in unintended or incorrect behaviour. These errors do not cause the program to crash but lead to unexpected results. | * A mathematical calculation that produces the wrong result * A conditional statement that doesn't correctly evaluate a condition * A loop that doesn't terminate when it should | Logic errors can be challenging to detect because the program still runs. However, they can lead to incorrect output or behaviour, and debugging is required to identify and fix them. |
| Runtime | Runtime errors, also known as exceptions or run-time exceptions, occur when the program is running and encounters a condition that it cannot handle. These errors can lead to the program crashing or behaving unpredictably. | * Division by zero * Attempting to access an array element that doesn't exist * Trying to open a file that doesn't exist | Runtime errors can cause the program to terminate abruptly or produce error messages. They need to be handled through exception handling mechanisms to prevent program crashes. |

Using the algorithms below, spot the error and provide a correction for the algorithm. Including how the code will be changed.

|  |  |
| --- | --- |
| Sample algorithm | Spot the error |
| # Algorithm: Syntax Error Example  # Start # Input two numbers num1 = int(input("Enter the first number: ")) num2 = int(input("Enter the second number: "))  # Calculate their sum sum = num1 + num2  # Output the result print("The sum of", num1, "and", num2, "is:", sum) | The word sum is an in-built function and therefore, is a syntax error.  To fix this, a new variable name needs to be provided for the “sum” – could be changed to result. |
| # Algorithm: Logic Error Example  # Start # Input two numbers num1 = int(input("Enter the first number: ")) num2 = int(input("Enter the second number: "))  # Check if both numbers are positive if num 1 > 0 and num@> 0: Print ("Both numbers are positive.") else:  Print ("At lease one number is not positive.") # Stop | The if statement checks if the number is greater than 0 but doesn’t check if the number is 0 which is also a positive number.  To fix this, need to add = to both conditions to ensure 0 is checked too. |
| # Algorithm: Runtime Error Example  # Start # Input two numbers num1 = int(input("Enter the first number: ")) num2 = int(input("Enter the second number: "))  # Attempt to divide num1 by num2 result = num1 / num2  # Output the result print ("The result of the division is:", result) # Stop | If num2 is entered as zero then this algorithm will result in a ZeroDivisionError, therefore, a conditional statement if num2 is not equal to zero. |

### Practical application – learning through coding

Demonstrate different types of error in python using a current project. How does the IDE provide the students with the errors – ensure students are able to interpret the different errors they will encounter.

In the space below paste an error you encounter.

|  |
| --- |
|  |

## Gantt chart

****Teacher introduces [How to draw a Gantt chart (2:30)](https://youtu.be/NcOmJSrXYoQ).

What is a Gantt chart and what is it used for?

|  |
| --- |
|  |

In the space below, paste a Gantt chart for your project.

|  |
| --- |
|  |

## Record of project development

Plan and manage a project using an iterative approach. Keep a record of project development and video record your system model attempting processes as it evolves.

Use the following pages as a diary to document the development of your project. Make note of skills and knowledge gained, challenges faced and successes demonstrating iterative design and evaluation.

|  |  |
| --- | --- |
| Date | Description |
| \_\_/\_\_/\_\_ |  |
| \_\_/\_\_/\_\_ |  |
| \_\_/\_\_/\_\_ |  |
| \_\_/\_\_/\_\_ |  |
| \_\_/\_\_/\_\_ |  |
| \_\_/\_\_/\_\_ |  |
| \_\_/\_\_/\_\_ |  |
| \_\_/\_\_/\_\_ |  |
| \_\_/\_\_/\_\_ |  |

## Testing and evaluating

These sections of the syllabus should be completed through a practical project.

* Evaluate their own project and that of their peers using predetermined functional and non-functional requirements.
* Validate algorithms and programs through tracing and test cases.
* Test and evaluate the functionality and performance of a simulation or game for specified requirements.

Students can conduct peer assessment via the rubric attached to Assessment task 2 and evidence of testing through screenshots and explanations of how they have fixed the errors.

## Careers

Teacher introduces the [chosen career (playlist)](https://www.youtube.com/playlist?list=PLA8lfpEv0vOvo5f78SHdtb3UgAsWfYnpR) in the [video game industry (2:43)](https://www.youtube.com/watch?v=58JHaUtcrE0) for career research.

Identify the vast number of pathways into the industry, with a focus on tertiary education. Explain the competitive nature of the industry and prevalence of small start-up companies.

Explore careers in game and simulation design and development by completing the table below.

|  |  |
| --- | --- |
| Field | Career example |
|  |  |
|  |  |
|  |  |
|  |  |

Research 3 current existing positions and their criteria from current employment websites such as [SEEK](https://www.seek.com.au/), [Indeed](https://au.indeed.com/) and [CareerOne](https://www.careerone.com.au/).

|  |  |
| --- | --- |
| Criteria | Response |
| Job title |  |
| Training required |  |
| Personal requirements |  |
| Outline of duties |  |
| Average income |  |

|  |  |
| --- | --- |
| Criteria | Response |
| Job title |  |
| Training required |  |
| Personal requirements |  |
| Outline of duties |  |
| Average income |  |

|  |  |
| --- | --- |
| Criteria | Response |
| Job title |  |
| Training required |  |
| Personal requirements |  |
| Outline of duties |  |
| Average income |  |

## References

This resource contains NSW Curriculum and syllabus content. The NSW Curriculum is developed by the NSW Education Standards Authority. This content is prepared by NESA for and on behalf of the Crown in right of the State of New South Wales. The material is protected by Crown copyright.

Please refer to the NESA Copyright Disclaimer for more information <https://educationstandards.nsw.edu.au/wps/portal/nesa/mini-footer/copyright>.

NESA holds the only official and up-to-date versions of the NSW Curriculum and syllabus documents. Please visit the NSW Education Standards Authority (NESA) website <https://educationstandards.nsw.edu.au/> and the NSW Curriculum website <https://curriculum.nsw.edu.au/home>.

[Computing Technology 7–10 Syllabus](https://curriculum.nsw.edu.au/syllabuses/computing-technology-7-10-2022) © NSW Education Standards Authority (NESA) for and on behalf of the Crown in the State of New South Wales, 2022.

CareerOne Pty Ltd (2022) [*CareerOne*](https://www.careerone.com.au/)[website], accessed 4 September 2023.

Creately (2023) [*Snake Game Use Case Diagram [classic]*](https://creately.com/diagram/example/ig93s3du/snake-game-use-case-diagram-classic), Creately website, accessed 4 September 2023.

Creately (2023) [*Use Case Diagram of a Vending Machine*](https://creately.com/diagram/example/igdgb2ii1/use-case-diagram-of-a-vending-machine), Creately website, accessed 4 September 2023.

Digital Learning and Teaching Victoria (n.d.) [*Object Oriented Programming Getting Started for Australian Teachers*](https://sites.google.com/dltv.vic.edu.au/object-oriented), Digital Learning and Teaching Victoria website, accessed 4 September 2023.

Digital Schoolhouse (n.d.) [One Minute Mentor (OMM) playlist](https://www.youtube.com/playlist?list=PLA8lfpEv0vOvo5f78SHdtb3UgAsWfYnpR), YouTube website, accessed 4 September 2023.

Engineer4Free (23 December 2014) ['How to draw a Gantt chart' [video]](https://youtu.be/NcOmJSrXYoQ) *Engineer4Free*, YouTube, accessed 4 September 2023.

GCFLearnFree (4 October 2018) ‘[Computer Science Basics: Sequences, Selections and Loops](https://www.youtube.com/watch?v=eSYeHlwDCNA)’, *Computer Science Basics*, YouTube, accessed 4 September 2023.

Indeed (2023) [*Indeed*](https://au.indeed.com/) [website], accessed 4 September 2023.

Inside Edition (14 May 2020) ['Why This Data Expert Says TikTok Isn't Safe’ [video]](https://www.youtube.com/watch?v=hZNHJNcO7BM), Inside Edition, YouTube, accessed 4 September 2023.

Jabrils (9 July 2019) ‘[What Are Data Types’ [video]](https://www.youtube.com/watch?v=A37-3lflh8I), *Jabrils*, YouTube, accessed 4 September 2023.

Retro Game Mechanics Explained (14 July 2019) ‘[Pac-Man Ghost AI Explained’ [video]](https://www.youtube.com/watch?v=ataGotQ7ir8) , *Retro Game Mechanics Explained*, YouTube, accessed 4 September 2023.

Scott T (4 February 2020) ‘[Why TRUE + TRUE = 2: Data Types’ [video]](https://www.youtube.com/watch?v=6otW6OXjR8c), *Tom Scott*, YouTube, accessed 4 September 2023.

SEEK Limited (n.d.) [*SEEK*](https://www.seek.com.au/) [website], accessed 4 September 2023.

Simple Game Tutorials (n.d.) [*Bird Tutorial for Python and Pygame Zero 1.2*](https://simplegametutorials.github.io/pygamezero/bird/), Simple Game Tutorials website, accessed 4 September 2023.

Simple Game Tutorials (n.d.) [*Blocks Tutorial for Python and Pygame Zero 1.2*](https://simplegametutorials.github.io/pygamezero/blocks/), Simple Game Tutorials website, accessed 4 September 2023.

Simple Game Tutorials (n.d.) [*Snake Tutorial for Python and Pygame Zero 1.2*](https://simplegametutorials.github.io/pygamezero/snake/), Simple Game Tutorials website, accessed 4 September 2023.

Simplilearn (11 June 2020) ‘[Cyber Security in 7 Minutes’ [video]](https://www.youtube.com/watch?v=inWWhr5tnEA), *Simplilearn*, YouTube, accessed 4 September 2023.

SmartDraw (22 June 2019) ‘[All About Use Case Diagrams – What is a Use Case Diagram, Use Case Tutorial and More’ [video]](https://www.youtube.com/watch?v=Omp4RbHbB0s), *SmartDraw*, YouTube, accessed 4 September 2023.

The Fee Video Games Project (2005–2003) [*Pac-Man*](https://freepacman.org/), Free Pac-Man.org, accessed 4 September 2023.

Tutorials Point (India) Private Limited (n.d.) [*Flowchart Elements*](https://www.tutorialspoint.com/programming_methodologies/programming_methodologies_flowchart_elements.htm), Tutorials Point website, accessed 4 September 2023.

Two Point Campus (11 June 2021) ‘[Two Point Campus’ [video]](https://www.youtube.com/watch?v=_14i0Zxsfos)*, Two Point Campus*, YouTube, accessed 4 September 2023.

Ubisoft North America (17 September 2020) ‘['How to Start a Career in Video Games' [video]](https://www.youtube.com/watch?v=58JHaUtcrE0)’, *Ubisoft North America*, YouTube, accessed 4 September 2023.

Udacity (24 February 2015) ‘[Functional and Nonfunctional Requirements’ [video]](https://www.youtube.com/watch?v=zCX-N1H8Vps), *Udacity*, YouTube, accessed 4 September 2023.

**© State of New South Wales (Department of Education), 2023**

The copyright material published in this resource is subject to the *Copyright Act 1968* (Cth) and is owned by the NSW Department of Education or, where indicated, by a party other than the NSW Department of Education (third-party material).

Copyright material available in this resource and owned by the NSW Department of Education is licensed under a [Creative Commons Attribution 4.0 International (CC BY 4.0) license](https://creativecommons.org/licenses/by/4.0/).

[](https://creativecommons.org/licenses/by/4.0/)

This license allows you to share and adapt the material for any purpose, even commercially.

Attribution should be given to © State of New South Wales (Department of Education), 2023.

Material in this resource not available under a Creative Commons license:

* the NSW Department of Education logo, other logos and trademark-protected material
* material owned by a third party that has been reproduced with permission. You will need to obtain permission from the third party to reuse its material.

**Links to third-party material and websites**

Please note that the provided (reading/viewing material/list/links/texts) are a suggestion only and implies no endorsement, by the New South Wales Department of Education, of any author, publisher, or book title. School principals and teachers are best placed to assess the suitability of resources that would complement the curriculum and reflect the needs and interests of their students.

If you use the links provided in this document to access a third-party's website, you acknowledge that the terms of use, including licence terms set out on the third-party's website apply to the use which may be made of the materials on that third-party website or where permitted by the Copyright Act 1968 (Cth). The department accepts no responsibility for content on third-party websites.