Unit 77: Designing Tests for Computer Games

Unit code: H/600/6614

QCF Level 3: BTEC National

Credit value: 10

Guided learning hours: 60

Aim and purpose

The aim of this unit is to provide the knowledge and skills needed to effectively design and implement test cases and test suites for computer games.

Unit introduction

The success of each game title is not only dependent on how 'good' the game is in terms of playability, its look and feel, but also how well it performs on the platforms that it has been designed for.

There are a variety of bugs that can cause problems within a game. Learners who complete this unit will gain a clear understanding of the types of defects that can occur in a game, together with an understanding of where these bugs will be found.

If a game is shipped to the consumer still containing major bugs or defects that hinder the players' enjoyment of the product then the reputation of not only the game or brand can be affected but also that of the company which developed and tested it. Many games rely on loyal fan bases and gaming communities who are quick to share and spread opinions as well as games cheats and short cuts, all of which can affect how profitable a game ultimately is. Game testing is therefore a crucial part of the games production process.

It seems hard to believe that people are actually paid to sit down and play games but, although it is sometimes fun, effective games testing is hard work. During the development phase games need to be constantly and methodically checked to ensure that any problematic bugs or defects are identified and documented in a concise and explanatory fashion. Once documented, bugs are fixed by the developer and the game is then retested or verified by the tester. By the time the game goes on sale, all major bugs should have been eliminated.

This unit aims to provide learners with the knowledge and skills to effectively design test cases and test suites ready for computer games testing. Learners will be taught some of the technical theory and testing methodologies that lie behind effective computer testing. This underlying theory needs to be understood before learners apply it to the design and creation of test suites of their own.

Learning outcomes

On completion of this unit a learner should:

1 Understand types of defects present within a game build

2 Understand test phases and processes within a test programme

3 Be able to design a test suite for use with a game build or module

4 Be able to produce a bug report using a test suite.

Unit content

1 Understand types of defects present within a game build

Defect types: function; assignment; checking; timing; build package merge; algorithm; documentation

Defect triggers: orthogonal defect classification (ODC); hardware configurations; software configurations; configuration trigger; start-up trigger; exception handling; exception trigger; stress trigger; normal trigger; restart trigger

Operating regions: game operation; pre-game; game start; in-game; post-game

2 Understand test phases and processes within a test programme

Test phases: preparation; alpha testing; closed beta testing; open beta testing; gold testing; entry criteria; placeholder assets; regression

Processes: smoke testing; white box testing; black box testing; clean room testing; play testing; ad hoc testing; player types; minor patch; major patch

Life cycle: plan and design test; prepare test; perform test; report test; retest

3 Be able to design a test suite for use with a game build or module

What to test: game events; game settings; gameplay options; hardware configurations; character attributes; asset operation; customisation choices

Combinatorial testing: parameters; pairwise; values; ranges; boundaries; combinatorial tables; combinatorial templates; combinatorial tools

Test flow diagrams: test flow diagram (TFD) elements; game flows; actions; events; states; primitives; terminators; design; paths; data dictionary definitions; templates; software used

4 Be able to produce a bug report using a test suite

Test status: test pass; test fail; blocked; not available; priorities; versions

Supporting material: artefacts, eg taped play/playtest video capture, screenshots, storyboards, sound files, server logs, operating system error codes

Writing bug reports: report content (headline/summary, test status, severity, full description, reproducible, steps to reproduce, supporting artefacts); writing style (clarity, concise writing, descriptive, instructional)

Assessment and grading criteria

In order to pass this unit, the evidence that the learner presents for assessment needs to demonstrate that they can meet all the learning outcomes for the unit. The assessment criteria for a pass grade describe the level of achievement required to pass this unit.

Assessment and grading criteria					
To achieve a pass grade the evidence must show that the learner is able to:		**To achieve a merit grade the evidence must show that, in addition to the pass criteria, the learner is able to:**		**To achieve a distinction grade the evidence must show that, in addition to the pass and merit criteria, the learner is able to:**	
P1	describe the types of defects that can be present within a game build with some appropriate use of subject terminology [IE]	**M1**	explain the types of defects that can be present within a game build with reference to detailed illustrative examples and with generally correct use of subject terminology	**D1**	critically analyse the types of defects that can be present within a game build with supporting arguments and elucidated examples, consistently using subject terminology correctly
P2	describe the different test phases and processes within a test programme with some appropriate use of subject terminology [IE]	**M2**	explain the different test phases and processes within a test programme with reference to detailed illustrative examples and with generally correct use of subject terminology	**D2**	critically assess the different test phases and processes within a test programme with supporting arguments and elucidated examples, consistently using subject terminology correctly
P3	design a test suite for use with a game build or module with some assistance [CT]	**M3**	design a test suite for use with a game build or module to a good technical standard with only occasional assistance	**D3**	design a test suite for use with a game build or module to a technical quality that reflects near-professional standards, working independently to professional expectations
P4	generate a bug report using a test suite with some appropriate use of subject terminology. [SM]	**M4**	generate a detailed and illustrated bug report using a test suite to a good technical standard and with generally correct use of subject terminology.	**D4**	generate a fully detailed and thoroughly illustrated bug report using a test suite to a technical quality that reflects near-professional standards and consistently using subject terminology correctly.

PLTS: This summary references where applicable, in the square brackets, the elements of the personal, learning and thinking skills applicable in the pass criteria. It identifies opportunities for learners to demonstrate effective application of the referenced elements of the skills.

Key	IE – independent enquirers	RL – reflective learners	SM – self-managers
	CT – creative thinkers	TW – team workers	EP – effective participators

Essential guidance for tutors

Delivery

This unit has been designed to provide learners with the theoretical knowledge needed for designing test cases and test suites (sometimes known as test sheets) for computer games, and to develop their ability to apply them. To make the learning more relevant it is suggested that it is taught within the context of a broader games development course where this knowledge can be applied to games that learners are studying or creating and testing.

It is suggested that teaching follows the order of the learning outcomes, commencing with an introduction to why and how defects and bugs occur within computer games. A combination of teaching techniques could be used including formal lectures, practical activities and discussion.

A combination of formal teaching and personal investigative study can be used to develop understanding of the different test phases and processes within a test programme. To add authenticity and industry relevance, learners should be encouraged to gain some of this knowledge first hand from games companies and practitioners. This can be achieved through visits to games development companies, project-based research and bringing in guest speakers.

In order that learners can fully appreciate what needs be considered when developing tests for computer games, it is important for them to understand what elements need to be tested and are included for testing. This can be achieved through a combination of formal teaching techniques and exploratory gameplay. Again, this learning can be based upon already published titles or work that has been produced as part of a broader base of study. Learners can then apply their understanding to designing test cases and suites.

It is important that learners have some experience of designing computer tests using both combinatorial tables and test flow diagrams. Although specialist software can be used to help design tests for use with computer games, it is important that learners understand the underlying methods that are used. It is recommended that techniques for generating combinatorial tables and test flow diagrams are taught as practically as possible, reserving an introduction to specialist software until core concepts have been fully understood.

When designing tests learners need to be aware of the kinds of parameters that can be covered within a test together with their possible values. Both combinatorial tables and test flow diagrams can become very complex and it is suggested that delivery uses a simple yet thorough approach so learners have good opportunities to understand the underlying concepts. Learners should practise combinatorial and test flow drawing techniques in small-scale exercises before looking at larger sections of a game build.

Existing games titles on any platform can be used as practice for writing test cases and reports and for learning the formats and required levels of content. Tutors may wish to use free online games or virtual communities to form the basis of some of learners' tests. Where institutions have good relationships with industry, it may be possible to base bug tests on donated builds of old titles. Opportunities exist to use learner-developed games from other units as test subjects and in this way the unit may be linked to work done for other units, or at a different level.

Learners should use the test sheets they have designed to produce a series of detailed bug reports to help evaluate their test design. To replicate the environment of a test department, learners could be encouraged to swap test sheets and use each others' tests for the basis of their bug reports.

It is important that learners understand the importance of concise and grammatically correct writing when producing reports. Learners should be encouraged to illustrate their findings by using such techniques as capturing screenshots and incorporating them into storyboards with written explanations. Tutors are encouraged to provide learners with bug reporting forms based on those provided for input to industry-standard bug database software. If the software itself cannot be provided, tutors may wish to source sample screenshots of bug reporting forms from software websites. These screenshots can then be used to develop word-processed or portable document format forms and templates for completion by learners. The introduction of these formal layouts can enhance the simulation of a professional quality assurance environment.

Outline learning plan

The outline learning plan has been included in this unit as guidance and can be used in conjunction with the programme of suggested assignments.

The outline learning plan demonstrates one way of planning the teaching and assessment of this unit.

Topics and suggested assignments and activities
Introduction to unit and unit assessment.
Introduction to defect types and triggers. Learners will: • attend lectures and tutor-led demonstrations explaining defect types • attend lectures and tutor-led demonstrations explaining defect triggers • attend lectures and tutor-led demonstrations explaining operating regions • conduct private study researching defects in games and practising identification of their defect types, their triggers and the operating regions in which they occur.
Assignment 1 – Game Defect Types: What You Need to Know Learners will write an article for an online game fan ezine explaining game defect types and triggers. Article must cover: • defect types • defect triggers • operating regions.
Introduction to test phases and processes. Learners will: • attend lectures explaining phases of game testing • attend lectures and tutor-led demonstrations explaining processes involved in game testing • attend lectures explaining the game testing life cycle • conduct private study researching game test phases and processes.
Assignment 2 – Game Test Phases and Processes: What You Need to Know Learners will write an article for an online game fan ezine on game test phases and processes. Article must cover: • test phases • processes • life cycle.

Topics and suggested assignments and activities
Introduction to test suite design. Learners will: • attend lectures explaining what should be tested during a computer game's development • attend lectures and complete directed practical work examining combinatorial testing • attend lectures and complete directed practical work examining and constructing test flow diagrams • conduct personal study researching combinatorial tables, identifying useful combinatorial templates and developing skills in constructing test flow diagrams.
Assignment 3 – My Test Suite Exercise on designing test cases and creating a test suite to test a game build effectively. Learners will prepare a test suite covering: • use of pairwise testing (combinatorial tables) to select test areas to check game settings and game options • drawing test flow diagrams and use of these to select efficient tests to check game states, repeatable functions and operational flows • creation of a data dictionary holding definitions for each primitive in the test flow diagrams • writing efficient test cases for each test identified from the combinatorial tables and test flow diagrams, and combining them to form a comprehensive well-written test suite to check performance of the given game build.
Introduction to bug report writing. Learners will: • attend lectures explaining the various test status descriptors • attend lectures and complete directed practical work examining and practising how to generate artefacts to support comments made in bug reports • attend lectures and complete directed practical work examining and practising the writing of accurate helpful bug reports following test cases • conduct personal study practising authoring illustrated bug reports.
Assignment 4 – My Bug Reports Exercise on authoring a comprehensive set of illustrated bug reports following the test suite created for Assignment 3. Learners will prepare a portfolio of illustrated bug reports containing: • systematic and extensive coverage of the test suite • comprehensive bug reports generated from following the test cases of the test suite • artefacts to illustrate points made in the bug reports • all reports and artefacts combined into a single summary document.
Unit learning and assessment review.

Assessment

Evidence for assessment

For assessment of achievement of learning outcome 1 learners could present any new-found or researched information. This could be done via a presentation, report, blog or vlog that explains the types of defects that can be found in a game, where they can be found and what causes them to occur in a game build. Research may include extracts from books, trade magazines, internet articles or information from practitioners working within the field. Found knowledge can be expanded upon and illustrated by using primary resources such as screen grabs and photographs. Learners' understanding of technical knowledge and terminology should be included.

Evidence for learning outcome 2 could also take the form of a report, blog, vlog or presentation. Again, learners may include extracts from books, magazines and the internet but this learning outcome also provides the ideal opportunity for them to approach games testing companies directly to develop an understanding of current industry practice.

For learning outcomes 3 and 4 learners should be provided with a game build known to contain defects, upon which to base their test suite. Tutors may wish to use virtual communities to identify beta versions to form the basis of learners' tests. Where institutions have good relationships with industry, it may be possible to base bug tests on donated builds of old titles. Opportunities exist to use learner-developed games from other units as test subjects.

For learning outcome 3 learners must produce documentation that shows their effective planning and understanding of designing tests for computer games. Background documentation that demonstrates an understanding of underpinning knowledge should be produced together with final, well-produced word-processed and near-professional test sheets. This documentation may be presented either on paper or electronically.

Evidence for achievement of learning outcome 4 should be in the form of bug reports based on the tests that have been designed. Illustrations demonstrating bugs or defects that are found can be included by using such methods as screen grabs, audio files, storyboards etc. Learners should be encouraged to provide supporting information describing what they were doing during gameplay at each stage of the test.

For some elements of these units, and for some learners, a formal viva voce assessment might be appropriate. When more than one learner in a cohort is assessed in this way, care must be taken to ensure that all learners are asked equivalent questions, and that all are given equal opportunities to expand or clarify their answers. Interviewers must also ensure that questions are not phrased in such a way as to provide or suggest an answer. Formal vivas should be recorded for the purposes of internal and external verification and at least 50 per cent of such assessments must be internally verified.

Application of grading criteria

When applying the grading criteria, tutors should follow the advice given below. Please note that any examples of evidence given here are indicative only. This advice is not exhaustive and the examples need not specifically be included in a learner's work in order for that learner to achieve the exemplified grade.

Pass

To achieve a pass grade, learners must achieve all the criteria at pass level. For each of the criteria learners must present evidence that addresses each italicised sub-heading of the content for the learning outcome.

P1: learners will name the types of bugs and defects that can be found within a game build. They will name the operating regions within a game and some of the defect triggers that can span the game operating regions. Descriptions will be correct and although learners may know the names of the bugs and defects that can be found they will not describe exactly what causes them and why they materialise. When discussing a defect a pass grade learner might comment, 'Defects are bad in games because they make the game go wrong when you are playing it. A type of defect that you can find when testing games is a documentation type defect.'

P2: learners will name the different stages of a test programme in an appropriate order. They will describe what happens within each of the phases. At this grade learners will describe different testing processes that are used within the industry and describe the typical life cycle of a games test. When describing the stages of a test programme a pass grade learner might comment, 'There are different stages when testing games. Some of these stages are called smoke testing, alpha testing and beta testing.'

P3: learners will produce designs for a games test evidencing the use of both combinatorial tables and test flow diagrams. The tests designed will be of a basic nature and should consider at least one parameter or element of a game tested against two values. When designing the tests learners may have needed guidance on what exactly should be tested.

P4: learners will produce a bug report that describes problems found, using basic terms. The bug report will be structured and include sections such as headings and summary descriptions, and bug descriptions of the defect they have discovered. They will describe what they were doing in the game when the defect was noted. The bug report will be illustrated to support some of the issues identified.

P1, P2 and P4: evidence will show a basic understanding of technical terminology but learners will generally be unsure about this vocabulary and will make fairly frequent mistakes when they do use it.

Merit

To achieve a merit grade, learners must achieve all the pass and all the merit grade criteria. For each of the criteria learners must present evidence that addresses each italicised sub-heading of the content for the learning outcome.

M1: learners will name correctly the types of bugs and defects that can be found within a game build, identify them and explain what they are and how they may have been introduced. They will describe what the different operating regions within a game are, giving examples of what happens in each region. They will describe the range of defect triggers and be able to illustrate them with examples from games they have seen. When discussing a defect a merit grade learner might comment, 'Defects are often introduced into games because of an error in the code that has been produced. There are various types of defects that can be found when testing games. One defect is known as a documentation type defect and this type of defect occurs in such content as text audio and graphics.'

M2: learners will name the different stages of a test programme, explaining accurately what happens within each test phase. They will describe what types of activity happen at each phase and say why they are important. They will describe the purpose of each phase and identify the kinds of criteria that indicate when each phase is seen to be completed. Learners will describe the different testing processes and explain why different processes may be used. When describing the different stages of a test programme a merit grade learner might comment, 'In order that a game is tested effectively there are different stages within the test programme. The test programme can typically begin with a preparation phase for the testers where they can become familiar with both the game that they are about to test and the hardware. Testers can use different methods to test games. One method is called black box testing where testers can test games by playing the game using the normal input devices as a player.'

M3: learners will create a sound and valid test suite using both combinatorial tables and test flow diagrams. They will identify what parameters and values can be tested, requiring only occasional guidance.

M4: learners will produce a detailed bug report that is well structured and includes most of the sections and criteria found in professional work. Brief descriptions will clearly identify the defects found. Any bugs or defects that have been found will be, on the whole, concisely described. When and where bugs were identified will be clearly described with a variety of illustrations. Full descriptions will be written in such a way that any instructions to re-create a bug can be understood with little difficulty. The bug report will be meaningfully supported with a number of illustrations generated by screen grab, videotaping gameplay or other techniques.

M1, M2 and M4: learners will use technical vocabulary for the most part correctly, but will at times make mistakes or be unsure about usage.

Distinction

To achieve a distinction grade, learners must achieve all the pass, all the merit and all the distinction grade criteria. For each of the criteria learners must present evidence that addresses each italicised sub-heading of the content for the learning outcome.

D1: learners will name and comprehensively describe the types of bugs or defects that can be found within a game build showing a confident understanding of what these bugs are and how they could have been introduced. They will show a comprehensive understanding of the different operating regions within a game giving a full description of what happens in each region. Learners will explain each defect trigger using technical terminology and giving examples of how these may manifest themselves when playing a game. A distinction grade learner might comment, 'As a result of badly written or missing code, computer games may not function properly for the player. There are different types of defects that may occur within a game build and these can be described in different ways. One defect is a documentation type defect and these can occur in text assets such as instructions or dialogue, audio assets such as sound effects and background music and graphics assets such as items of clothing or weapons.'

D2: learners will comprehensively describe each phase within a test programme explaining why each stage is important. They will describe what the purpose of each stage is. They will describe comprehensively what types of activity happen at each phase and say why they are important, often using highly technical language. They will identify the relationship that each phase has to the others and understand which criteria are used to understand how and when test phases are complete. When describing test processes a distinction grade learner will describe each process in detail and also suggest why certain processes may be used for different types of testing. A learner at this grade might comment 'It is important that, when testing games, companies have a thorough plan of what and when to test so that most of the bugs that may occur in the build are eliminated. Testing a computer game is made up of a testing cycle that occurs within different phases of a test programme. These phases can differ from company to company but they may typically start with a test preparation phase and finish at a gold testing phase. Depending on what is being tested in a game build and why, games testers can use different methods to test games. Black box testing is used to test a game using the normal input devices such as keyboards and joysticks and is useful as it tests the game in the same way a user would play it.'

D3: learners will design near industry-standard games test suites using both combinatorial tables or templates and test flow diagrams. Learners will identify the parameters and values within a game without guidance.

D4: learners will produce a bug report which use clear, concise language and are presented to a near-professional standard. All the elements that would be found in a professional bug report will be present and the bugs reported will be communicated clearly and effectively. Brief descriptions will explain each defect clearly and will be followed by full descriptions that will allow any user that reads the report to recreate the bug without any difficulty. Effective, appropriate techniques will be used to illustrate the descriptions of bugs found.

D1, D2 and D4: technical vocabulary will be secure and used correctly and confidently at all times.

Programme of suggested assignments

The table below shows a programme of suggested assignments that cover the pass, merit and distinction criteria in the assessment and grading grid. This is for guidance and it is recommended that centres either write their own assignments or adapt any Pearson assignments to meet local needs and resources.

Criteria covered	Assignment title	Scenario	Assessment method
P1, M1, D1	Assignment 1 – Game Defect Types: What You Need to Know	Article on game defect types and triggers for an online game fan ezine.	• All preparatory notes. • Ezine article.
P2, M2, D2	Assignment 2 – Game Test Phases and Processes: What You Need to Know	Article on game test phases and processes for an online game fan ezine.	• All preparatory notes. • Ezine article.
P3, M3, D3	Assignment 3 – My Test Suite	Brief from quality assurance leader to devise a comprehensive test suite to test a given game build.	• Report test suite document (word processed or electronic).
P4, M4, D4	Assignment 3 – My Bug Reports	Brief from quality assurance leader to conduct systematic tests following a given test suite on a given game build.	• Illustrated comprehensive bug report document (word processed or electronic).

Links to other BTEC units

This unit forms part of the BTEC Creative Media Production suite. This unit has particular links with the following units in the BTEC Creative Media Production suite:

Level 2	Level 3
	Computer Game Design
	Computer Game Engines
	Human-computer Interfaces for Computer Games
	Sound for Computer Games

Essential resources

Learners will need access to:

- a range of computer games including PC games, console games and multiplayer online games
- hardware such as PCs, consoles, keyboards, joysticks and joypads
- screen grab software
- sample game builds with defects.

Access to professional bug reporting software is desirable.

Employer engagement and vocational contexts

Centres should develop links with local interactive media production studios which could be approached to provide visiting speakers, study visits or samples of typical early game builds.

Delivery of personal, learning and thinking skills

The table below identifies the opportunities for personal, learning and thinking skills (PLTS) that have been included within the pass assessment criteria of this unit.

Skill	When learners are …
Independent enquirers	researching defect types and test phases and processes, judging the value of the information found and using it to support their arguments carrying out research into writing styles and language used, and layout conventions for their bug reports carrying out research to identify game features to be tested in their given game build
Creative thinkers	generating ideas for test cases, asking questions to ensure all necessary features are tested and combining their practice experiences with learned material to devise a comprehensive test suite trying out different combinations of tests to most effectively test their game build
Self-managers	following their test suite to prepare a comprehensive defect report, showing dedication and determination to exhaustively test a game build while organising their time and resources and prioritising their actions seeking out challenges or new responsibilities and showing flexibility when circumstances change while testing and authoring their bug report dealing with competing pressures, including personal and work-related demands while preparing work for assessment.

Although PLTS are identified within this unit as an inherent part of the assessment criteria, there are further opportunities to develop a range of PLTS through various approaches to teaching and learning.

Skill	When learners are …
Reflective learners	setting goals with success criteria for their test cases inviting feedback on their test designs and bug reports and dealing positively with praise, setbacks and criticism evaluating their learning and experience to inform future progress
Team workers	if working in a group to produce a set of linked test suites, taking responsibility for their own role managing their personal contribution to and acting upon information from others in discussions to reach agreements and achieve results when debating the design of the test suites.

Functional Skills – Level 2

Skill	When learners are …
ICT – Use ICT systems	
Select, interact with and use ICT systems independently for a complex task to meet a variety of needs	handling systems to playtest their game build and author their bug report
Use ICT to effectively plan work and evaluate the effectiveness of the ICT system they have used	preparing their fresh computer set-ups while planning the tests on their game build
Manage information storage to enable efficient retrieval	managing bug report forms, test flow diagrams and combinatorial table documents while preparing, testing and reporting on their given game build
Follow and understand the need for safety and security practices	handling systems while playing games to test them
Troubleshoot	handling systems while playing games to test them
ICT – Find and select information	
Select and use a variety of sources of information independently for a complex task	sourcing illustrative artefacts to support their bug reports
Access, search for, select and use ICT-based information and evaluate its fitness for purpose	researching defect types and triggers
ICT – Develop, present and communicate information	
Enter, develop and format information independently to suit its meaning and purpose including: • text and tables • images • numbers • records	building and presenting their test flow diagrams and combinatorial tables, and building and presenting their illustrated bug reports showing the results of conducting tests following their test cases
Bring together information to suit content and purpose	preparing and authoring their bug reports
Present information in ways that are fit for purpose and audience	
Evaluate the selection and use of ICT tools and facilities used to present information	
Select and use ICT to communicate and exchange information safely, responsibly and effectively including storage of messages and contact lists	communicating with QA professionals to research current game test practices

Skill	When learners are …
Mathematics	
Understand routine and non-routine problems in a wide range of familiar and unfamiliar contexts and situations	using estimation and calculation to observe defects in scoring or when checking game physics
Identify the situation or problem and the mathematical methods needed to tackle it	
Select and apply a range of skills to find solutions	
Use appropriate checking procedures and evaluate their effectiveness at each stage	
Interpret and communicate solutions to practical problems in familiar and unfamiliar routine contexts and situations	
Draw conclusions and provide mathematical justifications	
English	
Reading – compare, select, read and understand texts and use them to gather information, ideas, arguments and opinions	studying manufacturers' manuals for sample game builds for testing studying game design documents to identify features to be tested
Writing – write documents, including extended writing pieces, communicating information, ideas and opinions, effectively and persuasively	creating their bug reports and effectively commenting on the performance of the test game.