Games Design Frameworks: A Novel Approach for Games Design Pedagogy

David James and Christopher J. Headleand
Staffordshire University, Games Institute, UK

c.headleand@staffs.ac.uk
D.James@staffs.ac.uk

Abstract: The discipline of Games Design has surged in popularity in higher education as a route to industry for aspiring game developers. Notably, at the time of writing 138 UK Universities are now offering game design degrees, and many further education institutions. This growth is not only in volume but also in the diversity of students who enrol in these courses. Young learners, coming from various backgrounds and with differing levels of professional and academic experience, are drawn to games as a field of study. Games disciplines (such as Games Art, Design, and Programming) are often seen as more accessible than their comparative parent disciplines (such as Computer Science). This diversity presents both an opportunity and a challenge for educators: to develop teaching methodologies that not only cater to a wide range of experiences but also effectively prepare students for the creative and technical demands of the games industry. This paper introduces a pioneering design concept utilized in the pedagogy of game design, namely the use of game frameworks. A framework is a structure built in a video games engine, containing a pre-made, playable game experience, which students are taught to build upon and modify. Frameworks abstract and modularise fundamental, complex elements of the game creation process, allowing new learners to quickly become engaged with the iterative process of designing gameplay. It also allows educators to focus on a specific area of interest. For example, a student can focus just on the UI elements of the game for a whole module of study, without needing to have built the entire game first. This allows students to focus on the elements being taught, without introducing a significant amount of non-assessed content just to enable to learning. This position paper discusses our application of this approach, including the process of framework design, supportive assessment, successes and challenges.

Keywords: Games Design, Pedagogy, Video Game, Games Education, Design Framework

1. Games Engines

Game engines are software tools that serve as a means of game creation (Sobota and Pietriková, 2023). A game engine contains many of the core pre-requisite systems involved in the creation of games, such as graphics rendering, lighting, collision detection, physics, sounds and artificial intelligence. Two of the most popular game engines are Unreal Engine and Unity, which provide the fundamental systems for a multitude of popular games, such as Fortnite and Pokémon.

Historically, the game industry would use in-house solutions to provide these core systems, which would incur a significant amount of time and cost to the developer. In modern game development, game engines are used extensively, and knowledge of game engine usage is a sought-after skill in people entering the industry.

Using game engines to develop games still requires a significant amount of technical, design, scripting and artistic ability, however. Game frameworks build upon these core systems provided by the game engine to create a coherent and structured environment in which students can develop their game creation skills (Mora, Riera and González, et al, 2017)

2. Pedagogic Challenges Teaching Games Engine Development

The study of games development at University is an incredibly popular field, with 138 UK Universities offering game development courses (Hot Courses Abroad, 2024), Staffordshire University's Game Institute has approximately 1800 students studying games development-related disciplines, including games design, games programming and games art.

As interest in these courses grows, Universities are met with the target and challenge of recruiting students from a diverse range of experiences (Universities UK, 2019). Some students will study games due to their passion for the technical creation of game systems, whilst others will study for their passion for games as an entertainment medium and pastime, but with potentially less technical experience learnt at College or School.

Whilst games are inherently technical products– requiring developers to have a core set of technical skills, such as scripting/programming – they are also incredibly creative. Games should immerse a player in an experience, providing challenge, reward and narrative. We have found that a good game developer could be someone with

a strong set of technical skills with an appreciation for designing these experiences, or someone with a strong set of design skills, who has enough technical skill to implement their designs. Therefore, a one-size-fits-all approach to teaching games development is not a workable approach when teaching such a large cohort with varying technical and design skills.

With a large cohort of game development students comes the necessity for a large, multi-disciplinary staff base. At Staffordshire, the Games Institute consists of over 60 games-specific academics. To ensure a supportive and successful student-to-staff ratio, many instances of the same class will run – often in parallel with different class tutors – which creates the challenge of ensuring all students receive a consistent learning experience.

3. Industry Expectations

Games industry development companies are often comprised of large, multi-disciplinary teams, with hundreds of employees (Berg Marklund, Engström and Hellkvist, et al, 2019). Upon completing their degrees, games graduates will be expected to join these teams and seamlessly integrate themselves into the companies' development pipeline.

Significantly, it is the case that game development is a long process, with some games taking many years to come to fruition (Aleem, Capretz and Ahmed, 2016). – from initial design to commercial release. As such, graduates will invariably find themselves joining a company where the game product is very much in-development – the initial design will have been completed, alongside the core systems. Graduates will be expected to quickly familiarise themselves with the product, and begin contributing to the project by developing upon the already pre-made systems and design brief.

Additionally, the games design role is equal parts creativity and technical ability (Gross, Sumner and Thürey, 2010) – a games designer is expected to conceptualise a game idea, and then prove this concept via the creation of playable prototypes. This requires designers to have a broad range of technical skills, including the ability to script/program these prototypes, but also design and implement the environments in which the gameplay takes place.

4. Alternative Approaches

Prior to the implementation of frameworks at Staffordshire University, the requirement for graduates to have technical game development skills whilst also being able to work with a range of in-development games was addressed to a limited extent by the use of commercial game level editors.

Games would sometimes be released with additional software, such as a level editor, that would allow the user to create their own worlds to play in. Epic's Unreal Tournament (2014) was a competitive multiplayer first-person action game, and users could download the Unreal Tournament Editor – in order to create new levels of their own design.

Staffordshire University adopted the use of this level editor in its "Introduction to 3D Games Engines" module, with first year students being tasked with creating a new level of their own design. Learning how to create engaging, aesthetically-pleasing levels for this pre-existing game taught students the importance of working with the limitations of the game. Marks were also awarded for the thematical consistency of their level – did it fit well with the levels already provided with the game?

Whilst this was a valuable experience, it was the case that whilst students learned how to use the Unreal Tournament editor, they did not learn the wider skills outside of this software that were often key requirements of a games designer role – namely, the ability to script and create their own prototypes.

Additionally, since Unreal Tournament contained thousands of industry-standard assets, such as highly-detailed 3D models and sounds, students' individual projects would often have very large file sizes, sometimes over 20 gigabytes per project. This meant that students would have to spend significant time in class saving their project and transferring it external media (or cloud backup) at the end of each class. Furthermore, the submission of the project at the end of the module via the University's digital upload system was a long, protracted process – as was the process for staff of downloading the student work for marking.

5. What is a Framework?

A framework is essentially a game product that provides more functionality than would be found in a base game engine template – but is less developed than a fully complete game (Cowan and Kapralos, 2017).

The initial complex set up of required game systems – such as creating inputs/controls, animations for characters etc – is already completed. However, there will be important areas of the framework that are deliberately missing, and lesson content will involve developing these areas and making the framework fully functional.

With this approach, staff and students can focus more upon the design and implementation of gameplay experiences – such as creating and balancing abilities and enemies.

To visualise: Without frameworks, a significant portion of the module content is spent building the underlying systems of the game, with very little time allocated to creating compelling gameplay experiences with these systems (Fig 1.):

Table 1: Representation of module time spent on development without frameworks.

Developing Core Systems						Designing Gameplay		
Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9

With the implementation of a framework-approached, many of the core systems will already be implemented – requiring students to improve and expand upon these in order to create gameplay experiences within their projects (Fig. 2). This allows the module to focus on the core learning outcomes being delivered, rather than over assessing foundational skills.

Table 2: Representation of module time spent on development with frameworks.

Developing Core Systems			Designing Gameplay					
Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9

Additionally, the framework should provide visual assets that can be used to create a playable environment. By providing these assets, lecturers can ensure that the games have a consistent theme -and also ensure that the assets are well-made and efficient. The Sentinel framework used in the University's "Introduction to Games Design" module included over 50 models that could be used in the creation and decoration of levels, but with the overall project size being only 16 megabytes – 1250 times smaller than the 20 gigabyte projects created in Unreal Tournament. This reduction also has a reducing university storage requirements. Given the UK's regulations, that require assignments to be stored for a number of years, this has a significant impact.

6. Designing Frameworks

Games are inherently multidisciplinary, containing design, programming, modelling and animation content. First year students are provided these frameworks as a structured foundation, with much of the core content already developed, and each week they will learn new techniques that expand upon this core framework content.

6.1 The Three Cs: Character, Control and Camera

Frameworks are constructed using the fundamental games design principle of the "Three Cs – Character, Camera and Control", which is used to develop games of all types (Pluralsight, 2014).

6.2 Character

A Game Development framework should include a core, central character which serves as the Player's avatar within the game. This includes the character model and associated animations that support the game mechanics that students will be expected to build within the framework.

By presenting students immediately with a pre-defined character, this helps to give the framework and the associated module a unique sense of personality. The character can be used throughout the lesson content – essentially as a module mascot – that drives the week-by-week content.

This approach was used in the Staffordshire University module "Introduction to Games Design". The character "Sentinel" is introduced both in the framework and the initial lecture slides. Sentinel is described as crash-landing in its spaceship on an alien planet, and must find a means of escaping. Over the course of the module, the lesson content describes a consistent narrative, with the Sentinel encountering new mechanics and gameplay systems (which the students themselves build). Firstly, students are taught how develop the Sentinel's damage and death systems, before creating combat mechanics in later weeks (see images below). This simple – yet coherent - narrative helps to add purpose and relevance to each week's content.

6.3 Control

Alongside the base animated character in the framework is the interactive control system. Creating core inputs – such as movement and aiming – is an involved process that could potentially be a significant hurdle to a new student.

By providing students with the initial controls, this initial blocker is mitigated and allows the student to immediately begin moving the character around the (largely empty) world space – a space which they will be tasked with designing over the course of the module.

It is also important to establish the control scheme as part of the framework, as it helps to set the style and scope of the game for students: The Sentinel framework, for example, contains simple movement controls on the keyboard, the ability to look around with the mouse, and the ability to attack.

6.4 Camera

The camera is the player's viewport into game world. Different games use the camera in various ways, and a game's genre is often described via it use of camera:

First Person – the player effectively sees through the eyes of the character they control. This an immersive view into the game world. The gameplay experience is designed around the player's limited field of view, with them being required to look around to solve environmental puzzles and/or engage in combat.

Third Person – the camera is positioned behind the Player character. This adds further personality to the player's character, as the player can see their character moving and reacting to the world around them (climbing, opening doors etc). Additionally, due to the positioning of the camera behind the player character, third person games allow for a greater view of the surrounding environment.

Top-down/isometric – the camera is positioned above the world, looking down at the player character. This view gives the player greater information about the surrounding environment, allowing them to make more informed decisions. Due to the distant positioning of the camera compared to first person games, the environment can also have less fine visual detail.

In the Sentinel framework, a top-down camera is used. This encourages students to focus on the encounters and puzzles within the level, without creating the requirement for fine visual detail that would be expected with a first-person view game.

6.5 Assessment Strategy

To encourage student exploration and development of the frameworks, the module should incorporate a supportive assessment strategy. In first year modules, this should include elements that support and reward less-technical learners, whilst also offering more experienced students the ability to grow and challenge themselves.

This concept of supportive assessment is based upon ensuring students are motivated by the creation of bitesize, attainable goals that reward innovation and exploration of the framework (Blaine Lawlor and Hornyak, 2012).

In "Introduction to Games Design", the initial weeks of the module are spent introducing core concepts which are used to develop missing parts of the framework – such as a health system for the player character, enemies and hazards to avoid etc. As each of these topics are introduced, so is a small set of bite-size tasks.

These tasks ask the student to evidence that they have completed the lesson content – for example, they will be required to evidence via a short video that they have a functioning health system working with the player character. Each of these bitesize, weekly tasks carry a small mark percentage, and students are introduced to the first of these tasks from the very first class.

The purpose of these tasks is to build confidence early in first year students. Students who are less technical are able to achieve initial marks by building the functionality along with the lecturers. By tasking students with building small, missing systems within the framework – as opposed to having to build the fundamental systems that drive the whole project – the scale of the work is made more immediate and achievable.

Once the main framework systems have been developed via bitesize tasks, a more open brief can be provided that encourages research and further development. In "Introduction to Games Design", students are introduced to assessment 2 – the "project". For this assessment, students are tasked with creating a full level for the framework, which should demonstrate all of the techniques shown in the initial weeks of the module, but with clear evidence of these developed beyond the content of the weekly tasks. This permits more capable students the opportunity to demonstrate their skills and achieve high marks, whilst also ensuring that less-technical students have the required knowledge to succeed at the assessment.

7. Results

7.1 Progression Improvement

Another challenge with Games Development subjects is their inherently ab initio subjects, with most students not having studied them in the past. This often leads to challenging first year modules, especially when the module is taught in the traditional method (no framework) requiring the students to effectively master several games development skills just to build a platform to explore the core learning outcomes.

We found that the framework approach has a significant impact on improving these progression rates. For example, we have the first-year module "Introduction to Games Design" (Fig. 3):

Table 3: Student progression rates from 2019 to 2023

Academic Year	2019/20 (Pre-Frameworks)	2020/21 (Pre-Frameworks)	2021/22 (Frameworks introduced)	22/23	23/24
Progression (Pass) Rate	62%	46%	76%	83%	85%

Notably 19-20 also coincided with the global pandemic, which significantly disrupted teaching. However, this is also as far back as we have data for this specific module. Even accounting for covid disruptions, we note a significant improvement in the overall pass rates of the module.

8. Student Satisfaction

In module feedback surveys, students have expressed an overwhelmingly positive reaction to the module following the implementation of frameworks. In feedback, students have expressed that use of frameworks has been an excellent introduction to the game engine software, with them going from little experience to creating fully playable levels and gameplay of their own design within a single teaching block.

9. Conclusion

The introduction of game design frameworks has demonstrably improved student engagement, pass rates, and general satisfaction in games development modules. This approach integrates supportive assessment strategies, and allows students to focus on the core learning requirement of a module, avoiding over-assessment.

By providing a structured and flexible games foundation, frameworks help to bridge the gap between varying levels of student experience. Less technical individuals can focus on learning and developing specific game components without the overwhelming complexity of building entire systems from scratch. This method further facilitates the development of both creative and technical skills, ensuring that students are well-prepared for the demands of the games industry.

The framework approach aligns with industry practices where game development often involves expanding upon pre-existing systems rather than creating everything from the ground up.

Additionally, the use of frameworks addresses several logistical challenges faced in traditional game design education. By incorporating bite-sized, attainable goals within the assessment strategy, students are encouraged to explore and innovate at their own pace. This approach supports less experienced learners by building their confidence and competence gradually while offering advanced students opportunities to push the boundaries of their skills.

The adoption of game design frameworks represents a significant advancement in the pedagogy of game development education. It not only aligns with industry expectations but also caters to the diverse needs of the student population. The positive impact on progression rates and student feedback underscores the effectiveness of this approach. As the field of game design continues to evolve, the integration of such frameworks will likely become an essential component of educational strategies aimed at preparing the next generation of game developers.

References

Aleem, S., Capretz, L.F. & Ahmed, F. (2016). 'Game development software engineering process life cycle: a systematic review'. J Softw Eng Res Dev 4, 6 (2016). https://doi.org/10.1186/s40411-016-0032-7

Berg Marklund, B., Engström, H., Hellkvist, M. et al. (2019). 'What Empirically Based Research Tells Us About Game Development'. Comput Game J 8, pp 179–198. https://doi.org/10.1007/s40869-019-00085-1

Blaine Lawlor, K., Hornyak, M. (2012). 'Smart Goals: How the Application of Smart Goals can Contribute to Achievement of Student Learning Outcomes', *Developments in Business Simulation and Experiential Learning,* 39 (2012), pp. 259-267. Smart Goals: How the Application of Smart Goals can Contribute to Achievement of Student Learning Outcomes | Developments in Business Simulation and Experiential Learning: Proceedings of the Annual ABSEL conference (tdl.org)

Cowan, B., Kapralos, B. (2017). 'An Overview of Serious Game Engines and Frameworks'. In: Brooks, A., Brahnam, S., Kapralos, B., Jain, L. (eds) Recent Advances in Technologies for Inclusive Well-Being. Intelligent Systems Reference Library, vol 119 . Springer, Cham. https://doi.org/10.1007/978-3-319-49879-9_2

Gross, M., Sumner, R.W., Thürey, N. (2010). 'The Design and Development of Computer Games'. In: Konsorski-Lang, S., Hampe, M. (eds) The Design of Material, Organism, and Minds. X.media.publishing. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-69002-3_4

Hot Courses Abroad (2024). 'Universities in the UK Offering Computer Games Design Degrees and Courses'. Available at: 138 Institutions offering Computer Games Design Courses In the UK (hotcoursesabroad.com) (Accessed: 5th May 2024).

Mora, A., Riera, D., González, C. et al. (2017). 'Gamification: a systematic review of design frameworks'. J Comput High Educ 29, 516–548. Available at: https://doi.org/10.1007/s12528-017-9150-4

Pluralsight (2014). 'Character, Controls, Camera: The 3Cs of Game Development'. Available at: Character, Controls, Camera: The 3Cs of Game Development | Pluralsight (Accessed: 1st May 2024).

Sobota, B., Pietriková, E. (2023). 'The Role of Game Engines in Game Development and Teaching'. *Computer Science for Game Development and Game Development for Computer Science*. IntechOpen. Available at: http://dx.doi.org/10.5772/intechopen.1002257.

Universities UK (2019). 'Widening Opportunity in Higher Education'. Available at: widening-opportunity-third-phase.pdf (universitiesuk.ac.uk) (Accessed: 3rd May 2024).