EG UK Computer Graphics & Visual Computing (2025)
Y. Sheng and A. Slingsby (Editors)

Microknight: A Shared Framework for Game Analysis in Design Education

David James[1], Kieran Hicks[1] and Chris Headleand[1]

[1]Staffordshire Games Institute
School of Digital, Technologies, Innovation and Business
University of Staffordshire, UK

Abstract

This paper presents MicroKnight, ab action game created for use in undergraduate Games Design curricula. MicroKnight was designed to address accessibility and consistency issues when teaching analysis. The game ia a complete commerical-quality experience, allowing all students to engage in a shared analytical experience. A modified 'broken' version is also provided, containing deliberate balance and usability flaws to support critique teaching exercises. This paper outlines the rationale, implementation, and pedagogic applications of the framework.

1. Background and Motivation

This technical report introduces MicroKnight, a framework [JH24] game developed to support the development of analysis skills in games design students. Analysing commercial games creates barriers. Access is limited by cost, platform compatibility, and hardware requirements, which can prevent students from engaging equally with the material.

Analysis is an important part of any higher education and is recognized as a key pillar of effective learning [Kra02]. This is especially true of creative disciplines, where analysis is recognised a key milestone on the development of subject mastery [Har23], and is embedded as a key-skill at various educational levels [Gre09]. However, this presents a challenge in games education. In other creative disciplines such as (Creative Writing), a student can be tasked with reading a text from the curriculum (which can be accessed from a library or often freely available online), games present the challenge of having significant costs to entry, both the purchase of a game (which is typically non-transferable), and the hardware to run it. As a result, students often analyse the different titles available to them, making collective comparison difficult, MicroKnight addresses this challenge.

2. MicroKnight Framework

MicroKnight is a top-down melee-focused action game developed in Unreal Engine 5. The player controls a knight navigating a procedurally generated environment. Each level increases in difficulty, introducing new enemies and challenges.

The game focuses on stamina management and melee combat.

Figure 1: *MicroKnight Gameplay: Including the player character, procedural environment, and user interface (health and stamina)*

All gameplay elements are built to interact with this core mechanic. For example, enemies are designed to affect stamina usage differently. Some drain stamina with repeated light attacks, while others pressure the player with heavy strikes. Players unlock abilities that alter how stamina is consumed or recovered, creating tactical variation. An example of the gameplay can be seen in figure 1.

As in many modern games, balance emerges from interactions between systems. Enemies that are trivial in isolation become difficult in groups or in combination with others. These layered interactions offer opportunities for students to analyse encounter design, pacing, and challenge escalation.

MicroKnight also includes visual feedback systems, UI elements, upgrade trees, and scoring mechanics. These features re-

inforce communication of goals and progress, and support analysis of feedback loops in gameplay.

2.1. Procedural Levels

To generate a level, a Level Manager spawns multiple FloorSpawners, which move in a grid-based manner and creating FloorTile objects wherever they move to. By having an actor such as the FloorSpawner physically move around and "laying down" FloorTiles, it guarantees that the whole floor space of the level is traversable and reachable by the Player. Once the Floor Spawners have spawned a target number of FloorTiles, the LevelManager asks each FloorTile to spawn Walls on any side that is not connected to another FloorTile. This results in the entire floor plan of the level being surrounded by Walls.

The LevelManager then reads through its current EnemyList, which contains the number and type of Enemy actors to spawn. The LevelManager next locates a FloorTile to serve as a Player Start Location. The LevelManager searches through each FloorTile until it find a FloorTile that is a minimum distance away from the nearest enemy.Once a suitable Player Start Location has been found, the LevelManager does a final pass of the level, randomly adding dressing to the FloorTiles and Walls, such as decals, rocks, torches etc.

3. Analytical Sandbox

To support deeper critical engagement, students are provided with a second version of MicroKnight that contains deliberate design flaws. These include:

1. Inconsistent or missing UI feedback
2. Poorly balanced stamina costs
3. Overpowered (unbalanced) enemies
4. Vague or contradictory goal communication
5. Ambiguous collision and attack telegraphing

This broken version allows students to experience poorly implemented systems in a controlled environment, compared to the complete version; allowing for concept abstraction. This approach supports tasks such as usability analysis, bug reporting, and improvement planning. Students are asked to identify issues, justify their findings, and propose design improvements using diagrams and standard terminology.

One example of a games mechanic which is adapted to a non-optimal position is the 'perks system'. Perks are additional abilities that the player unlocks as they move through the game. A good example (seen in figure 2 is the energy wave attack. This perk gives the player a chance to fire an energy projectile. Since this is a range attack,t risks potentially breaking the dydynamics of melee combat..

4. Pedagogic Application

MicroKnight is used to introduce key concepts in design communication and gameplay analysis. Students begin by playing the game without guidance. They are then asked a series of reflective questions, such as:

Figure 2: *The MicroKnight 'Energy Wave'.*

- What is the goal of the game and how is it conveyed?
- What are the primary player actions?
- How are game systems like stamina and abilities introduced?

These questions form the basis of individual and group discussion. Students then examine specific systems in detail, such as enemy types, ability design, or level pacing. They are required to present their findings through visual diagrams, annotated maps, and written analysis.

By using a common game that the teaching team is fully familiar with, assessments can focus on depth of understanding and clarity of communication, rather than on whether the student has described an unfamiliar system correctly.

5. Conclusion

MicroKnight provides a structured and accessible solution to a common issue in games education: the lack of a shared analytical reference. By offering a custom-designed, fully playable game aligned to key teaching objectives, it allows students to engage with gameplay systems in a consistent and meaningful way. The inclusion of both complete and deliberately flawed versions supports practical analysis, encourages reflective practice, and reinforces design theory through experimentation.

This framework has enabled educators to assess student understanding more reliably, improved classroom discussion through shared experience, and enabled focused critique of gameplay elements such as balance, feedback, and encounter design. Future work will explore extending the system to include new gameplay styles, user testing features, and multiplayer components to support collaborative learning.

References

[Gre09] GREEN A.: Creative writing in a level english literature. *New Writing 6*, 3 (2009), 187–195. 1

[Har23] HARPER G.: *Creative writing analysis*. Routledge, 2023. 1

[JH24] JAMES D., HEADLEAND C. J.: Games design frameworks: A novel approach for games design pedagogy. In *Proceedings of the 18th European Conference on Games Based Learning* (2024), Academic Conferences and publishing limited. 1

[Kra02] KRATHWOHL D. R.: A revision of bloom's taxonomy: An overview. *Theory into practice 41*, 4 (2002), 212–218. 1