

SPELLEXIS



A serious game for dyslexic learners aged 10+

20-minute project presentation

Dyslexia support

2D platformer

Unreal Engine 5

The problem: dyslexia and engagement

Why the project needed both educational purpose and game motivation

- Dyslexia affects reading, spelling and word decoding skills.
- Older learners often need support that feels mature and motivating.
- Many tools are useful, but can feel repetitive or too young for 10+ learners.
- The challenge: practise literacy without making the experience feel like a worksheet.

Educational need

Motivation

Accessible practice



Research gap and project aim

Positioning Spellexis within serious games for literacy support



- Many serious games focus on younger children or short mini-games.
- The dissertation focuses on older learners, multilingual support and stronger game identity.
- The aim is to design a serious game that connects literacy practice with exploration, combat and progression.

What is Spellexis?

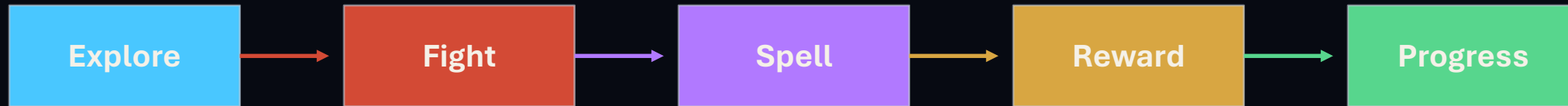
A dark fantasy 2D platformer with embedded learning mechanics



- Explore fantasy levels such as Eldrath Dungeons and Veilwood Forest.
- Fight enemies, collect gems and use checkpoints to preserve progress.
- Activate word-based mini-games that combine images, audio and letter selection.
- Progress toward boss fights where spelling becomes part of combat.

Core gameplay loop

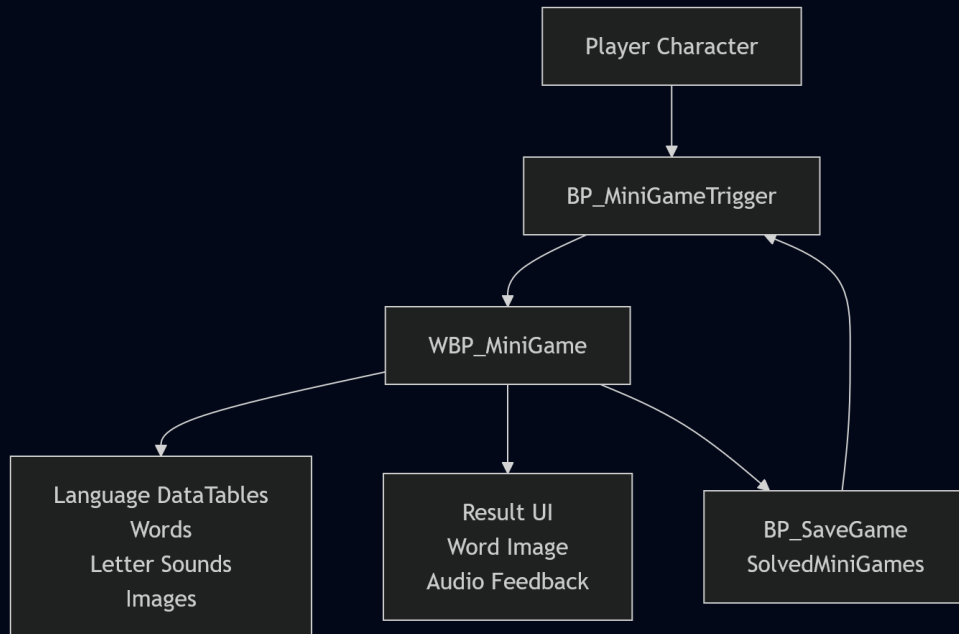
Learning is embedded into the player's normal progression



- Exploration reveals enemies, pickups, puzzles and mini-game triggers.
- Spelling tasks provide rewards and unlock further progression.
- Boss fights escalate the same learning mechanic under pressure.

Spelling mini-game system

Multisensory word practice: image + audio + letters + feedback



- The player activates a BP_MiniGameTrigger during exploration.
- WBP_MiniGame loads localized word data from DataTables.
- The player uses shuffled letters to construct the target word.
- Correct spelling updates progress, rewards the player and saves completion.

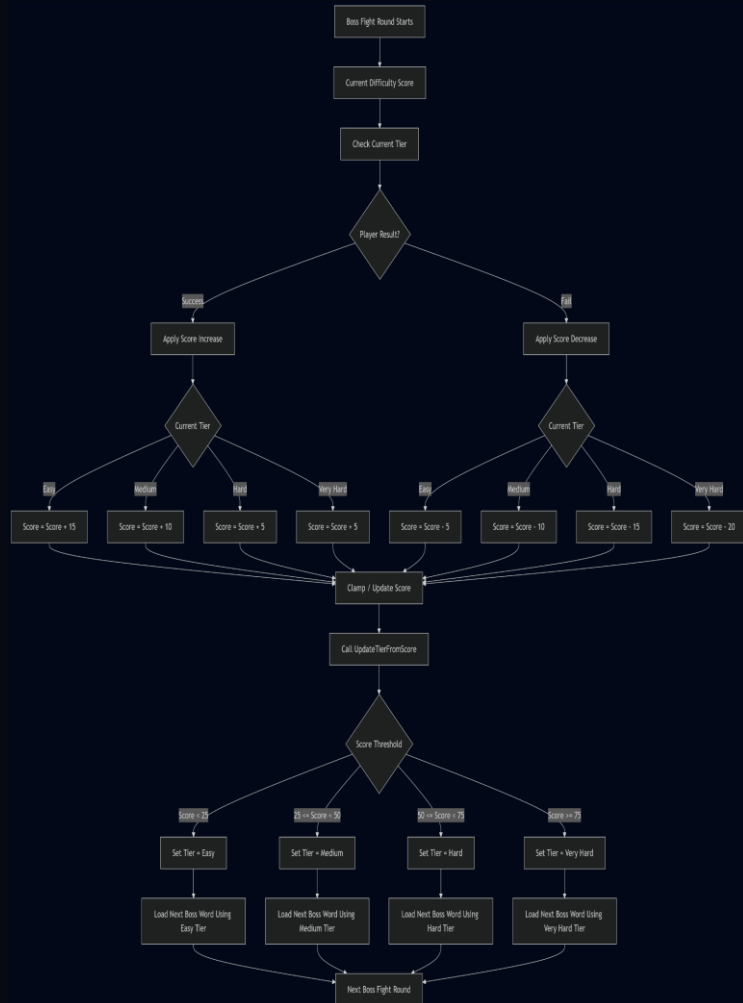
Visual cue

Letter sounds

Feedback

Boss fight and dynamic difficulty

Spelling mechanics become part of combat escalation



- Boss encounters act as narrative and mechanical gates.
- Correct spelling damages the boss. Failure increases pressure.
- Dynamic difficulty changes score and tier after each round.
- High tiers rise slowly and punish mistakes more strongly to avoid trapping the learner.

Easy

Medium

Hard

Very Hard

Characters, combat and enemies

Three playable classes supported by reusable enemy systems



Mage

- high mana growth
- spell-focused play
- ranged identity



Fighter

- high health growth
- strong melee attacks
- front-line survivability



Rogue

- middle health growth
- light melee attacks
- front-line

Enemy AI

- BP_BaseEnemy parent class
- Blackboard-driven targeting
- melee and ranged child enemies
- XPReward on death

Progression, save/load and HUD

Player progress persists across checkpoints, continue screens and map transitions



- Enemy death grants XP once using XPGranted protection.
- Level-up increases class-based stats and refills health, mana and stamina.
- XP, level, thresholds and upgraded attack damage are stored in S_PlayerSave.
- HUD updates through PlayerCharacter → PlayerController → HUD → Overlay.

Save/Load

DeathMap continue

Dungeon → Forest

Technical implementation

A modular Blueprint architecture in Unreal Engine 5



- Blueprint interfaces connect systems without hard-coding every dependency.
- DataTables support multilingual word, sound and image content.
- Event dispatchers update the HUD without constant polling.
- SaveGame stores player state, solved mini-games, defeated enemies, items and checkpoints.

Development challenges and solutions

The most important lessons from implementation

Challenge 1

Integrating learning without stopping the game flow

- mini-games appear from world triggers
 - boss fights reuse spelling under pressure

Challenge 2

Keeping progression consistent after loading and transitions

- `S_PlayerSave` stores XP, level and upgraded damage
- tested via `StartMap` and `DeathMap` continue

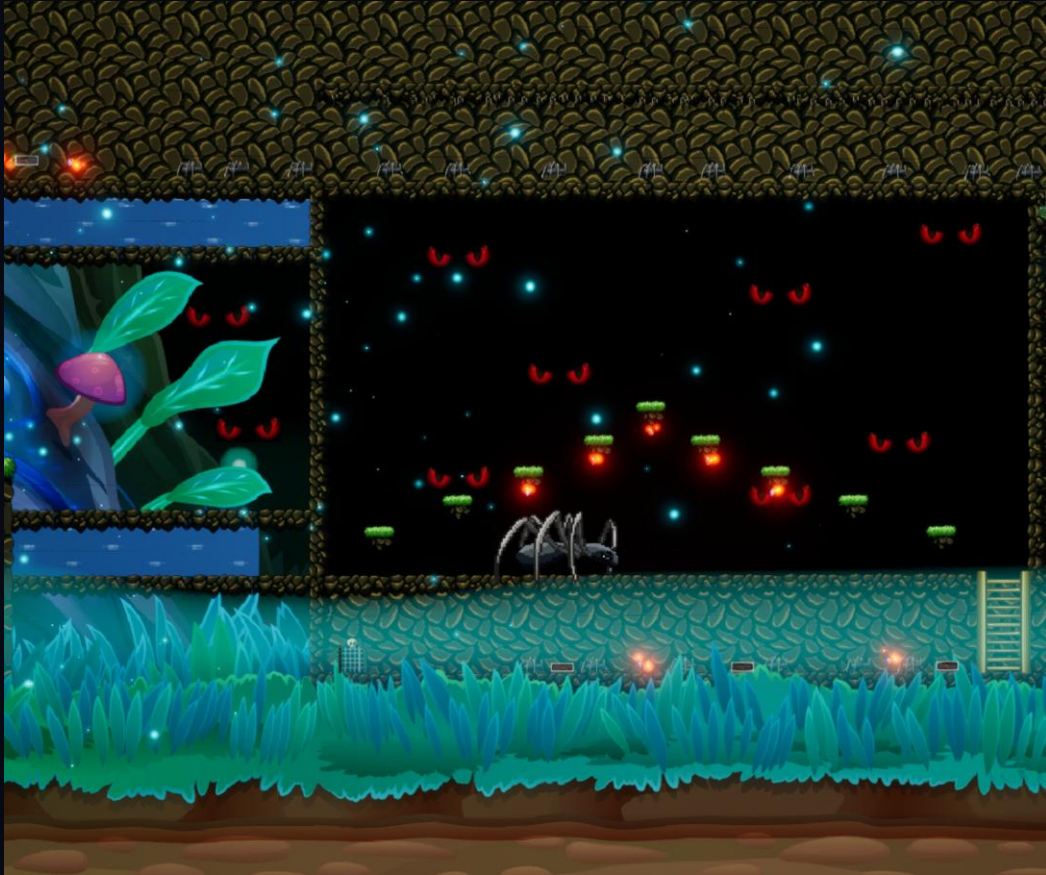
Challenge 3

Making systems reusable and maintainable

- parent enemy class handles XP reward
- dispatcher pipeline updates UI cleanly

Results and future work

What has been achieved and where Spellexis can go next



Achieved

- playable serious game prototype
- spelling mini-games + boss encounters
- XP, save/load, localization and HUD systems

Future work

- testing with dyslexic learners
- more levels, words and languages
- stronger adaptive learning and accessibility refinement

Thank you

Questions?