**Untitled Knight Game**

[**https://github.com/dertkoD/Final\_Project\_MonoGame\_Daniel\_Maxim**](https://github.com/dertkoD/Final_Project_MonoGame_Daniel_Maxim)

**The Goal of the Game**

Survive as long as you can while enemy arrows and bombs fly in from the screen edges. Deflect threats and keep your HP above zero. When you die, the end screen lets you restart or return to the main menu.

**Core Mechanics**

- Two combat states: Attack (sword) and Defend (shield).

- Attack cooldown: prevents spamming sword swings; timing matters.

- I-frames after damage: short invulnerability window on hit.

- Deflect rules

- Arrows: can be deflected by sword or shield. A deflected arrow bounces back, starts spinning, and no longer hurts on body contact.

- Bombs: Shield hit → bomb explodes immediately; you take 1 damage as a penalty. Sword hit → bomb is knocked away (reflected).

- Deflect streak heal: every N successful deflects (default 5) you heal **+1 HP**.

- Spawning cadence: enemies spawn from configurable edge points at semi-random intervals.

- Seamless transitions: main menu slides away into gameplay; same parallax/ground continue.

- Game over: waits for the death animation, then slides in the end screen.

**Controls**

- Move / Face: 'A' / 'D' (or '←' / '→')

- Attack (sword): 'E' (press)

- Defend (shield): 'Left Shift' (hold)

- Pause / Unpause: 'P'

- Main Menu: 'Esc' while paused in gameplay

- Toggle Music / SFX: 'M' / 'N'

**Gameplay**

1. Main Menu: parallax forest + ground are visible. Press **Start**.

2. Survive: arrows and bombs spawn from edges. Use E to swing (reflect/knock) or Left Shift to shield (block arrows, detonate bombs with −1 HP).

4. Damage & I-frames: hits reduce HP then grant a brief invulnerability window.

5. Skill heal: every 5 deflects (configurable) restores **+1 HP**.

6. Game Over: at 0 HP, show end screen with **Restart** and **Main Menu**.

**How the Project Is Organized**

**- Core Loop & Bootstrapping**

- 'Game1': sets resolution, loads content, builds scenes, runs updates/draws, handles global input and transitions.

- 'SceneManager': tiny scene graph (lists of updateables/drawables); 'Add', 'Remove', 'SwitchTo'.

- 'AudioManager': central music/SFX registry and playback (volumes and toggles).

- 'Collider': debug-drawable AABB with trigger/collision events.

**- Rendering & Animation**

- 'Sprite', 'Animation', 'SpriteSheet', 'SpriteSheetInfo': base rendering and frame-based animation.

- 'SpriteManager': loads and stores sprite sheets/metadata for lookup by name.

**- Player (modular by responsibility)**

- 'Player': orchestrator (state machine, wiring).

- 'PlayerController': reads keyboard → intents (AttackPressed / DefendHeld) and facing.

- 'PlayerAnimationController': plays named clips and short FX (shield flash).

- 'PlayerHealth': HP, invulnerability timer, damage/heal events, death.

- 'PlayerCombat': attack cooldown, sword/shield interactions, deflect logic, heal-on-streak.

- 'PlayerColliders': keeps body/sword/shield hitboxes in sync.

**- Enemies**

- 'Enemy' (base), 'Arrow', 'Bomb': movement, gravity, collisions; arrow spin after deflect; bomb explosion.

- 'EnemySpawner': semi-random spawns from edges toward the player.

- 'ExplosionFx': one-shot explosion animation that auto-removes.

**- Menus, Transitions & Background**

- 'MainMenu', 'EndGameScreen': UIs that can be slid in/out.

- 'ParallaxBackground', 'GroundLayer': forest layers and ground rendering (smooth offsets).

- 'MenuTransition', 'Config', 'Phase', 'SlideDirection': orchestrate seamless slides + call 'onComplete()'.

**- UI Framework & Menus**

- 'UIElement' — base for UI widgets ('Enabled', 'Visible', 'Position') with 'OnUpdate/OnDraw' hooks.

- 'TimerUI' — stopwatch HUD ('mm:ss'), optional center alignment.

- 'Menu' — base menu: window panel, buttons, selector, title, and 'SlideOffsetY' for transitions.

- 'MainMenu' — concrete menu with **Start** / **Exit** and a **HelpIcon** + 'HelpPopup' for tips.

- 'HelpIcon' — small “?” hover area that toggles **HelpPopup** visibility.

- 'HelpPopup' — word-wrapped tooltip panel listing controls and hints.

- 'EndGameScreen' — death overlay with **Restart** / **Main Menu**.

- `HelpIcon` — small “?” hover area that toggles `HelpPopup` visibility.

- `HelpPopup` — word-wrapped tooltip panel listing controls and hints.

- `Button` — clickable UI control with optional spritesheet background or solid fill, hover tint, centered label, and click event.

- **Utilities**

- 'OneShotTimer' — fires a callback once after N seconds (e.g., delay before showing Game Over).

**Core Classes Responsibilities**

- Game1 — boot, load assets, build scenes, global input, transitions, music swap.

- SceneManager — add/remove/update/draw objects; 'SwitchTo' wipes/builds a scene.

- SpriteManager — map string names → textures + grid meta.

- AudioManager — songs/SFX registry + playback + volume/mute toggles.

- Sprite / Animation — drawing and frame stepping for sprite sheets.

- Collider — axis-aligned rectangle trigger/collision events (debug outline in 'DEBUG').

- Player — orchestrates controller/health/combat/animation/colliders and state machine.

- PlayerController — keyboard → intents and facing.

- PlayerAnimationController — named clips + shield flash timing.

- PlayerHealth — HP, i-frames, death event.

- PlayerCombat — cooldowns, deflect rules, heal-on-streak.

- PlayerColliders — computes body/sword/shield rectangles each frame.

- Enemy — base movement/gravity/collider sync; offscreen lifetime rules.

- Arrow — projectile; spins & falls after deflecting, ignores body damage while spinning.

- Bomb — throwable; sword knocks away, shield detonates (−1 HP), explosion spawns FX.

- EnemySpawner — timed spawns from edges toward the player.

- ExplosionFx — self-removing one-shot animation.

- ParallaxBackground — layered scrolling background with speed smoothing.

- GroundLayer — tiled ground (cap + fill) with global Y offset for transitions.

- MenuTransition / Config — menu/overlay slides + parallax/ground easing; calls 'onComplete()'.

- UIElement — base widget contract (update/draw gates).

- TimerUI — `mm:ss` stopwatch HUD.

- OneShotTimer — fire a callback once after N seconds.

- HpUI — hearts HUD (reads `Player.HP`/`Player.MaxHP`).

- Button — clickable UI element (spritesheet or solid fill), hover tint, centered text, click event.

- Text — cached-measure label (+ optional shadow).

- HelpIcon — hover trigger that shows `HelpPopup`.

- HelpPopup — word-wrapped tooltip with padding