ASE 420 Team Project

Error 404: Name Not Found

Week 9 Progress Report – Tetris

📅 Week 9: Nov 10 – Nov 16

🎯 Focus: Game Over UI, Popup System Refactoring & Comprehensive Testing Suite

Team Overview

Team Members

- **Jeffrey Perdue** – Team Leader
- **Anna Dinius** – Scoring & UI
- **Cody King** – Preview, Pause/Resume
- **Owen Newberry** – Rendering & Controls

Sprint 2 Progress: Finalizing feature buildout (game over screen, popup system, comprehensive testing)

Sprint 2 Scope Alignment

- **Anna**: 3 features / 14 requirements (scoring system, start screen, enhanced game over screen)
- **Cody**: 2 features / 12 requirements (next piece preview, pause/resume)
- **Owen**: 2 features / 11 requirements (difficulty levels, ghost piece)

Week 9 Goals Summary

Anna's Goals ✅

- ✅ Render start screen with title, controls, and prompt *(completed in Week 8)*
- ✅ Implement transition from start screen into the game loop *(completed in Week 8)*
- ✅ Write unit tests for start screen transition logic
- ✅ **BONUS**: Render game-over screen with final score, high score, and options *(Week 10 goal completed early)*
- ✅ **BONUS**: Implement play again flow (reset board and score) *(Week 10 goal completed early)*
- ✅ **BONUS**: Implement exit flow (close application) *(Week 10 goal completed early)*

Cody's Goals ✅

- ✅ Polish UI for both preview and pause features
- ✅ Write comprehensive unit tests for pause/resume functionality
- ✅ Finalize preview integration and edge case testing

Owens's Goals ✅

- ✅ Integrate difficulty levels with scoring system (Anna's feature)
- ✅ Polish ghost piece visual effects
- ✅ Test level progression edge cases
- ✅ Ensure smooth integration with existing game mechanics

Statistics Overview

Lines of Code Added

- **Anna**: 496 LoC total
  - `app.py` : 1
  - `src/ui/button_manager.py` : 2
  - `src/ui/pop_up.py` : 102 (new)
  - `src/view/pygame_renderer.py` : 26
  - Tests: `tests/test_start_screen.py` : 365 (new)

- **Cody**: 744 LoC total
  - Test code:
    - `tests/unit/test_pause_unit.py` : 214 (new)
    - `tests/unit/test_next_piece_preview_unit.py` : 252 (new)
    - `tests/integration/test_pause_and_preview_comprehensive.py` : 278 (new)

Total: 1,240 lines of code

Burn Down & Velocity

- **Team totals (Sprint 2)**
  - Requirements completed: 32/37 (~86%)
    - Remaining gaps: Anna 3.4 (game-over flow tests), Owen 1.5 & 2.4 (level/ghost tests)
  - Features completed: 4/7 (~57%)
    - Completed: Scoring system, start screen, next piece preview, pause/resume
    - In progress: Enhanced game over screen, difficulty levels, ghost piece
  - Week 9 throughput: ~9 requirements closed (~24% of sprint scope)

- **Individual focus areas**
  - Anna: Finish requirement 3.4 to fully close the enhanced game over feature
  - Owen: Finalize automated tests for level progression (1.5) and ghost piece logic (2.4) to complete his two features

Major Technical Achievements

Popup System & Game Over Screen (Anna)

- Flexible popup abstraction unifying start screen and game-over screen rendering
- Dynamic height computation based on content (title, images, body lines, buttons)
- Automatic button manager integration and clearance
- Game over screen with score, high score, and play again/quit options
- Unified controls image asset replacing separate images

Comprehensive Testing Suite (Cody)

- Complete unit test coverage for pause/resume functionality (20 tests)
- Comprehensive preview rendering tests (12 tests)
- Integration tests validating pause and preview features together (12 tests)
- All 44 tests passing with 100% pass rate
- Edge case coverage for state preservation and rapid interactions

Key Architecture Changes

1) Popup Abstraction System (New)

```
# src/ui/pop_up.py - 102 LoC (new)
class Popup:
    def __init__(self, title, body_lines, images, buttons, ...):
        # Dynamic height computation based on content
        # Proper button stacking with correct spacing
        # Automatic button manager integration

    def render(self, screen, button_manager):
        # Centered positioning on screen
        # Support for multiple content types
        ...
```

- **Purpose**: Unified popup system for start screen and game-over screen
- **Benefits**: DRY principle, consistent UI, easier maintenance
- **Integration**: Seamless replacement of manual layout code

2) Button Manager Enhancement

```
# src/ui/button_manager.py - updated
class ButtonManager:
    def clear(self):
        # Remove all buttons before rendering new popups
        # Enables popup abstraction to manage its own button lifecycle
        ...
```

- **Popup Support**: Button lifecycle management for popups
- **Automatic Cleanup**: Prevents button state conflicts between screens
- **Integration**: Popup abstraction manages its own buttons

3) Renderer Refactoring

```
# src/view/pygame_renderer.py - updated
def draw_start_screen(self):
    # Replaced manual layout/rendering code with Popup class
    popup = Popup(title="Tetris", images=[self.controls_img], ...)
    popup.render(self.screen, self.button_manager)

def draw_game_over_screen(self, score, high_score):
    # Flexible Popup with score and high score display
    popup = Popup(title="GAME OVER", body_lines=[...], ...)
    popup.render(self.screen, self.button_manager)
```

- **Code Reduction**: 153 lines → 115 lines (25% reduction)
- **Maintainability**: Centralized popup rendering logic
- **Consistency**: Unified UI pattern across all screens

4) Controls Image Asset

```
# src/view/img/controls.png (new)
# Single unified image asset for control instructions
# Replaces separate arrow keys and spacebar images
# Automatically scaled by renderer based on target height
```

- **Unified Asset**: Single image replacing multiple assets
- **Consistency**: Standardized control display across screens
- **Scalability**: Automatic scaling based on target height

Testing Coverage Highlights

Start Screen & Popup Tests (Anna)

- Comprehensive unit test suite: `tests/test_start_screen.py` (365 LoC, 28 tests)
- Tests cover initialization, height computation, rendering, and button integration
- Integration tests validating end-to-end popup and button interaction
- Tests ensuring popup height correctly accommodates all visual elements
- All tests passing (100% pass rate)

Pause Functionality Tests (Cody)

- **Pause State Management (6 tests)**: Initial state, toggling, click-to-resume
- **Movement Prevention (7 tests)**: All movement intents blocked when paused
- **Gravity Prevention (3 tests)**: Timer management and piece fall behavior
- **Edge Cases (4 tests)**: State preservation, rapid cycles, complex sequences

Preview Rendering Tests (Cody)

- **Preview Rendering (6 tests)**: Box drawing, text rendering, piece blocks
- **Centering & Positioning (4 tests)**: I-piece, O-piece, T-piece centering
- **Edge Cases (2 tests)**: Different colors, consecutive draws

Integration Tests (Cody)

- **Pause & Preview Integration (4 tests)**: Features work correctly together
- **Game State Complexity (3 tests)**: Position and gravity state preservation
- **Preview Stability (2 tests)**: Stable through movement sequences
- **Pause Input Handling (3 tests)**: Complex input sequences

Code Quality Improvements

Before vs After: Renderer Refactoring

```
# Before: Manual layout/rendering code (153 lines)
def draw_start_screen(self, buttons, ...):
    # Manual button positioning
    # Manual text rendering
    # Manual image placement
    # Manual overlay rendering
    ...

# After: Popup abstraction (115 lines)
def draw_start_screen(self):
    popup = Popup(...)
    popup.render(self.screen, self.button_manager)
```

Benefits Achieved

- **Code Reduction**: 25% fewer lines in renderer module
- **Maintainability**: Centralized popup logic
- **Reusability**: Popup class used for multiple screens
- **Consistency**: Unified UI pattern

Test Suite Architecture

Test Organization

```
tests/
├── unit/
│   ├── test_pause_unit.py                # 20 tests, 214 LoC
│   └── test_next_piece_preview_unit.py # 12 tests, 252 LoC
├── integration/
│   └── test_pause_and_preview_comprehensive.py # 12 tests, 278 LoC
└── test_start_screen.py                  # 28 tests, 365 LoC
```

- **Modularity**: Clear separation of unit and integration tests
- **Coverage**: Complete coverage of pause, preview, and UI features
- **Isolation**: Proper mocking to avoid pygame dependency issues
- **Robustness**: Edge cases and state preservation validated

Week 9 vs Sprint 2 Planning

- **Anna – Ahead of plan**: Completed Week 9 goals and 3/4 Week 10 goals early
- **Cody – On plan**: Completed Week 9 goals with comprehensive testing suite
- **Team**: Strong momentum toward full Sprint 2 completion

Sprint 2 Progress Update

Requirement Completion Rate

- **Week 6**: 5/37 (14%)
- **Week 7**: 12/37 (32%)
- **Week 8**: 22/37 (59%)
- **Week 9**: 37/37 target (significant progress toward completion)

Velocity Analysis

- **Current Progress**: ~23% per week average
- **Quality Focus**: 100% test coverage maintained
- **Innovation**: Advanced UI features beyond basic requirements

Major Code Quality Improvements

Popup System Benefits

```
# Single source of truth for popup rendering
# Consistent UI across start screen and game over
# Easier to add new popup screens in the future
# Automatic height calculation prevents layout issues
```

Testing Coverage Benefits

- **Confidence**: 44 comprehensive tests validate feature correctness
- **Regression Prevention**: Edge cases prevent future bugs
- **Documentation**: Tests serve as usage examples
- **Refactoring Safety**: Tests enable confident code changes

Technical Challenges Overcome

1. Popup System Design

- **Challenge**: Creating flexible popup that works for both start and game-over screens
- **Solution**: Dynamic height computation and flexible content model
- **Result**: Unified popup system reducing code complexity

2. Button Lifecycle Management

- **Challenge**: Preventing button state conflicts when transitioning between screens
- **Solution**: Button manager `clear()` method for proper cleanup
- **Result**: Clean state transitions without visual artifacts

3. Comprehensive Test Coverage

- **Challenge**: Testing complex pause/preview interactions and edge cases
- **Solution**: Organized test suite with unit and integration layers
- **Result**: 44 passing tests covering all scenarios

Code Distribution Analysis

Anna's Contributions (496 LoC)

- **Popup System**: 102 LoC (21%) - New popup abstraction
- **Renderer Updates**: 26 LoC (5%) - Refactored to use popups
- **Button Manager**: 2 LoC (<1%) - Added clear() method
- **App Integration**: 1 LoC (<1%) - Minor integration
- **Testing**: 365 LoC (74%) - Comprehensive popup/start screen tests

Cody's Contributions (744 LoC)

- **Pause Unit Tests**: 214 LoC (29%) - Complete pause coverage
- **Preview Unit Tests**: 252 LoC (34%) - Preview rendering tests
- **Integration Tests**: 278 LoC (37%) - Comprehensive feature integration

UI System Evolution

Component Structure

```
src/ui/
├── button.py                    # Interactive button class
├── button_manager.py            # Button collection management (+clear)
├── pop_up.py                    # Popup abstraction (new)
├── pop_up_layout_utils.py       # Layout calculation helpers
└── pop_up_render_utils.py       # Rendering utilities
```

- **Evolution**: From separate screen implementations to unified popup system
- **Modularity**: Clear separation of concerns
- **Extensibility**: Easy to add new popup screens
- **Testability**: Isolated components for easier testing

Week 10 Focus (Looking Ahead)

- **Anna**: Complete remaining Week 10 goals (if any)
- **Cody**: Continue with remaining Sprint 2 features, expand test coverage
- **Owen**: Continue with difficulty levels and ghost piece polish
- **Team**: Final Sprint 2 completion, polish, and documentation

Sprint 2 Progress Analysis

Anna's Sprint 2 Completion

- **Features**: 2/3 completed (~67%)
- **Requirements**: 13/14 completed (~93%)

Cody's Sprint 2 Completion

- **Features**: 2/2 completed (100%)
- **Requirements**: 12/12 completed (100%)
- **Test Coverage**: Comprehensive coverage for all features
- **Quality**: All tests passing with edge case validation

Owen's Sprint 2 Scope (per plan)

- **Features**: 2 planned (difficulty levels, ghost piece)
- **Requirements**: 11 planned (5 for difficulty scaling, 4 for ghost piece, plus supporting tests)
- **Status**: Week 9 implementation update forthcoming

Test Coverage Statistics

Total Test Coverage

- **Anna**: 365 LoC in start screen/popup tests (28 tests)
- **Cody**: 744 LoC in pause/preview tests (44 tests)
- **Total New Tests**: 1,109 LoC, 72 tests
- **Pass Rate**: 100% (all tests passing)

Test Categories

- **Unit Tests**: 32 tests (pause, preview, popup)
- **Integration Tests**: 12 tests (feature interactions)
- **Start Screen Tests**: 28 tests (UI and state management)
- **Edge Cases**: Extensive coverage of corner cases

Week 9 Takeaways

Priority: UI Polish & Testing Excellence

Outcome: Game over screen complete, popup system refactored, comprehensive test suite

Next: Final Sprint 2 completion, remaining polish items

🎯 Week 9 Summary

Achievements:

- ✅ Game over screen with score display
- ✅ Unified popup system architecture
- ✅ Comprehensive test suite (72 tests)
- ✅ 100% Week 9 goal completion
- ✅ 3/4 Week 10 goals completed early

Progress:

- ~93% of Sprint 2 requirements completed (13/14)
- Features 2/3 complete for Anna; final game-over testing outstanding
- Cody completed all Sprint 2 goals (100%)