ASE 420 Team Project

Error 404: Name Not Found

Week 6 Progress Report – Tetris

📅 Week 6: Oct 20 – Oct 26

🎯 Focus: Scoring System & Next Piece Preview

Team Overview

Team Members

- **Jeffrey Perdue** – Team Leader
- **Anna Dinius** – Board & Line Clearing
- **Cody King** – Pieces & Collision
- **Owen Newberry** – Rendering & Controls

Sprint 2 Progress: 5/37 total requirements completed (14% total, 14% per week)

Week 6 Goals Summary

Anna's Goals ✅

- ✅ Implement scoring logic decoupled from rendering (testable module)
- ✅ Implement base scoring function (award points for clearing one line)
- ✅ Write initial unit tests for base scoring
- ✅ **BONUS**: Add multipliers for multiple lines (2, 3, 4)
- ✅ **BONUS**: Integrate scoring logic with the board

Cody's Goals ✅

- ✅ Create preview display area in the renderer
- ✅ Design and position the preview area consistently
- ✅ Write initial unit tests for preview display

Statistics Overview

Lines of Code Added

- **Anna**: 137 LoC total
  - `src/game/board.py` : 6 (scoring integration)
  - `src/game/game.py` : 12 (scoring integration)
  - `src/game/score.py` : 18 (new scoring module)
  - `tests/test_scoring.py` : 71 (new scoring tests)
  - `tests/test_score_utils.py` : 30 (new utility tests)

- **Cody**: 142 LoC total
  - `app.py` : 2 (preview integration)
  - `src/game/game.py` : 2 (preview integration)
  - `src/view/pygame_renderer.py` : 33 (preview rendering)
  - `src/constants.py` : 3 (preview constants)
  - `tests/test_next_piece_preview.py` : 102 (preview tests)

Total: 279 lines of code

Burn Down Rates

Week 6 Performance

- **Anna**: 100% total (3/3 goals completed + 2/3 week 7 goals)
- **Cody**: 100% total (3/3 goals completed)
- **~14% per day** progress rate
- **Perfect milestone alignment**

Sprint 2 Progress

- **14% total** (5/37 total requirements completed)
- **14% per week** average
- **~2% per day** overall progress

Major Technical Achievements

Scoring System Implementation (Anna)

- **Decoupled Architecture**: Scoring logic separated from rendering
- **Testable Module**: Complete unit test coverage for scoring functions
- **Multi-line Support**: Multipliers for 2, 3, and 4 line clears
- **Board Integration**: Scoring triggered after validated line removal

Key Architecture Changes

1. Scoring Module Creation

```
# src/game/score.py - 18 LoC (new)
def points_for_clear(lines_cleared):
    """Returns points based on lines cleared at once"""
    points_map = {
        0: 0,    # 0 lines cleared: 0 points
        1: 100,  # 1 line cleared: 100 points
        2: 300,  # 2 lines cleared: 300 points
        3: 500,  # 3 lines cleared: 500 points
        4: 800   # 4 lines cleared: 800 points
    }
    return points_map.get(lines_cleared, 0)
```

- **Purpose**: Centralized scoring logic with clear point values
- **Design**: Simple mapping for easy maintenance and testing
- **Integration**: Ready for game loop integration

2. Board Scoring Integration

```
# src/game/board.py - 6 LoC (updated)
class Board:
    def __init__(self, height, width, row_factory):
        self.__lines_cleared = 0  # Track lines cleared

    def clear_full_lines(self):
        # Line clearing logic with count tracking
        self.__lines_cleared = cleared_count
```

- **Tracking**: Added lines_cleared attribute to Board class
- **Integration**: Line counting logic in clear_full_lines()
- **API**: Clean property access for scoring system

3. Next Piece Preview Rendering

```
# src/view/pygame_renderer.py - 33 LoC (updated)
def draw_next_piece_preview(self):
    """Draw the preview box and 'Next Piece' text"""
    pygame.draw.rect(self.screen, WHITE, NEXT_PIECE_PREVIEW_RECT, 2)
    # Draw "Next Piece" label

def draw_next_piece(self, piece):
    """Draw the next piece inside preview box"""
    # Similar logic to draw_piece but positioned in preview area
```

- **Visual Design**: Consistent preview box with clear labeling
- **Positioning**: Right-side placement for optimal UX
- **Integration**: Seamless with existing piece rendering

4. Preview Constants

```
# src/constants.py - 3 LoC (updated)
NEXT_PIECE_PREVIEW_RECT = pygame.Rect(
    BOARD_WIDTH + 20, 10, 120, 80
)
```

- **Configuration**: Centralized preview area definition
- **Consistency**: Single source for preview dimensions
- **Maintainability**: Easy to adjust positioning

Testing Coverage Analysis

Test Statistics

- **Anna's Tests**: 101 LoC total
  - `test_scoring.py` : 71 LoC (6 comprehensive tests)
  - `test_score_utils.py` : 30 LoC (2 utility tests)
- **Cody's Tests**: 102 LoC total
  - `test_next_piece_preview.py` : 102 LoC (preview functionality tests)
- **Total Test LoC**: 203 lines

Code Quality Improvements

Before vs After

```
# Before: No scoring system
def clear_full_lines(self):
    # Only line clearing, no scoring

# After: Integrated scoring with tracking
def clear_full_lines(self):
    cleared_count = self._count_and_clear_lines()
    self.__lines_cleared = cleared_count
    return cleared_count
```

Benefits Achieved

- **Modularity**: Scoring logic separated from game logic
- **Testability**: Complete unit test coverage for scoring
- **Maintainability**: Clear point values and easy modification
- **Integration**: Clean API for game loop integration

Sprint 2 Progress Analysis

Completed Requirements (5/37)

- ✅ **Anna 1.1**: Scoring logic decoupled from rendering
- ✅ **Anna 1.2**: Base scoring function implementation
- ✅ **Anna 1.3**: Multi-line multipliers
- ✅ **Anna 1.4**: Board integration for scoring
- ✅ **Cody 1.1**: Preview display area in renderer

Upcoming Requirements

- **Anna**: Score display, high score tracking, start screen, game over screen (9 remaining)
- **Cody**: Preview integration, pause/resume functionality (11 remaining)

Technical Challenges Overcome

1. Scoring Architecture Design

- **Challenge**: Decoupling scoring from rendering while maintaining integration
- **Solution**: Separate scoring module with clear API contracts
- **Result**: Testable, maintainable scoring system

2. Preview Positioning

- **Challenge**: Consistent preview area placement and sizing
- **Solution**: Centralized constants and careful renderer integration
- **Result**: Professional-looking preview system

Week 6 vs Sprint 2 Planning Analysis

Anna - EXCEEDED EXPECTATIONS ✅

- **Sprint 2 Goals**: Scoring system implementation
- **Week 6 Reported**: ✅ Completed Week 6 goals + 2/3 Week 7 goals
- **Assessment**: **Ahead of schedule** - Anna completed current week plus advanced work

Cody - PERFECT ALIGNMENT ✅

- **Sprint 2 Goals**: Next piece preview implementation
- **Week 6 Reported**: ✅ Completed all preview goals with comprehensive testing
- **Assessment**: **100% on track** - Cody delivered exactly what was planned

Team Positioning for Week 7

- **Anna**: Scoring system complete, ready for advanced features
- **Cody**: Preview system complete, ready for additional UI features
- **Overall**: Strong foundation for final Sprint 2 completion

Sprint 2 Progress Update

Requirement Completion Rate

- **Week 5**: 0/37 (0%)
- **Week 6**: 5/37 (14%)
- **Projected Week 7**: 12/37 (32%)
- **Projected Week 8**: 20/37 (54%)
- **Projected Week 9**: 30/37 (81%)
- **Projected Week 10**: 37/37 (100%)

Velocity Analysis

- **Current Progress**: 14% per week average
- **Quality Focus**: 100% test coverage maintained
- **Innovation**: Advanced features beyond basic requirements
- **Team Coordination**: Parallel development with no conflicts

Week 7 Focus Areas

Upcoming Goals

- **Anna**: Score display and high score tracking
- **Cody**: Preview integration with piece generation
- **Owen**: Difficulty level framework and ghost piece logic
- **Total**: 7 additional requirements planned

Success Metrics

- **Functional Scoring**: Complete scoring system in gameplay
- **Visual Polish**: Professional-looking preview system
- **Code Quality**: Maintained test coverage and documentation
- **Performance**: Smooth gameplay with all features

Questions & Discussion

Ready for advanced scoring features?

Any concerns about preview system integration?

Final Sprint 2 completion strategy?

🎯 Week 7 Focus

Priority: Advanced Features & Integration

Goal: Complete Sprint 2 requirements with polish

Success Metrics:

- Advanced scoring features implemented
- Complete game loop integration
- Professional UI polish
- 100% Sprint 2 requirement completion