Iteration 3 Report EECE 2560: Fundamentals of Engineering Algorithms Brickbreaker Game Project

Samyuktha Jeevananthan
Department of Electrical and Computer Engineering
Northeastern University

November 4, 2025

Important: Each student must submit individually, even though the project is completed as a team. Prepare this report in **Overleaf** and export it as a **PDF** for submission.

GitHub Link: [https://github.com/YourTeamName/Brickbreaker-Project]

Contents

1 Summary of Team Progress and Development Updates

During Iteration 3 (October 19–27), our team focused on implementing and integrating the core gameplay systems of the Brickbreaker project. By this stage, the foundational structure of the game is in place, including paddle control, ball physics, and brick collision behavior.

Key accomplishments this iteration include:

- Implemented paddle and ball motion logic for smooth input handling.

- Integrated collision detection between ball, paddle, and bricks using bounding box collision logic.

- Created a functional game-over state that resets when the ball exits the screen.

- Added initial scoring functionality and brick removal logic.

Pending tasks for Iteration 4:

- Improve collision accuracy (currently inconsistent at paddle edges).

- Add sound effects and visual feedback when bricks break.

- Finalize score tracking and display in the main UI.

Overall, approximately 85

2 Implemented Core Features

The following features were completed or refined during this iteration.

Game Loop and Window Setup

Goal: Establish a functional main game window and continuous update loop using `pygame`. **Implementation:** A 600x800 pixel window was created with event handling for input, collision, and frame updates. **Validation:** The loop runs at 60 FPS with consistent object updates across multiple machines.

Paddle and Ball Mechanics

Goal: Implement user-controlled paddle movement and realistic ball bouncing behavior. **Implementation:** Paddle movement is restricted to screen boundaries. Ball speed and direction adjust upon paddle collision based on contact position. **Validation:** Tested for 100+ iterations; ball correctly reverses direction upon wall and paddle collisions.

Brick Collision System

Goal: Create grid of bricks that disappear on ball contact. **Implementation:** Bricks represented as `pygame.Rect` objects in a 6x10 grid. Collision detection removes the specific brick instance and increases score. **Validation:** Tested using manual gameplay; approximately 90 percent of collisions register correctly. Further refinement planned.

Game Over State and Reset

Goal: Implement a reset condition when the ball misses the paddle. **Implementation:** When ball Y-coordinate exceeds window height, a "Game Over" message appears, and the system waits for player input to restart. **Validation:** Game loop resets correctly without errors or duplication of assets.

3 Challenges and Resolutions

- **Collision Inaccuracy:** The ball occasionally passes through bricks at high velocity. **Resolution:** Planned to implement smaller frame time steps and continuous collision detection in Iteration 4.

- **Merge Conflicts:** Code edits by multiple members led to Git conflicts. **Resolution:** Adopted a pull-request workflow with review approval before merging.

- **Frame Rate Variance:** Ball speed differed across systems. **Resolution:** Added `pygame.time.Clock()` to lock frame rate at 60 FPS.

- **Coordination Delays:** Team members worked independently without consistent updates. **Resolution:** Instituted daily Discord progress logs and brief weekend sync meetings.

4 Leadership Rotation and Team Contributions

Leadership Summary

Week/Span	Leader / Responsibilities	Key Outcomes
Oct 19–27	**Samyuktha Jeevananthan** Coordinated task division; oversaw core logic completion.	Game loop and collision system integrated.
Oct 28–Nov 3	**Ryan Yang** Managed debugging and visual refinement.	Planned fixes for collision and UI issues.

Individual Contributions

Team Member	Contributions (Technical / Documentation)	Hours
Ryan Yang	Implemented paddle and ball mechanics; integrated collision physics.	9 hrs
Josue Argueta	Tested gameplay flow; documented bugs and progress in milestone report.	8 hrs
Samyuktha Jeevananthan	Designed UI elements (score display, game-over screen); adjusted visuals.	7 hrs

Statement by the Individual Submitter

I, **Samyuktha Jeevananthan**, confirm that the above table accurately reflects my personal contributions during Iteration 3.

5 Updated Timeline (Excerpt)

Week / Dates	Focus	Status
Week 1 (Oct 19–27)	Core development (paddle, ball, brick logic)	Completed
Week 2 (Oct 28–Nov 3)	Debugging, score system, and UI polish	Completed
Week 3 (Nov 4–10)	Add sound effects, visual feedback, and menu screen	In Progress

| Week 4 (Nov 11–17) | Final testing and documentation submission | Planned |

6 Next Steps

- Finalize score system and integrate it into the main display.

- Refine collision precision using smaller frame steps.

- Conduct full playtesting and record bug reports.

- Prepare updated Overleaf PDF and Excel timeline for next milestone submission.

7 Appendix: Space for Updates

New Bugs Found: [Placeholder] **Fixes Completed:** [Placeholder] **New Features Added:** [Placeholder]