

TOBB ETU BIL482 - Project Plan Document

Project Scope

This project covers the planning and development of **Color Clash**, a web-based and two-dimensional fighting game. The project scope is clearly defined in order to keep the game development process under control and to ensure that the defined objectives are completed on time.

Project Includes

- A 2D game running on a web browser
- Local multiplayer mode for two players using the same keyboard
- Local tournament mode:
 - Allows multiple players to participate by entering their names
 - Matches are played sequentially
 - Uses an elimination and final match structure
- At least two fighting modes (e.g., attack-oriented and defense-oriented)
- Basic game mechanics (movement, attack, defense)
- A child-friendly game design with a low level of violence

Project Excludes

- Online multiplayer or network-based gameplay
- Single-player (player vs. bot) game mode
- Blood, intense violence, or mature content
- Advanced physics engines or realistic combat simulations
- Commercial distribution, monetization, or user account systems

Project Organization – People (Roles and Responsibilities)

The project is developed by a three-member team. Responsibilities are distributed according to the project timeline and task assignments defined in the Gantt chart. All team members contribute to development, testing, and documentation activities throughout the project lifecycle.

Mehmet Umur ÖZÜ

- Contribution to project definition and overall architecture design
- Implementation of fighting modes and local multiplayer functionality
- Contribution to tournament brackets, match scheduling, and deployment
- Implementation of audio integration and sound-related components
- Participation in unit testing, bug fixing, and final quality assurance
- Contribution to architecture documentation and deployment-related reports

Mehmet Yasin TOSUN

- Contribution to project planning and UI/UX design
- Implementation of character behavior, state-based logic, and physics-related components
- Contribution to game engine development, UI component integration, and optimization
- Implementation of player registration, animations, and gameplay balancing
- Participation in integration testing and user documentation

Mustafa GÖZÜTOK

- Contribution to software requirements analysis and project documentation
- Implementation of input handling, combat system, and element-based attribute logic
- Contribution to state management and pattern-oriented design decisions
- Participation in UI testing, debugging, and gameplay logic refinement
- Contribution to design pattern mapping and final pattern report

Shared Responsibilities

- Architectural discussions and design decisions
- Development environment setup
- Cross-browser testing and bug fixing
- Code documentation, final QA, and project presentation
- GitHub-based version control and final submission

Project Objectives (Execution and Management)

- To complete the project according to the defined timeline
- To ensure balanced and traceable task distribution within the team
- To identify and resolve technical issues early during development
- To apply design patterns in a meaningful and practical way within the project
- To evaluate design trade-offs and justify pattern selections based on project requirements and constraints
- To maintain modularity and separation of concerns throughout the development process

Key Phases and Timeline

The detailed project schedule, including phases, tasks, and time allocation, is provided in **Appendix 1**.

Resource Planning

The basic resources to be used in the project are:

- HTML, CSS, and JavaScript
- Simple 2D graphics and sprite assets
- GitHub for version control and team collaboration
- Web browsers (Chrome, Firefox, etc.) for testing purposes

Risk Management

Risk	Description	Mitigation
Asset incompatibility	Inconsistencies in size, format, or visual style between character and background assets	Define common asset standards and perform early asset integration
Asset development time	Visual improvements taking longer than expected	Use placeholder assets initially and refine visuals incrementally
Game balance	Power imbalance between fighting modes affecting fairness	Conduct early playtesting and apply numerical balancing
Tournament logic errors	Flow issues in multi-player match sequencing	Use a simple tournament algorithm and step-by-step testing
Time management	Delays in implementing certain features	Prioritize core features and postpone optional components
Small team dependency	The project is developed by a three-person team, which may cause workload imbalance if a task takes longer than expected	Adopt an agile working approach where tasks can be reallocated and team members support each other as needed
Task overlap risk	Parallel tasks may depend on each other and cause blocking	Maintain continuous communication and adjust task order dynamically
Team availability	Temporary unavailability of a team member	Ensure knowledge sharing and keep code and documentation accessible to all team members

Communication Plan

- Weekly team meetings to track progress
- Daily short communications via online messaging tools
- Tracking all contributions through GitHub commit history

Change Management Plan

- Change requests **will be discussed and evaluated** collectively by the team
- The potential impact on project scope, timeline, and technical complexity **will be analyzed**
- Priority **will be given** to core gameplay features and required deliverables
- If necessary, non-essential or optional features **will be simplified or postponed**
- Approved changes **will be reflected** in the project plan and task timeline

Budget Plan

- **Planning and Design (≈ 20%)**
Time and effort will be allocated to requirements analysis, project planning, software architecture design, and design pattern selection.
- **Development (≈ 45%)**
The largest portion of effort will be dedicated to implementing core gameplay mechanics, local multiplayer functionality, fighting modes, tournament logic, and audio integration.
- **Testing and Refinement (≈ 20%)**
Effort will be allocated to unit testing, integration testing, gameplay balancing, bug fixing, and cross-browser testing.
- **Documentation and Submission (≈ 15%)**
Time will be reserved for preparing technical documentation, architecture reports, pattern mapping, final presentation, and GitHub submission.

Acceptance Tests and Acceptance Criteria

- The game must function correctly in all defined game modes
- Local multiplayer and tournament modes must be fully implemented
- Player names must be entered in the tournament mode and matches must proceed in the correct order
- At least two fighting modes must be integrated into the game
- The game must maintain a low level of violence and be suitable for children

Appendix

Appendix 1) Gantt Chart

Color Clash - Project Gantt Chart

TOBB ETU BIL 482 | 12-Week Development Timeline

