

Integrative project - Big two Specification

Group A5

LAU King

Project manager

BARBUTOV Filip

Development User Interface

AKAKPO Mensanh Crepin

KAMEL Seifeldin

Development Database and Server

CARPENTIER-KEITZL Edgar

IMHOFF Guillaume

KHADJIEV Djakhar

Development Game Mechanics and Algorithms

Game Description

Presentation

Big Two is a popular card game played with a standard 52-card deck. The goal of the game is to be the first player to get rid of all his cards by making valid combinations. This game combines strategy, observation and speed, making it an ideal choice for a digital version adaptation, both in online multiplayer mode and single player against artificial intelligence (AI).

Rules of the game

- ❖ **Distribution of cards** : Each player receives 13 cards at the beginning of the game.
- ❖ **Start of the game** : The player with the 3 of ♦ (three of diamonds) starts the game.
- ❖ **How a turn is played:**
 - In turn, a player may:
 - Lay a **single card** or a **valid combination** (pair, three of a kind, five cards, etc.) that must be higher than the previous combination.
 - **Pass** if he cannot or does not want to play.
 - If all players except one pass in succession, the round ends. The played cards are discarded, and a new turn begins, initiated by the last player to have played.
- ❖ **Valid combinations:**
 - Single card.
 - Pair (two cards of the same value).
 - Brekan (three cards of the same value).
 - Sequence of five cards (like in poker: straight, flush, full house, four of a kind, straight flush).
- ❖ **End of round:** The round ends when a player has no cards left.
- ❖ **End of the game:** The game ends and displays the winner and players must decide whether to play again.

Transposition into a Digital Version

The **Big Two** game will be developed as an application available both on **PC** and as a **web version**. It will be an **online multiplayer game** allowing users to compete in real-time or play against an **artificial intelligence (AI)**. The goal of the client is to create a **user-friendly** application that provides a **smooth, competitive, and engaging gaming experience**, with social features such as a **ranking system**.

Technical Implementation

1. Home Screen:

- ❖ Players will be able to **register** and **log in** to their accounts.
- ❖ Choice between:
 - **Solo Mode:** Compete against an AI.
 - **Multiplayer Mode:** Play against other users online.

2. Game Engine:

- ❖ **Card Distribution:** The game engine will randomly deal **13 cards** to each player.
- ❖ **Turn Management:** The engine will handle **turn order**, starting with the player holding the **3 of ♦**.
- ❖ **Combination Validation:** An algorithm will verify the played **combinations** to ensure they follow the game rules and are **stronger** than the previous combination.
- ❖ **Game End Detection:** The engine will detect when a player has **no cards left** and will display the **winner**.

3. Player Interface:

- ❖ Display of the **player's hand** and the **cards played on the table**.
- ❖ **Drag-and-drop** functionality to select and play card combinations.
- ❖ Buttons to **play a move, pass a turn, or leave the game**.
- ❖ **System notifications** (e.g., "**Player X played a pair of 7s**").

4. Communication System:

- ❖ **System Messages:** Inform players of **important events** (turn start, round end, etc.).

Client-Server Architecture

The **Big Two** game is built on a **client-server model** to ensure a **smooth and interactive experience**.

- ❖ **Client:**
 - Responsible for **displaying the user interface** (cards, buttons).
 - Sends **player actions** (playing cards, passing a turn) to the server.
 - Receives **updates from the server** (game state, played cards).

❖ **Server:**

- Manages the **game logic**: card distribution, combination validation, player synchronization.
- Ensures **real-time communication** between clients using protocols like **WebSocket**.
- Validates all player actions to **prevent cheating**.

Exchanged Data

The communication between client and server will be **secure and structured** to ensure **real-time synchronization**.

❖ **From Client to Server:**

- **Login information** (username, password).
- **Player actions** (played cards, turn passes).

❖ **From Server to Client:**

- **Game state updates** (card distribution, played cards, turn progress).
- **System notifications** (turn start, round end, etc.).

Game Logic and Algorithms

Turn Execution Algorithm:

Start of the Turn

1. The **active player** receives control.
2. If it's the **first turn**, the player holding the **3 of ♦** starts.
3. The player can:
 - a. **Play a valid combination** that is **stronger** than the previous one.
 - b. **Pass their turn**.
4. If **all players except one pass**, the turn **ends**.
5. The **last player to have played** starts the new turn.

End of Turn

Combination Validation Algorithm:

ValidateCombination(combination, previous_combination):

1. Check if the **combination is valid** (single card, pair, three of a kind, etc.).
2. If a **previous combination exists**:
 - a. Verify that the **new combination is stronger**.
3. **Return True if valid, otherwise False.**

Key Elements

The development of the **Big Two** project will require several essential technical skills to ensure a **smooth and secure gaming experience**.

Client-Server Architecture

The game will be designed to function in **multiplayer mode** using a **client-server architecture**.

- ❖ The **server** will manage **player connections, validate played moves, and synchronize ongoing games**.
- ❖ The **clients** (user interfaces) will be responsible for **displaying the game and sending player actions to the server**.

Database

A **database** will be set up on the server to **store and manage various information**, including:

- ❖ **Player credentials**, such as **username, email, and password (hashed for security)**.

Security

Client-server communication will be **secured** through **HTTPS and TLS encryption**.

- ❖ **Player authentication** will be managed via **email and password** or an **OAuth system**.
- ❖ To **prevent cheating, access control mechanisms** will be implemented, ensuring that **all player actions are validated on the server side**.

Game Logic and Algorithms

The game will include a **game engine** responsible for:

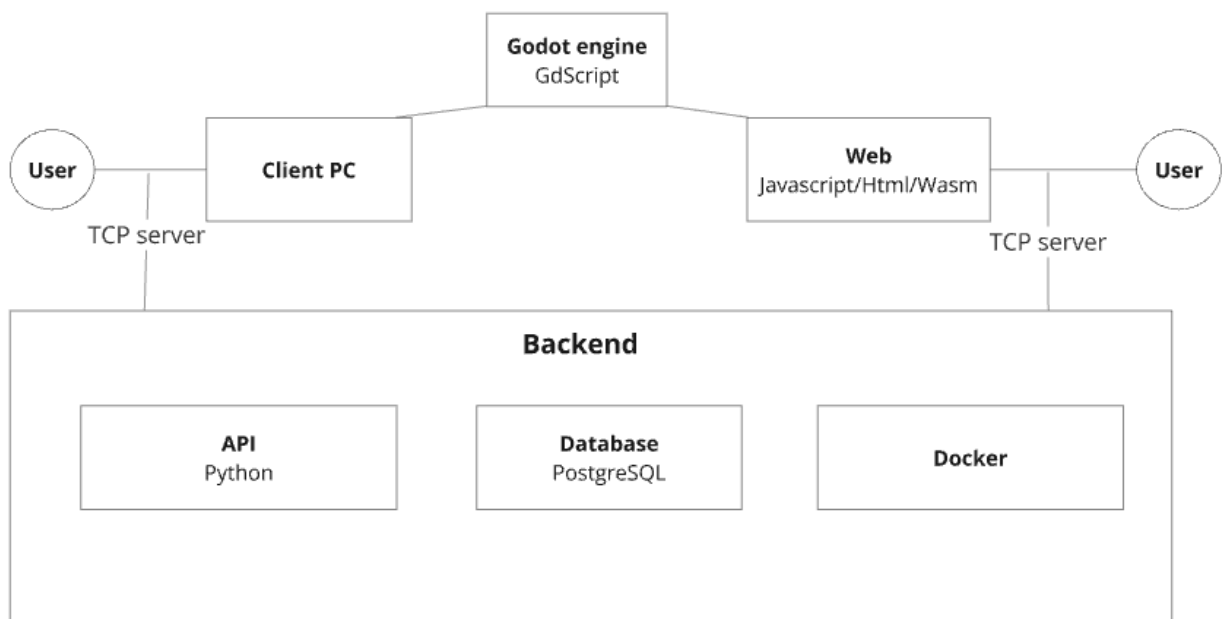
- ❖ **Validating card combinations** and enforcing **Big Two** rules.
- ❖ **Managing turn order**, ensuring that each player plays in sequence.
- ❖ **Developing an AI opponent**, allowing players to compete against virtual opponents in **solo mode**.

User Interface and Gaming Experience

The **user interface** will be designed to be **intuitive and user-friendly**.

- ❖ Players will be able to **drag and drop** their cards to easily select and play combinations.
- ❖ Smooth **animations** and **visual effects** will be integrated to provide an **engaging and immersive** gaming experience.

Functional Diagram



Technologies Used

The development of the Big Two game requires the use of modern and suitable technologies to ensure optimal performance, scalability, secure data management, and a smooth user experience. Below is a detailed presentation of the chosen technologies, accompanied by justifications for each selection.

Game Engine

Godot Engine

- ❖ **Description:** Godot is an open-source, lightweight, and high-performance game engine, particularly well-suited for 2D games like Big Two.
- ❖ **Justification:**
 - **Performance:** Optimized for 2D games, it ensures a smooth experience even on modest devices.
 - **Native Multiplayer Support:** Godot provides native support for network communications via TCP/UDP, which is essential for real-time multiplayer mode.
 - **GDScript:** Its integrated scripting language, GDScript, is easy to learn and similar to Python, which accelerates development.
 - **HTML5 Export:** Allows exporting the game to the web version, making it accessible directly from a browser without requiring installation.

Programming Languages

GDScript

- ❖ **Description:** GDScript is the integrated scripting language in Godot, designed for game development.
- ❖ **Justification:**
 - **Native Integration:** Perfectly suited for Godot, it enables efficient management of game logic (rules, turns, combinations).
 - **Simplicity:** The syntax is similar to Python, which makes development and debugging easier.
 - **Performance:** Lightweight and optimized for games, ensuring fast execution.

Python

- ❖ **Description:** A server-side programming language used for data management, authentication, and backend logic.
- ❖ **Justification:**
 - **Flexibility:** Ideal for manipulating databases, handling HTTP requests, and implementing complex algorithms.

- **Community and Libraries:** A large range of libraries (such as Flask, SQLAlchemy) facilitates backend development.
- **Scalability:** Well-suited for handling a large number of simultaneous connections, which is crucial for a multiplayer game.

Database

PostgreSQL

- ❖ **Description:** Open-source relational database management system.
- ❖ **Justification:**
 - **Performance:** Capable of handling complex transactions and large volumes of data.
 - **Security:** Provides advanced features for access control and data encryption.
 - **Scalability:** Suitable for scaling with the increase in users and game sessions.

Backend (Server)

Python (Flask ou FastAPI)

- ❖ **Description:** Lightweight web framework for handling HTTP requests, authentication, and database communication.
- ❖ **Justification:**
 - **Simplicity:** Easy to configure and maintain.
 - **Performance:** Well-suited for handling real-time requests required for a multiplayer game.
 - **Integration with Godot:** Allows smooth communication between the client (Godot) and server via WebSocket.

Docker

- ❖ **Description:** Containerization platform for deploying and managing backend services.
- ❖ **Justification:**
 - **Portability:** Enables consistent deployment across different environments (development, testing, production).
 - **Scalability:** Facilitates server scaling based on load.
 - **Maintenance:** Simplifies updates and continuous deployments.

Frontend (Game Client)

Godot (HTML5 Export)

- ❖ **Description:** Exporting the game to the web version for maximum accessibility.
- ❖ **Justification:**
 - **Accessibility:** Players can access the game directly from their browser without installation.
 - **Compatibility:** Works across all operating systems (Windows, macOS, Linux) and devices (PC, tablets, smartphones).
 - **Performance:** Godot ensures a smooth experience even within a browser.

Hosting and Deployment

Unistra Server

- ❖ **Description**
- ❖ The Big Two project will be hosted on two dedicated servers provided by the University of Strasbourg (Unistra). These servers are OpenStack virtual machines with the following specifications:
 - **CPU:** 4 virtual cores
 - **RAM:** 4 GB
 - **Storage:** 200 GB
 - **Operating System:** Ubuntu 22.04 Server

These servers will be used to host essential services for the project, such as:

- ❖ Web server (for the user interface and APIs)
- ❖ Database (for storing player information, scores, etc.)
- ❖ Backend services (game management, real-time communication via WebSocket)

Security

TLS (HTTPS)

- ❖ **Description:** Encryption protocol to secure communications between the client and the server.
- ❖ **Justification:**
 - **Confidentiality:** Protects sensitive data (usernames, passwords) from interception.
 - **Integrity:** Ensures that data is not modified during transmission.

OAuth

- ❖ **Description:** Secure authentication protocol allowing users to log in via third-party accounts (Google, Apple).
- ❖ **Justification:**
 - **Ease of Use:** Players can log in without creating a new account.
 - **Security:** Reduces risks associated with password management.

Outils Complémentaires

GitLab

- ❖ **Description:** Version control system for tracking source code.
- ❖ **Justification:**
 - **Collaboration:** Facilitates team collaboration and version management.
 - **History:** Allows reverting to a previous version in case of issues.

Trello

- ❖ **Description:** Trello is a project management tool based on the Kanban method. It allows organizing tasks in boards, lists, and cards, making it easier to track project progress in real-time.
- ❖ **Justification:**
 - **Visual Management:** Trello allows representing project progress in a clear and intuitive way through its card-based interface.
 - **Team Collaboration:** Team members can assign tasks, add comments, attach files, and set deadlines.
 - **Effective Tracking:** It helps structure work with columns (e.g., To Do, In Progress, Done) and add labels to better organize tasks.

Vision of the Project Manager Role

The **project manager** of the **Big Two** game must ensure **coordination**, **planning**, and smooth development of the game while maintaining the **quality** of the game and meeting **deadlines**.

Roles and Responsibilities

❖ Objective Definition and Planning

- Write the **project brief** and define the **project scope**.
- Establish a **schedule** with clear **milestones** (registration/login, game engine, interface, etc.).
- Assign **tasks** to team members based on their **skills**.

❖ Coordination and Communication

- Organize regular **meetings** to track progress and adjust the **roadmap**.
- Ensure effective **communication** between **developers**, **designers**, and **testers**.
- Act as a **liaison** with **stakeholders** (players/testers, potential sponsors).

❖ Technical and Quality Management

- Oversee the **project architecture** (database, client-server, security).
- Ensure **code consistency** and enforce good **development practices** (CI/CD, unit tests).
- Validate **technical choices** and ensure **project documentation**.

❖ Risk and Issue Tracking

- Identify and anticipate **technical roadblocks**.
- Find **solutions** and adjust the **schedule** in case of **delays**.
- Manage **technical disagreements** by prioritizing a **compromise** based on the project's best interests.

❖ Team Management and Motivation

- Create a **positive** and **collaborative work environment**.
- Encourage **team members**, value their **ideas**, and resolve **conflicts** constructively.
- Keep **motivation** high by celebrating **successes** and setting **achievable goals**.

Key Qualities of the Project Manager

- ★ **Leadership:** Ability to guide the team and make decisions.
- ★ **Organization:** Time management and task prioritization.
- ★ **Communication:** Ability to listen and transmit information effectively.
- ★ **Problem-Solving:** Adaptability in the face of unforeseen challenges.
- ★ **Team Spirit:** Encourage collaboration and manage conflicts calmly.

Tasks and Teams + Gantt

The breakdown of the teams based on the assigned tasks:

Analysis and Design

- ❖ **Members:** King, Seif, Mensanh, Filip, Tout le groupe
- ❖ **Tasks:**
 - Definition of rules and features (King)
 - Definition of valid combinations (King)
 - Formalization of game rules into algorithms (King)
 - Design of software architecture (client-server, database) (Seif, Mensanh)
 - Development of interface mockups (Filip, Seif)
 - Selection of technologies (DB, Backend, WebSockets) (Entire Team)
 - Design of the database schema (Seif, Mensanh)
 - Writing specifications (Game, Database, Backend) (Entire Team)

Database

- ❖ **Members:** Mensanh, Seif, Filip
- ❖ **Tasks:**
 - Creation of tables (Mensanh, Seif)
 - Implementation of tables (Mensanh, Seif)
 - Definition of relationships and constraints (Mensanh, Seif)
 - Implementation of associated Python functions (Filip, Seif, Mensanh)

Alpha Development (Offline, AI only)

- ❖ **Members:** Filip, Seif, Mensanh, Edgar, Djakhar, Guillaume
- ❖ **Tasks:**
 - Implementation of player registration and login (Filip, Seif, Mensanh)
 - Account creation and management system (Filip, Seif, Mensanh)
 - Password security (Filip, Seif, Mensanh)
 - Frontend interface implementation (Edgar, Djakhar, Guillaume)

- Display of cards, game interface (Edgar, Djakhar, Guillaume)
- Interaction management (drag-and-drop) (Edgar, Djakhar, Guillaume)
- Backend connection (Edgar, Djakhar, Guillaume)
- Session management (Edgar, Djakhar, Guillaume)
- Client-server communication (player actions) (Edgar, Djakhar, Guillaume)
- Card rendering engine and animations (Filip, Seif, Mensanh)
- User event management (Filip, Seif, Mensanh)
- Game engine development (Edgar, Djakhar, Guillaume)
- Card distribution (Edgar, Djakhar, Guillaume)
- Turn management (Edgar, Djakhar, Guillaume)
- Combination validation (Edgar, Djakhar, Guillaume)

Beta Development (Online + AI)

- ❖ **Members:** The whole group
- ❖ **Tasks:**
 - Online game management
 - Implementation of WebSockets
 - Game synchronization
 - Server-side rule verification
 - Real-time validation of player actions
 - Error handling and invalidations
 - Player synchronization setup
 - Real-time game state updates
 - Communication with the database

Server Setup and Deployment

- ❖ **Members:** Mensanh, Seif, Filip, Guillaume, Edgar, Djakhar
- ❖ **Tasks:**
 - Backend server preparation (Mensanh, Seif)
 - Initial configuration (Docker) (Mensanh, Seif)
 - TLS security setup (Filip, Guillaume)
 - Security testing (Filip, Guillaume)
 - Server modifications for beta (Edgar, Djakhar)

Testing and Debugging

- ❖ **Members:** The whole group
- ❖ **Tasks:**

- Unit and integration tests
- Game engine verification
- Client-server interaction testing
- Server load testing
- Large-scale game simulation
- Performance optimization
- Bug fixing

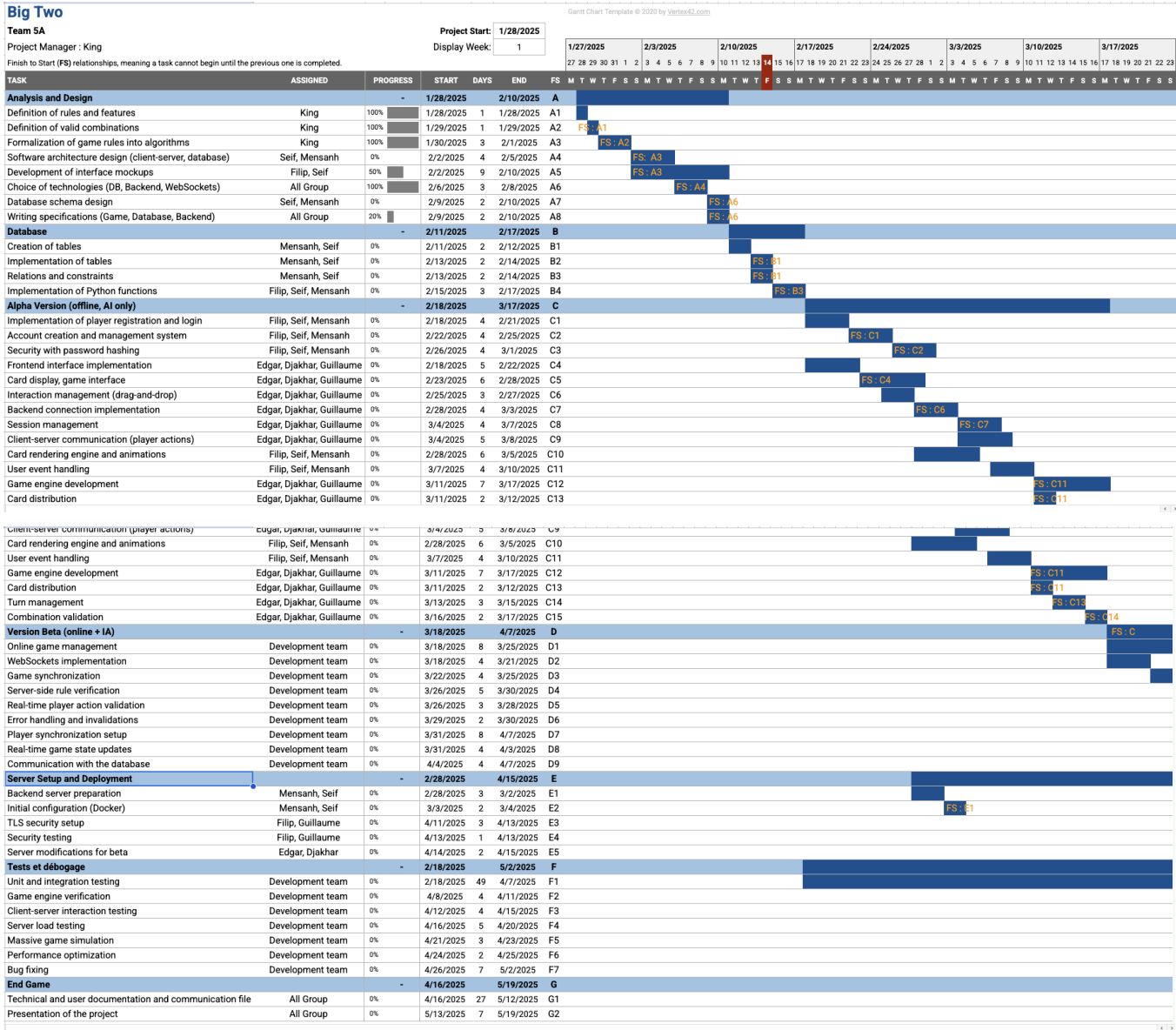
Project Finalization

- ❖ **Members:** The whole group
- ❖ **Tasks:**
 - Writing technical and user documentation
 - Project presentation

GANTT :

<https://docs.google.com/spreadsheets/d/15E0ZqsRuYU18wtngXLdZtjkoHMBxgvyk6md2QXh5oIPc/edit?usp=sharing>

Gantt - Week (1-8)



[illegible]

Gantt - Week (10-17)

Big Two

Team 5A

Project Manager : King

Finish to Start (FS) relationships, meaning a task cannot begin until the previous one is completed.

Project Start:1/28/2025

Display Week:10

3/31/2025

4/7/2025

4/14/2025

4/21/2025

4/28/2025

5/5/2025

5/12/2025

5/19/2025

31

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

TASK	ASSIGNED	PROGRESS	START	DAYS	END	FS	M	T	W	T	F	S	M	T	W	T	F	S	M	T	W	T	F	S	M	T	W	T	F	S
Analysis and Design						-	1/28/2025			2/10/2025			A																	
Definition of rules and features	King	100%	1/28/2025	1	1/28/2025	A1																								
Definition of valid combinations	King	100%	1/29/2025	1	1/29/2025	A2																								
Formalization of game rules into algorithms	King	100%	1/30/2025	3	2/1/2025	A3																								
Software architecture design (client-server, database)	Seif, Mensanh	0%	2/2/2025	4	2/5/2025	A4																								
Development of interface mockups	Filip, Seif	50%	2/2/2025	9	2/10/2025	A7																								
Choice of technologies (DB, Backend, WebSockets)	All Group	100%	2/6/2025	3	2/8/2025	A5																								
Database schema design	Seif, Mensanh	0%	2/9/2025	2	2/10/2025	A6																								
Writing specifications (Game, Database, Backend)	All Group	20%	2/9/2025	2	2/10/2025	A8																								
Database						-	2/11/2025			2/17/2025			B																	
Creation of tables	Mensanh, Seif	0%	2/11/2025	2	2/12/2025	B2																								
Implementation of tables	Mensanh, Seif	0%	2/13/2025	2	2/14/2025	B1																								
Relations and constraints	Mensanh, Seif	0%	2/13/2025	2	2/14/2025	B3																								
Implementation of Python functions	Filip, Seif, Mensanh	0%	2/15/2025	3	2/17/2025	B4																								
Alpha Version (offline, AI only)						-	2/18/2025			3/17/2025			C																	
Implementation of player registration and login	Filip, Seif, Mensanh	0%	2/18/2025	4	2/21/2025	C1																								
Account creation and management system	Filip, Seif, Mensanh	0%	2/22/2025	4	2/25/2025	C2																								
Security with password hashing	Filip, Seif, Mensanh	0%	2/26/2025	4	3/1/2025	C3																								
Frontend interface implementation	Edgar, Djakhar, Guillaume	0%	2/18/2025	5	2/22/2025	C4																								
Card display, game interface	Edgar, Djakhar, Guillaume	0%	2/23/2025	6	2/28/2025	C5																								
Interaction management (drag-and-drop)	Edgar, Djakhar, Guillaume	0%	2/25/2025	3	2/27/2025	C6																								
Backend connection implementation	Edgar, Djakhar, Guillaume	0%	2/28/2025	4	3/3/2025	C7																								
Session management	Edgar, Djakhar, Guillaume	0%	3/4/2025	4	3/7/2025	C8																								
Client-server communication (player actions)	Edgar, Djakhar, Guillaume	0%	3/4/2025	5	3/8/2025	C9																								
Card rendering engine and animations	Filip, Seif, Mensanh	0%	2/28/2025	6	3/5/2025	C10																								
User event handling	Filip, Seif, Mensanh	0%	3/7/2025	4	3/10/2025	C11																								
Game engine development	Edgar, Djakhar, Guillaume	0%	3/11/2025	7	3/17/2025	C12																								
Card distribution	Edgar, Djakhar, Guillaume	0%	3/11/2025	2	3/12/2025	C13																								
Client-server communication (player actions)	Edgar, Djakhar, Guillaume	0%	3/4/2025	5	3/8/2025	C9																								
Card rendering engine and animations	Filip, Seif, Mensanh	0%	2/28/2025	6	3/5/2025	C10																								
User event handling	Filip, Seif, Mensanh	0%	3/7/2025	4	3/10/2025	C11																								
Game engine development	Edgar, Djakhar, Guillaume	0%	3/11/2025	7	3/17/2025	C12																								
Card distribution	Edgar, Djakhar, Guillaume	0%	3/11/2025	2	3/12/2025	C13																								
Turn management	Edgar, Djakhar, Guillaume	0%	3/13/2025	3	3/15/2025	C14																								
Combination validation	Edgar, Djakhar, Guillaume	0%	3/16/2025	2	3/17/2025	C15																								
Version Beta (online + IA)						-	3/18/2025			4/7/2025			D																	
Online game management	Development team	0%	3/18/2025	8	3/25/2025	D1																								
WebSockets implementation	Development team	0%	3/18/2025	4	3/21/2025	D2																								
Game synchronization	Development team	0%	3/22/2025	4	3/25/2025	D3																								
Server-side rule verification	Development team	0%	3/26/2025	5	3/30/2025	D4																								
Real-time player action validation	Development team	0%	3/26/2025	3	3/28/2025	D5																								
Error handling and invalidations	Development team	0%	3/29/2025	2	3/30/2025	D6																								
Player synchronization setup	Development team	0%	3/31/2025	8	4/7/2025	D7																								
Real-time game state updates	Development team	0%	3/31/2025	4	4/3/2025	D8																								
Communication with the database	Development team	0%	4/4/2025	4	4/7/2025	D9																								
Server Setup and Deployment						-	2/28/2025			4/15/2025			E																	
Backend server preparation	Mensanh, Seif	0%	2/28/2025	3	3/2/2025	E1																								
Initial configuration (Docker)	Mensanh, Seif	0%	3/3/2025	2	3/4/2025	E2																								
TLS security setup	Filip, Guillaume	0%	4/11/2025	3	4/13/2025	E3																								
Security testing	Filip, Guillaume	0%	4/13/2025	1	4/13/2025	E4																								
Server modifications for beta	Edgar, Djakhar	0%	4/14/2025	2	4/15/2025	E5																								
Tests et débogage						-	2/18/2025			5/2/2025			F																	
Unit and integration testing	Development team	0%	2/18/2025	49	4/7/2025	F1																								
Game engine verification	Development team	0%	4/8/2025	4	4/11/2025	F2																								
Client-server interaction testing	Development team	0%	4/12/2025	4	4/15/2025	F3																								
Server load testing	Development team	0%	4/16/2025	5	4/20/2025	F4																								
Massive game simulation	Development team	0%	4/21/2025	3	4/23/2025	F5																								
Performance optimization	Development team	0%	4/24/2025	2	4/25/2025	F6																								
Bug fixing	Development team	0%	4/26/2025	7	5/2/2025	F7																								
End Game						-	4/16/2025			5/19/2025			G																	
Technical and user documentation and communication file	All Group	0%	4/16/2025	27	5/12/2025	G1																								
Presentation of the project	All Group	0%	5/13/2025	7	5/19/2025	G2																								