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

Summary

1. Game Description

a. Introduction

Big Two is a popular card game originating from East Asia. It is played with a standard deck of 52 cards and accommodates four players. The objective is to be the first to discard all of one's cards by playing valid combinations.

This game relies on strategy, observation, and quick decision-making. It is well-suited for a digital adaptation featuring an online multiplayer mode and a solo mode against artificial intelligence (AI). Its strategic depth makes it appealing to both casual and competitive players.

b. Game Rules

i. Card Distribution

1. Each player receives 13 cards.
2. The game uses a standard 52-card deck without jokers.

ii. Game Start

1. The player holding the 3 of Diamonds (3♦) goes first.
2. They must play the 3♦ either alone or as part of a valid combination.

iii. Game Mechanics

1. Players take turns in a clockwise direction.
2. A player can:
3. Play a stronger combination than the previous one.
4. Pass if they cannot or do not wish to play.
5. If all players except one pass, the round ends. The played cards are discarded, and the last player to play starts a new round.

iv. Number of Cards per Turn

The first combination played determines the number of cards required for that round.

Example:

1. If a player plays a pair of 7s (7♠ 7♦), subsequent players must play a higher pair (e.g., 9♣ 9♥) or pass.
2. If the round starts with a three-of-a-kind (e.g., 5♠ 5♦ 5♥), following players must play a higher three-of-a-kind (e.g., 8♣ 8♥ 8♠) or pass.

v. Valid Combinations

1. **Single Card**: Any individual card.
2. **Pair**: Two cards of the same rank.
3. **Three-of-a-Kind**: Three cards of the same rank.
4. **Five-Card Combinations**:
5. **Straight**: Five consecutive cards of any suit (e.g., 5♣ 6♦ 7♠ 8♥ 9♣).
6. **Flush**: Five cards of the same suit (e.g., A♦ 8♦ 6♦ 5♦ 3♦).
7. **Full House**: A three-of-a-kind and a pair (e.g., 10♣ 10♦ 10♥ 7♠ 7♣).
8. **Four-of-a-Kind**: Four cards of the same rank plus one extra card (e.g., Q♣ Q♦ Q♥ Q♠ 5♦).
9. **Straight Flush**: A straight where all cards are of the same suit (e.g., 6♦ 7♦ 8♦ 9♦ 10♦).

vi. Combination Hierarchy (From Weakest to Strongest)

1. Single Card
2. Pair
3. Three-of-a-Kind
4. Straight
5. Flush
6. Full House
7. Four-of-a-Kind
8. Straight Flush

c. Hand Rankings

i. Card Rank (From Strongest to Weakest)

1. 2 > A > K > Q > J > 10 > 9 > 8 > 7 > 6 > 5 > 4 > 3

ii. Suit Order (From Strongest to Weakest)

1. ♠ Spades > ♥ Hearts > ♣ Clubs > ♦ Diamonds

iii. Comparison Rules

1. **Single Card**: The highest-ranked card wins. If tied, the suit order is used.

   Example: K♠ beats K♥ (since ♠ is stronger than ♥).

2. **Pair**: Compare the rank first, then the suit.

   Example: J♠ J♦ beats 10♣ 10♥ (J is stronger than 10).

3. **Three-of-a-Kind**: The highest-ranked set wins.

   Example: 6♠ 6♥ 6♦ beats 5♠ 5♣ 5♦.

4. **Straight**: Compare the highest card first, then the suit.

   Example: 8♠ 9♠ 10♠ J♣ Q♦ beats 7♠ 8♥ 9♦ 10♣ J♣ (Q > J).

5. **Flush**: The highest card wins. If tied, compare the suits.

   Example: A♠ K♠ J♠ 9♠ 3♠ beats A♥ K♥ J♥ 9♥ 3♥.

6. **Full House**: Compare the three-of-a-kind first, then the pair.

   Example: 10♠ 10♣ 10♦ 8♥ 8♣ beats 9♠ 9♥ 9♦ A♦ A♣.

7. **Four-of-a-Kind**: Compare the four matching cards, then the extra card.

   Example: Q♠ Q♥ Q♣ Q♦ 5♠ beats J♠ J♥ J♣ J♦ A♦.

8. **Straight Flush**: Compare the highest card first, then the suit.

   Example: A♠ K♠ Q♠ J♠ 10♠ beats A♦ K♦ Q♦ J♦ 10♦ (♠ is stronger than ♦).

iv. Tiebreaker Criteria

1. Compare the highest card.
2. If still tied, use suit order (♠ > ♥ > ♣ > ♦).

d. 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**.

e. Technical Implementation

i. Home Screen:

1. Players will be able to **register** and **log in** to their accounts.
2. Choice between:

Solo Mode: Compete against an AI.

Multiplayer Mode: Play against other users online.

ii. Game Engine:

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

iii. Player Interface:

1. Display of the **player's hand** and the **cards played on the table**.
2. Functionality to **select** and **play** card combinations.
3. Buttons to **play a move, pass a turn, or leave the game**.
4. **System notifications** (e.g., "**Player X played a pair of 7s**").

iv. Communication System:

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

f. Client-Server Architecture

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

i. Client:

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

ii. Server:

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

iii. Exchanged Data:

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

iv. **From Client to Server:**

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

v. **From Server to Client:**

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

g. Game Logic and Algorithms

i. **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

ii. **Combination Validation Algorithm:**

ValidateCombinationFunction(combination, previous_combination):

1. Check if the **combination is valid**.
2. If a **previous combination exists**:
    a. Verify that the **new combination is stronger**.

Return True if valid, otherwise False.

2. Key Elements

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

a. Client-Server Architecture

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

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

b. Database

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

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

c. Security

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

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

d. Game Logic and Algorithms

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

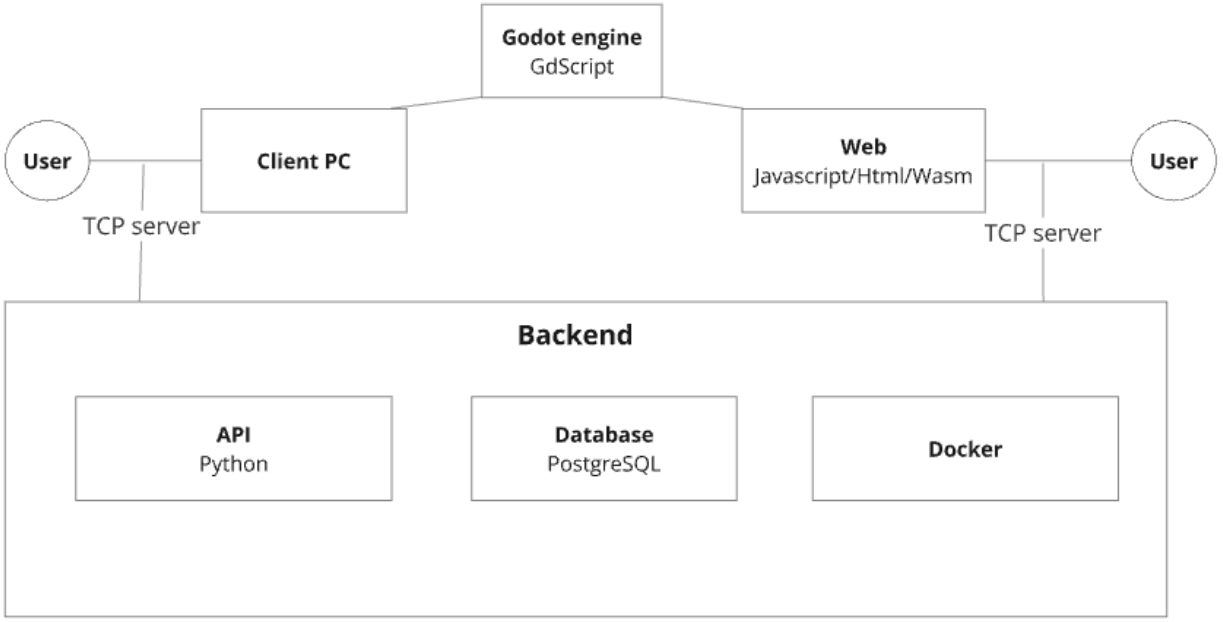
i. **Validating card combinations** and enforcing **Big Two rules**.
ii. **Managing turn order**, ensuring that each player plays in sequence.
iii. **Developing an AI opponent**, allowing players to compete against virtual opponents in **solo mode**.

e. User Interface and Gaming Experience

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

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

f. Functional Diagram



3. 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.

a. Game Engine

i. Godot Engine

Description: Godot is an open-source, lightweight, and high-performance game engine, particularly well-suited for 2D games like Big Two.

Justification:

1. **Performance**: Optimized for 2D games, it ensures a smooth experience even on modest devices.

2. **Native Multiplayer Support**: Godot provides native support for network communications via TCP/UDP, which is essential for real-time multiplayer mode.
3. **GDScript**: Its integrated scripting language, GDScript, is easy to learn and similar to Python, which accelerates development.
4. **HTML5 Export**: Allows exporting the game to the web version, making it accessible directly from a browser without requiring installation.

b. Programming Languages

i. GDScript

Description: GDScript is the integrated scripting language in Godot, designed for game development.

Justification:

1. **Native Integration**: Perfectly suited for Godot, it enables efficient management of game logic (rules, turns, combinations).
2. **Simplicity**: The syntax is similar to Python, which makes development and debugging easier.
3. **Performance**: Lightweight and optimized for games, ensuring fast execution.

ii. Python

Description: A server-side programming language used for data management, authentication, and backend logic.

Justification:

1. **Flexibility**: Ideal for manipulating databases, handling HTTP requests, and implementing complex algorithms.
2. **Community and Libraries**: A large range of libraries (such as Flask, SQLAlchemy) facilitates backend development.

3. **Scalability**: Well-suited for handling a large number of simultaneous connections, which is crucial for a multiplayer game.

c. Database

i. PostgreSQL

Description: Open-source relational database management system.

Justification:

1. **Performance**: Capable of handling complex transactions and large volumes of data.
2. **Security**: Provides advanced features for access control and data encryption.
3. **Scalability**: Suitable for scaling with the increase in users and game sessions.

d. Backend (Server)

i. Python (Flask ou FastAPI)

Description: Lightweight web framework for handling HTTP requests, authentication, and database communication.

Justification:

1. **Simplicity**: Easy to configure and maintain.
2. **Performance**: Well-suited for handling real-time requests required for a multiplayer game.
3. **Integration with Godot**: Allows smooth communication between the client (Godot) and server via WebSocket.

ii. Docker

Description: Containerization platform for deploying and managing backend services.

Justification:

1. **Portability**: Enables consistent deployment across different environments (development, testing, production).
2. **Scalability**: Facilitates server scaling based on load.
3. **Maintenance**: Simplifies updates and continuous deployments.

e. Frontend (Game Client)

i. Godot (HTML5 Export)

Description: Exporting the game to the web version for maximum accessibility.

Justification:

1. **Accessibility**: Players can access the game directly from their browser without installation.
2. **Compatibility**: Works across all operating systems (Windows, macOS, Linux) and devices (PC, tablets, smartphones).
3. **Performance**: Godot ensures a smooth experience even within a browser.

f. Hosting and Deployment

i. 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:

1. **CPU**: 4 virtual cores
2. **RAM**: 4 GB
3. **Storage**: 200 GB
4. **Operating System**: Ubuntu 22.04 Server

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

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

g. Security

i. TLS (HTTPS)

Description: Encryption protocol to secure communications between the client and the server.

Justification:

1. **Confidentiality**: Protects sensitive data (usernames, passwords) from interception.
2. **Integrity**: Ensures that data is not modified during transmission.

ii. OAuth

Description: Secure authentication protocol allowing users to log in via third-party accounts (Google, Apple).

Justification:

1. **Ease of Use**: Players can log in without creating a new account.
2. **Security**: Reduces risks associated with password management.

h. Outils Complémentaires

i. GitLab

Description: Version control system for tracking source code.

Justification:

1. **Collaboration**: Facilitates team collaboration and version management.
2. **History**: Allows reverting to a previous version in case of issues.

ii. 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:

1. **Visual Management**: Trello allows representing project progress in a clear and intuitive way through its card-based interface.
2. **Team Collaboration**: Team members can assign tasks, add comments, attach files, and set deadlines.
3. **Effective Tracking**: It helps structure work with columns (e.g., To Do, In Progress, Done) and add labels to better organize tasks.

4. 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**.

a. Roles and Responsibilities

i. Objective Definition and Planning

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

ii. Coordination and Communication

1. Organize regular **meetings** to track progress and adjust the **roadmap**.

2. Ensure effective **communication** between **developers**, **designers**, and **testers**.
3. Act as a **liaison** with **stakeholders** (players/testers, potential sponsors).

iii. **Technical and Quality Management**

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

iv. **Risk and Issue Tracking**

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

v. **Team Management and Motivation**

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

b. Key Qualities of the Project Manager

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

5. Tasks and Teams + Gantt

The breakdown of the teams based on the assigned tasks:

a. Analysis and Design

i. **Members**: Mensanh, Seifeldin, Filip, Guillaume, Edgar, Djakhar, King

ii. **Tasks**:

1. Definition of rules and features
2. Definition of valid combinations
3. Formalization of game rules into algorithms
4. Design of software architecture (client-server, database)
5. Development of interface mockups
6. Selection of technologies (DB, Backend, WebSockets)
7. Design of the database schema
8. Writing specifications (Game, Database, Backend)

b. Database

i. **Members**: Mensanh, Seifeldin, Filip

ii. **Tasks**:

1. Creation of tables
2. Implementation of tables
3. Definition of relationships and constraints
4. Implementation of associated Python functions

c. Alpha Development (Offline, AI only)

i. **Members**: Mensanh, Seifeldin, Filip, Guillaume, Edgar, Djakhar

ii. **Tasks**:

1. Implementation of player registration and login
2. Account creation and management system
3. Password security
4. Frontend interface implementation
5. Display of cards, game interface
6. Interaction management (drag-and-drop)
7. Backend connection
8. Session management
9. Client-server communication (player actions)

10. Card rendering engine and animations
11. User event management
12. Game engine development
13. Card distribution
14. Turn management
15. Combination validation

d. Beta Development (Online + AI)

i. **Members**: Mensanh, Seifeldin, Filip, Guillaume, Edgar, Djakhar

ii. **Tasks**:

1. Online game management
2. Implementation of WebSockets
3. Game synchronization
4. Server-side rule verification
5. Real-time validation of player actions
6. Error handling and invalidations
7. Player synchronization setup
8. Real-time game state updates
9. Communication with the database

e. Server Setup and Deployment

i. **Members**: Mensanh, Seifeldin, Filip, Guillaume, Edgar, Djakhar

ii. **Tasks**:

1. Backend server preparation
2. Initial configuration (Docker)
3. TLS security setup
4. Security testing
5. Server modifications for beta

f. Testing and Debugging

i. **Members**: Mensanh, Seifeldin, Filip, Guillaume, Edgar, Djakhar

ii. **Tasks**:

1. Unit and integration tests
2. Game engine verification
3. Client-server interaction testing
4. Server load testing
5. Large-scale game simulation
6. Performance optimization
7. Bug fixing

g. Marketing and Communication

i. **Members**: Seifeldin, Filip, King

ii. **Tasks**:

1. Development of marketing strategy
2. Creation of the visual identity (logo, colours, design)
3. Creation and management of social networks
4. Content production (videos, teasers)
5. Social media marketing
6. Organization of a launch event

h. Project Finalization

i. **Members**: Mensanh, Seifeldin, Filip, Guillaume, Edgar, Djakhar, King

ii. **Tasks**:

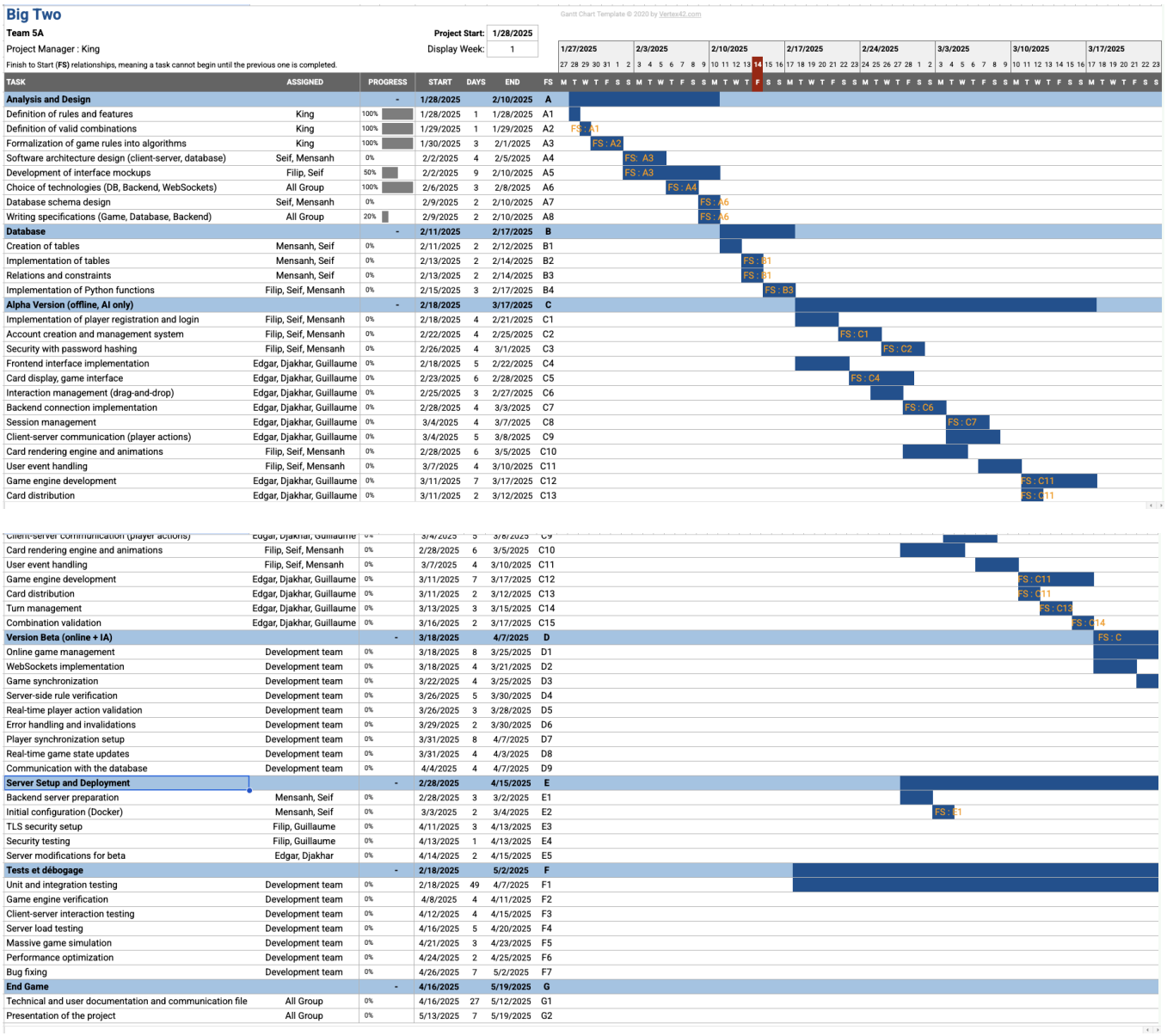
1. Writing technical and user documentation
2. Project presentation

i. GANTT

Link:

https://docs.google.com/spreadsheets/d/15E0ZqsRuYU18wtngXLdZtjkoHMbxgvk6md2QXh5olPc/edit?usp=sharing

Gantt - Week (1-8)



Big Two

Gantt Chart Template © 2020 by Vertex42.com

Team 5A — Project Start: 1/28/2025

Project Manager : King — Display Week: 1

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

TASK	ASSIGNED	PROGRESS	START	DAYS	END	FS
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	A5
Choice of technologies (DB, Backend, WebSockets)	All Group	100%	2/6/2025	3	2/8/2025	A6
Database schema design	Seif, Mensanh	0%	2/9/2025	2	2/10/2025	A7
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	B1
Implementation of tables	Mensanh, Seif	0%	2/13/2025	2	2/14/2025	B2
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
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

Gantt - Week (8-15)

Big Two

Gantt Chart Template © 2020 by Vertex42.com

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: 8

TASK	ASSIGNED	PROGRESS	START	DAYS	END	FS
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
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

Gantt - Week (10-17)

Big Two

Gantt Chart Template © 2020 by Vertex42.com

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

TASK	ASSIGNED	PROGRESS	START	DAYS	END	FS
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

TASK	ASSIGNED	PROGRESS	START	DAYS	END	FS
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