

1. General(horizontal game)

1.1 Card Distribution

At the beginning of the game, each player is dealt 8 cards. These cards consist of army cards and fate cards. Army cards belong to four different suits/colors and have attack and defense values. Fate cards have two types: Spells and Curses, each with unique effects.

1.2 Rounds

The game progresses in rounds. In each round, one player is designated as the defender, and the other as the attacker.

1.3 Determining Battle Results

To win a round, the attacker's total attack value must be higher than the defender's total defense value. The attacker's total attacking power is calculated by adding the attack values of their army cards. The defender's total defense strength is calculated by adding the defense values of their army cards. In case of a tie in total strengths, the round ends in a draw, and both players discard their armies.

1.4 Fate Cards

Fate cards are divided into Spells and Curses and can be used to influence battle outcomes. Spells provide various advantages, such as drawing cards, modifying hands, or multiplying hand values. Curses can hinder opponents by stealing cards or reducing hand values.

Spells can:

- Allow a player to draw a card from the deck.
- Combine a hand with cards of different colors and types.
- Multiply the hand value by five.
- Steal curses from opponents.
- End curses.
- Prevent curses from affecting the color or type of cards.

Curses can:

- Steal an army card from the opponent.
- Remove up to four army cards from the game.
- Steal spells from the opponent.
- End spells.
- Divide the hand value by two for a specified color.
- Prevent the casting of spells.
- Any spell or curse can be canceled if the corresponding card is played.

2. Game Mechanics

The following mechanics govern the implementation of the game "YET BATTLE ONLINE":

2.1 Card Management

Implement shuffling and dealing of cards at the beginning of each game. Track the hands of each player, including army and fate cards.

2.2 Battle Resolution

Calculate and compare the total attack and defense values during battles. Determine round results, including victories, draws, and card discards.

2.3 Fate Card Effects

Implement unique effects of Spells and Curses that allow players to use them strategically. Include a mechanism to cancel the effects of fate cards when the corresponding cards are played.

2.4 Turn Sequence

Implement a turn sequence where players take turns as attackers and defenders in rounds.

2.5 Additional Features

Consider adding special properties to certain cards, as described on the cards themselves. Implement the option for a player to skip a turn if they have no playable cards. Ensure that fate card effects are applied immediately upon playing. Fate cards can be used to influence the outcome of battles.

Game Screens:

1. Main Menu:

- Start
- Navigation Map
- Settings
- Exit

Options:

- Sound and Music
- Language (eng)

2. Lore Screen (you can't skip it)

3. Navigation Map Screen

Will be designed later

4. **Game Screen:**

- Display of cards in the player's hand
- Playing field with cards in battle (ігальне поле)
- Statistics of the current round (сумарні дані карт у раунді)
- Game status (attacker's or defender's turn)

5. **Round Results Screen:**

- Displaying the results of the fight
- Information about who won the round
- Ability to move on to the next round

6. **Game Completion Screen:**

- Display of the winner and final score
- Options to **start a new game** or **return to the main menu (exit)**

General Plan and Tasks for Developer:

1. Game Logic and Mechanics:

1.1 Card Distribution:

- Implement logic for distributing 8 cards to each player at the beginning of the game.
- Define the attributes and values for the army and fate cards.

1.2 Rounds:

- Develop the round system(3 max rounds, the player that wins two of them is the winner, for the last round it will be a coin flip for attacker and defender role), where players take turns as attackers and defenders.
- Establish a mechanism to switch roles between players in each round.

1.3 Determining Battle Results:

- Create algorithms to calculate total attack and defence values during battles.
- Implement logic to determine round results (victory, draw, or discard).

1.4 Fate Cards:

- Design and implement the functionality of Spells and Curses.
- Develop the effects of each spell and curse as described in the game design

2. Game Mechanics:

2.1 Card Management:

- Code the shuffling and dealing of cards at the start of each game.
- Implement tracking mechanisms for player hands, including army and fate cards.

2.2 Battle Resolution:

- Develop algorithms to calculate and compare total attack and defence values during battles.
- Determine and implement round results, including handling victories, draws, and card discards.

2.3 Fate Card Effects:

- Code the unique effects of Spells and Curses.
- Implement a mechanism to cancel fate card effects when the corresponding cards are played.

2.4 Turn Sequence:

- Establish a turn sequence where players take turns as attackers and defenders in rounds.
- Implement the logic for switching between attacker and defender roles.

2.5 Additional Features:

- Add special properties to certain cards as described in the game design.
- Implement the option for a player to skip a turn if they have no playable cards.
- Ensure immediate application of fate card effects upon playing.

3. Game Screens:

3.1 Main Menu:

- Develop UI elements for Start, Navigation Map, Settings, and Exit.
- Implement options for adjusting sound, music, and language settings.

3.2 Lore Screen:

- Create an introductory screen with information about the game's lore.
- Make the lore screen mandatory, ensuring it can't be skipped.

3.3 Navigation Map Screen:

- Design and implement the navigation map screen.

3.4 Game Screen:

- Develop UI for displaying player hands, the playing field, round statistics, and game status.
- Implement game logic for playing cards in battle.

3.5 Round Results Screen:

- Design UI to display the results of the round.
- Include information about the round winner and an option to move to the next round.

3.6 Game Completion Screen:

- Develop UI for displaying the winner and final score.
- Include options to start a new game or return to the main menu (exit).

4. Testing and Debugging:

4.1 Unit Testing:

- Conduct thorough testing for each implemented feature.
- Debug and resolve any issues found during testing.

4.2 User Acceptance Testing (UAT):

- Involve testers to perform UAT to identify any user experience issues.
- Address feedback and make necessary adjustments.

5. Finalization

5.1 Optimization:

- Optimize code for performance on both iOS and Android platforms.

5.2 Cross-Platform Compatibility:

- Ensure that the game runs seamlessly on both iOS and Android devices.

5.3 Deployment:

- Prepare the game for deployment on the App Store (iOS) and Google Play (Android).

5.4 Documentation:

- Document the codebase, mechanics, and any necessary instructions for future reference.