Name: Ayden Kelley Mark _____/50

1. Brief introduction _/3

My feature for our game Galleon Gambit is the menus, which connect the flow of the game and also provide a beginning and end to the game. These menus include the starting screen, pause & settings, end of battle rewards, the game over screen, and the game won screen. My feature is therefore split into a couple of parts throughout the game.

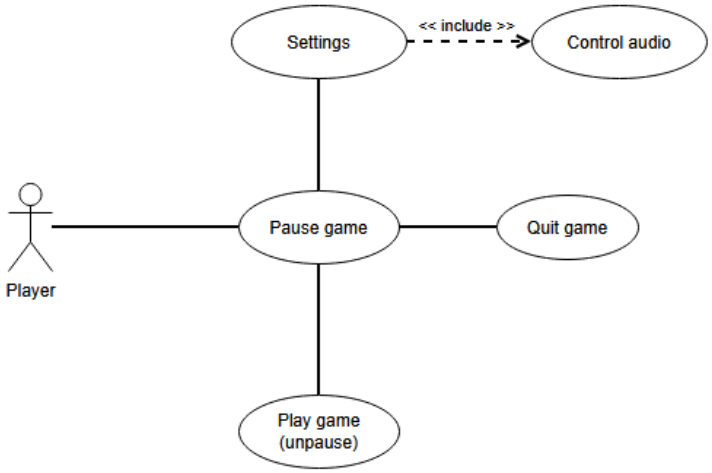
When the player loads the game, I need to present them with a start menu that lets them enable/disable BC mode as well as start the game. I need to have a pause and settings menu that the player can open at any time throughout the game that freezes everything else and allows the player to control certain settings like audio levels.

Whenever the player wins a battle, I need to provide a screen that summarizes and grants their rewards for winning that battle and lets them progress to the next part of the game when they are ready.

Finally, I need two game end menus. One is if the player dies and therefore loses, offering the option to retry. The second is if the player wins, which will also offer the option to play again.

2. Use case diagram with scenario _14

The primary use case diagram for my feature centers around the end of battle rewards menu which interacts with the main game flow. The other use case diagram for my feature involves the pause and settings menu.



Scenarios

Name: End of Battle Rewards

Summary: When the game detects a battle has been won, it will grant the player rewards for winning. Once all rewards have been processed, the player can continue onto the next level.
Actors: Battle System
Preconditions: Player won a battle level.
Basic sequence:

Step 1: Detect type of battle won. Continue if regular battle was won.
Step 2: Award the player an amount of money calculated by some formula.
Step 3: Prompt the player to pay for healing and heal the player.
Step 4: Randomly generate 3 possible cards from the deck card database.
Step 5: Player chooses one of the cards to add to their deck.
Step 6: Add the chosen card to the deck.
Step 7: Once all rewards are processed, prompt the player to continue to next level.

Exceptions:

Step 1: Boss battle was won, not a regular battle. Then, the player wins the game.
Step 3: Player chooses to not pay for healing. Then, no healing is given.
Step 5: Player chooses not to add any of the cards to their deck. Then, don't add any cards.

Post conditions: Applicable rewards have been given to the player, and the game proceeds to the map for the next level.
Priority: 2*
ID: C01
*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

Name: Start Game
Summary: Either upon launching the game or after a failed or successful playthrough of the game, the player begins a fresh new playthrough.
Actors: Player, Loss Detection
Preconditions: Either the player won or lost a previous playthrough, or the game was just launched.
Basic sequence:

Step 1: If game won or game lost, proceed to start screen.
Step 2: If game just launched, proceed to start screen.
Step 3: Prompt player to turn on BC mode.
Step 4: Start fresh, new game playthrough.

Exceptions:

Step 3: If BC mode not enabled, game mode is regular difficulty.

Post conditions: BC mode is either enabled or disabled, and game proceeds to a new randomly generated map and a reset player.
Priority: 2*

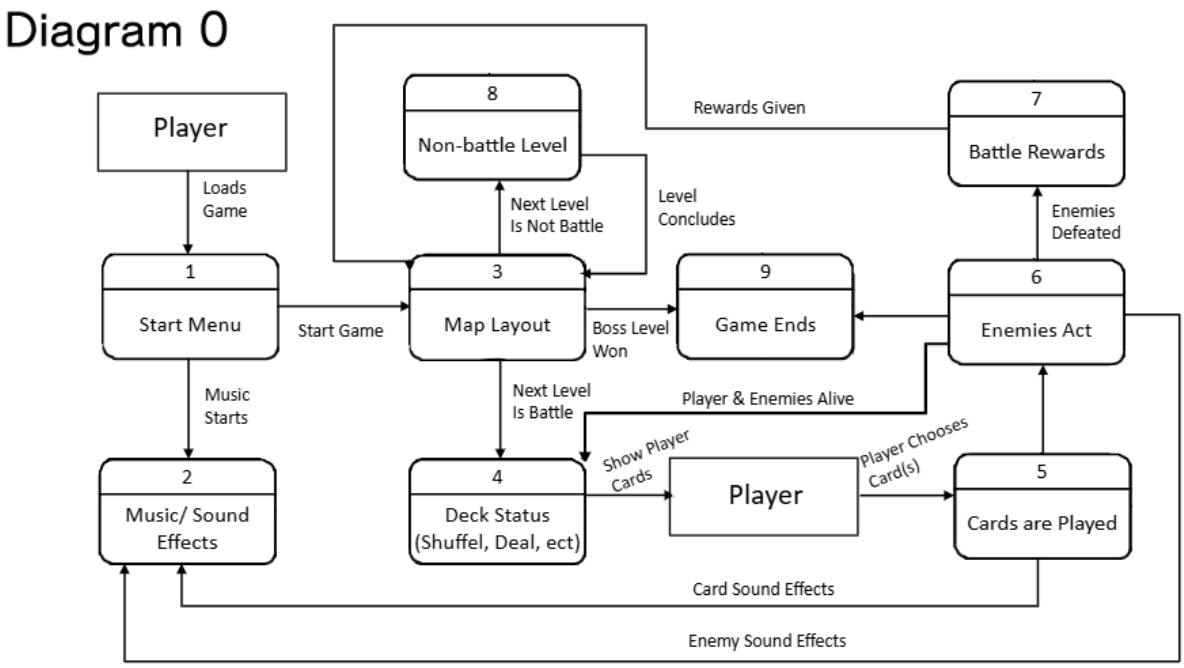
ID: C02
*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

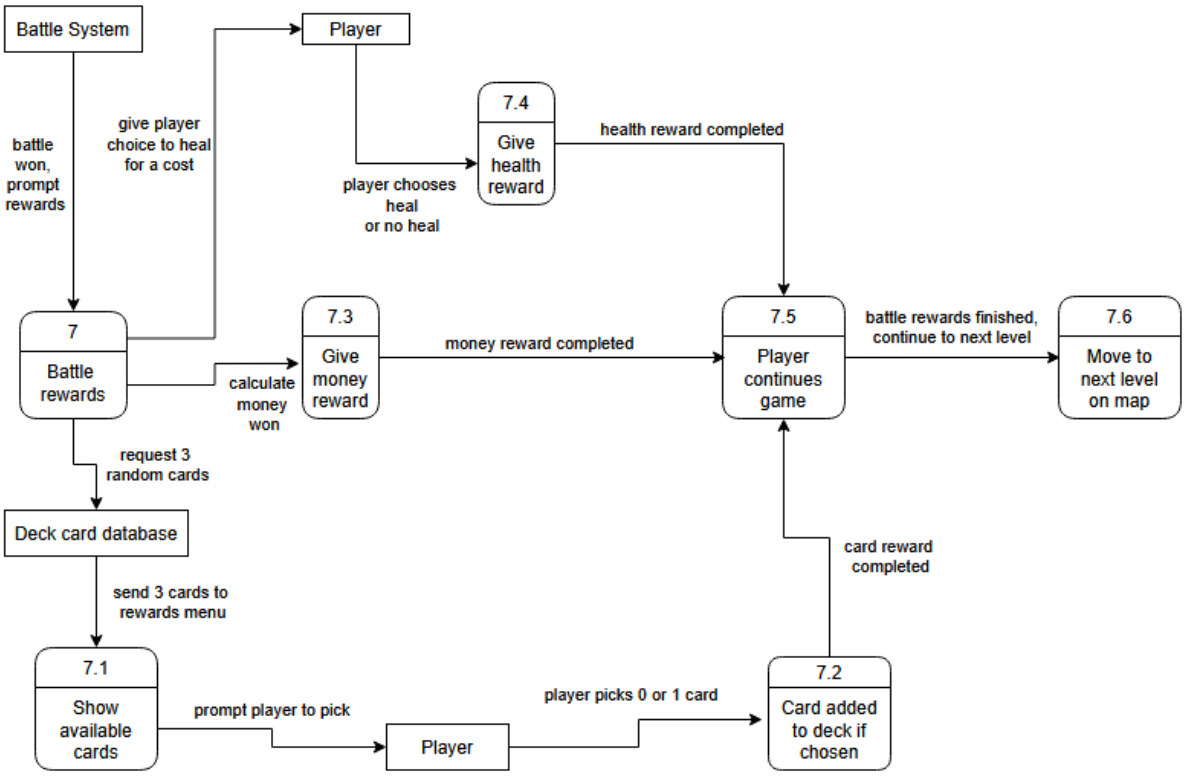
Name: Pause Game
Summary: The player is able to pause the game, freezing all game activity, and modify sound or quit the game.
Actors: Player
Preconditions: Pause key is pressed.
Basic sequence:

Step 1: Player presses pause key, bringing up the pause menu and freezing all game activity.
Step 2: Player can go to settings and change audio.
Step 3: Player can quit the game.
Step 4: Player can unpause, unfreezing game activity.

Post conditions: Game is unpaused, and game activity resumes as normal.
Priority: 2*
ID: C03 OK period
*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

3. Data Flow diagram(s) from Level 0 to process description for your feature _____14





Process Descriptions

7.0 Battle rewards:

Process money reward

Process health reward

Process deck card reward

7. 1 Show available cards:

Reward Card 1 = Random card from database

Reward Card 2 = Random card from database

Reward Card 3 = Random card from database

7.2 Card added to deck if chosen:

IF player picked card, add chosen card to player deck

ELSE do not add card to player deck

7.3 Give money reward:

Money reward = formula with some randomization + factor in enemies defeated

Player money += money reward

7.4 Give health reward:

IF player has enough money (X), then

IF player chooses to pay for heal, then player health += Y, and subtract X money from player money

ELSE player does not heal

ELSE player can't heal

7.5 Player continues game:

When player clicks next, then transition back to map layout.

7.6 Move to next level on map:

Determine what next level is, move there on map, load next level.

4. Acceptance Tests _____9

Volume Control

Randomly adjust volume 1000 times, record what volume value this gives. The values should be between 0 and a decided upon maximum sound. Ensure that the volume is properly set back within the volume boundary.

Battle rewards – add new card to deck

This feature involves randomly pulling three unique cards from the database of possible new deck cards. These three cards are then given to the player to choose from to add to their deck. Run feature 1000 times and record output. Each instance of pulling from the database should pull three unique cards. Each possible card should have been pulled a reasonably equal number of times.

Battle rewards – player healing

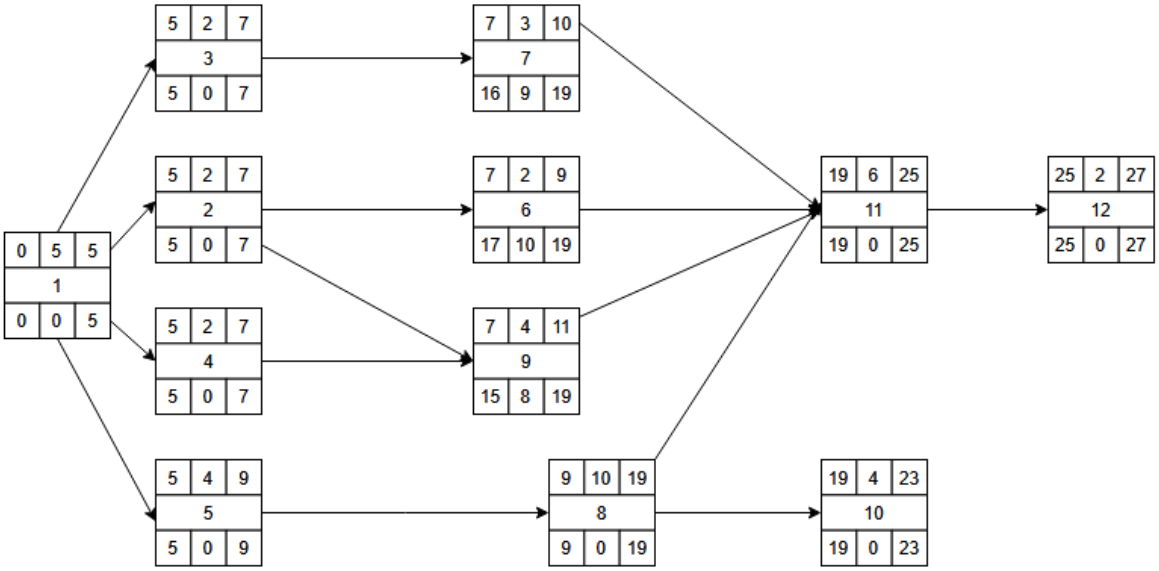
This feature involves the player getting the choice to heal X amount of life one time for Y amount of money. Attempt to continuously heal 20 times, ensure the player is prevented from doing so. Attempt to heal when the player does not have enough money, ensure the player is prevented from doing so.

5. Timeline _____/10

Work items

Task	Duration (hours)	Predecessor Task(s)
1. Requirements Collection	5	-
2. Start menu design	2	1
3. Pause menu design	2	1
4. Game end menu design	2	1
5. Battle rewards menu design	4	1
6. BC mode enable/disable	2	2
7. Sound settings	3	3
8. Battle rewards programming	10	5
9. Game end and game loop	4	2, 4
10. After battle transition programming	4	8
11. Testing & Documentation	6	6, 7, 8, 9, 10
12. Installation	2	11

Pert diagram



Gantt chart

Key

Work hours

Slack