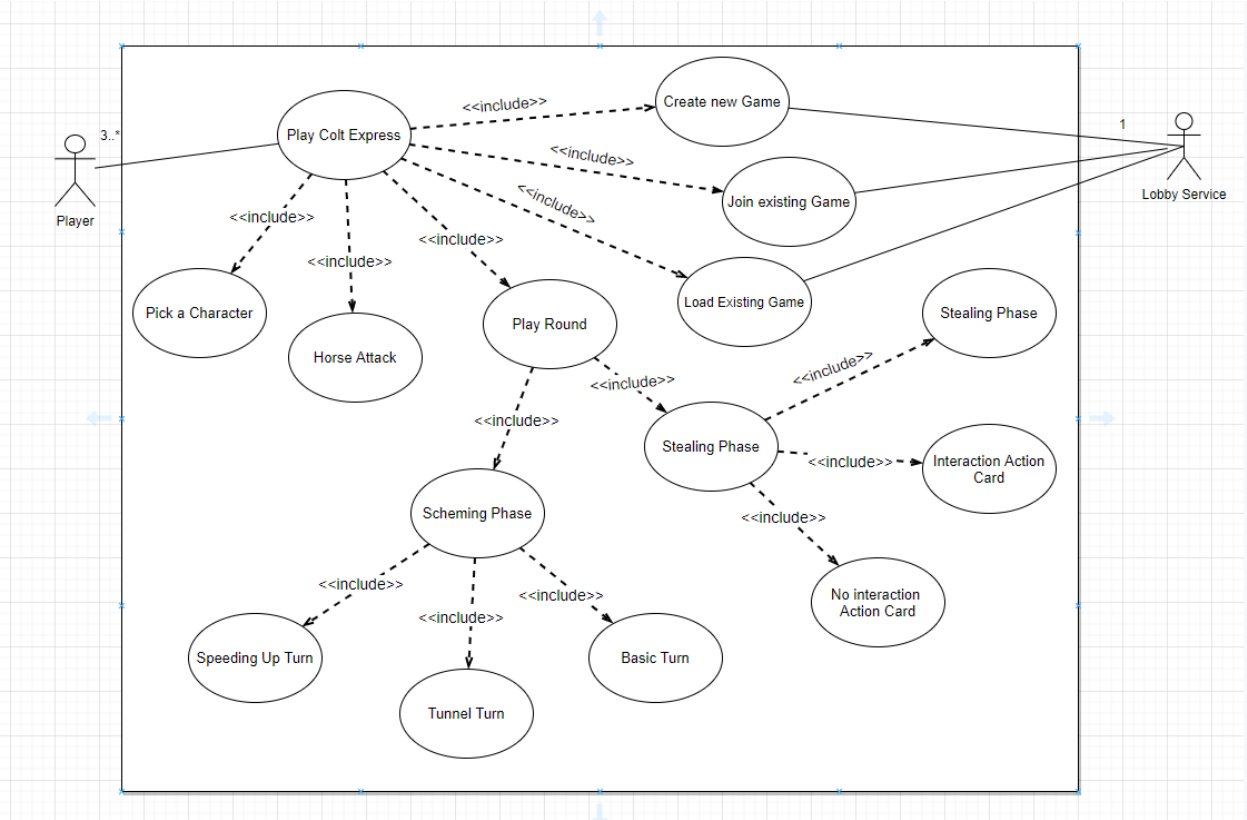
Use Case Model



Play Colt Express

Use Case: Play Colt Express
Scope: ColtExpress
Level: User Goal
Intention in Context: The intention of the *Player* is to play a game of Colt Express against other players.
Multiplicity: Multiple *Players* can play Colt Express concurrently. A given player cannot play multiple games at the same time.
Primary Actor: *Player*
Secondary Actor: *Player (who play the roles of opponents)*
Main Success Scenario:

1. *Player* logs onto *System*
2. *System* presents to *Player* the existing list of games (Colt Express).
3. *Player* chooses to create a new game, join an existing game or load an existing game. *Step 4 is only executed when System determines that enough players have joined the game.*
4. *Players* Pick A Character
5. *Players* place their corresponding Bandit during the Horse Attack.

6. *Players* play Rounds.
   Step 4 is executed 5 times so that 5 rounds are played.
7. *System* informs *Players* about which *Player(s)* receives the title of Gunslinger.
8. *System* informs all *Players* about who won the game.

Extensions:

4a. *Player* was not able to create, join or load an existing game. Use Case continues at step 3.

CreateNewGame

Use Case: CreateNewGame
Scope: ColtExpress
Level: Subfunction
Intention in Context: The intention of the *Player* is to start a new game from scratch.
Primary Actor: *Player*
Secondary Actor: *Lobby Service*
Main Success Scenario:

1. *Player* informs *System* to create a new game for x number of players.
2. *System* registers at the *Lobby Service*.
3. The *Lobby Service* creates a random unique id for the newly created session.
4. The *Lobby Service* marks *Player* as the responsible creator.
5. *System* presents the id and proposed board to *Player*.
6. *Player* informs *System* that he is ready to begin game.
   System determines that enough players have joined.

Extensions:

6a. *Player* informs *System* that he wishes to regenerate another board (i.e. the placement of the Loot). Use Case continues at step 5.
(6-7)b. *Player* informs *System* that he wishes to cancel game creation. Use Case ends in failure.

JoinExistingGame

Use Case: JoinExistingGame
Scope: ColtExpress
Level: Subfunction
Intention in Context: The intention of the *Player* is to join a game that someone else has started.
Primary Actor: *Player*
Secondary Actor: *Lobby Service*
Main Success Scenario:

1. *Player* issues *System* a join request using the specific session id.
2. *System* presents the session id to the *Lobby Service.*
3. *Lobby Service* determines if the provided id matches any existing ids.
4. *System* presents game board from the matched session to *Player.*
5. *Player* informs *System* that he is ready to begin game.
   System determines that enough players have joined.

Extensions:

3a. The *Lobby Service* informs *System* that the id matches an existing id. Use Case continues at step 4.
3b. The *Lobby Service* informs *System* that the id does not match with any existing id. Use Case ends in failure.
5a. *Player* informs *System* that he does not wish to join this game. Use Case ends in failure.

LoadExistingGame

Use Case: LoadExistingGame
Scope: ColtExpress
Level: Subfunction
Intention in Context: The intention of the *Player* is to load a previously saved game.
Primary Actor: *Player*
Secondary Actor: *Lobby Service*
Main Success Scenario:

1. *Player* informs *System* that he wishes to load a specific game by providing the Savegame identifies.
2. *System* presents the identifies to the *Lobby Service*.
3. *Lobby Service* determines if the identifies matches any existing identifies.
4. *System* presents game board to *Player*.
5. *Player* informs *System* that he is ready to begin game.
   System determines that the number of participants matches.
   System map the players in order of appearance.

Extensions:
3a. The *Lobby Service* informs *System* that the identifies matches with an existing one. Use Case continues at step 4.
3b. The *Lobby Service* informs *System* that the identifies does not match with any existing ones. Use Case ends in failure.
5a. *Player* informs *System* that he does not wish to join this game. Use Case ends in failure.

Pick a Character

Use Case: Pick Character
Scope: Colt Express
Level: Subfunction
Intention in Context: Player wants to choose which avatar to adopt in the game state
Multiplicity: All Players simultaneously vie for available characters
Primary Actor: Player
Secondary Actor: Players (play the role of opponents)
Main Success Scenario:

1. *System* presents *Players* with character options
   Step 2 and 3 repeat until all players have selected a character
2. *Player* informs *System* with which character they would like to play
3. *System* binds the game state with the concept that the *Player* has the chosen characters attributes, and updates the availability of that character for choice.
4. *System informs* all *Players* of the update character options.

Horse Attack

Use Case: Horse Attack
Scope: ColtExpress
Level: Subfunction
Intention in Context: The intention of the *Player* is to place their Bandit in his initial position.
Multiplicity: Multiple *Players* are choosing the initial position of their corresponding Bandit simultaneously. Therefore, the *Players* are seeing the Bandits being positioned in real time.
Primary Actor: *Player*

Main Success Scenario:

System informs Players their Bandits will be riding in on their horses.

1. *System* informs *Players* they can choose the initial position of their Bandit.
   A Player's Bandit can be placed in any Car except the Locomotive.
2. *System* informs *Players* they must choose to enter the Car or Ride on.
3. *Players* inform *System* about their corresponding decision.
4. *System* moves the remaining Bandits who are on their horses up a Car towards the Locomotive.
5. *System* informs all *Players* about new game state.
   Steps 2, 3, 4 and 5 are repeated until no Bandits remain on their horse.

Extensions:
2a. If the last remaining *Players'* Bandits have ridden to the last Car before the Locomotive, *System* places the last remaining Bandits who are still riding in this Car. Use Case continues at step 5. *There are now no Bandits remaining on horses.*

PlayRound

Use Case: PlayRound
Scope: ColtExpress
Level: Subfunction
Intention in Context: The intention of the *Players* is to play the round.
Primary Actor: *Player*
Secondary Actor: *Player (who play the role of opponent)*
Main Success Scenario:

1. *System* informs *Players* that the Round has started.
2. *System* informs *Players* about the Round Card.
3. *System* informs *Players* that the Scheming Phase is starting.
4. *Players* draw cards until their hand contains 6 cards.
5. *System* executes the Scheming Phase.
6. *System* informs *Players* that the Scheming Phase is over.
7. *System* informs *Players* that Stealing Phase is starting.
8. *System* executes the Stealing Phase.

9. *System* informs *Players* that the Stealing Phase is over.
10. *System* informs *Players* that the Round has ended.

Extensions:

4a. If *Player's* character is Doc, *Player* draws cards until his hand contains 7 cards.

5a. If the Round Card contains an end of round bonus, *System* informs *Players* that the Scheming Phase is over and *System* informs *Players* they may keep an amount of Action Cards from their hand for the next Round corresponding to the end of round bonus.

5a.1a. *Player* informs *System* of the Action Card(s) he wants to set aside. Use Case continues at step 6.

5a.1b. *Player* informs *System* that he doesn't want to set any Action Cards aside. Use Case continues at step 6.

SchemingPhase

Use Case: SchemingPhase
Scope: ColtExpress
Level: Subfunction
Primary Actor: *Player*
Secondary Actor: *Player (who play the role of opponent)*
Main Success Scenario:

1. *System* informs *Player* of round card
2. *System* informs *Player* of turn type
   If the turn is a standard turn, a BasicTurn is played. If the turn is switching, a BasicTurn is played, but the order of the players turns is reversed. If the turn is turmoil, a BasicTurn is played, but all players play their turn at the same time
3. *Player* executes their turn which can be of these types:
   - StandardTurn
   - TunnelTurn
   - Speeding-upTurn
   - TurmoilTurn
   - SwitchingTurn
4. Steps 1 and 2 are repeated until there are no more turns on the round card.

BasicTurn

Use Case: BasicTurn
Scope: ColtExpress
Level: Subfunction
Primary Actor: *Player*
Secondary Actor: *Player (who play the role of opponent)*
Main Success Scenario:

1. *System* informs players of possible actions:
   - Draw 3 cards
   - Play a card

   Playing a whiskey is only possible on a Standard Turn type
   - Play a whiskey

2. *Player* informs *System* of the action he would like to take
3. *System* informs *Player* of the new game state.

Extensions:

1a. *Player* chooses to draw 3 cards from his deck
  1a. 1. Cards are added to *Player's* hand. Use case continues at step 2.
1b. *Player* chooses to play a card from his hand
  1b. 1. *System* adds *c*ard to the top of the Action Card Stack. If *Player's* character is Ghost and it is the first turn in the round, card is played face down, otherwise card is played face up. Use case continues at step 2
1c. *Player* chooses to play a normal whiskey flask
  1c. 1. *Player* draws 3 cards from his deck. Use case resumes at extension 1b.
1d. *Player* chooses to play an old whiskey flask
  1d. 1. *Player* chooses 2 cards to play.
  1d. 2. *System* adds card to the Action Card Stack in the order they were played. If *Player's* character is ghost and it is the first turn in the round, the first card is played face down, the second card is played face up, otherwise both cards are played face up. Use Case resumes at step 2.

TunnelTurn

Use Case: TunnelTurn
Scope: ColtExpress
Level: Subfunction
Primary Actor: *Player*
Secondary Actor: *Player (who play the role of opponent)*
Main Success Scenario:

1. *System* informs players of possible actions:
   - Draw 3 cards
   - Play a card
2. *Player* informs *System* of the action he would like to take
3. *System* informs *Player* of the new game state.

Extensions:

1a. *Player* chooses to draw 3 cards
  1a. 1. Cards are added to *Player's* hand. Use case continues at step 2.
1b. *Player* chooses to play a card
  1b. 1. *System* adds card to the top of the Action Card Stack face down. Use continues at step 2.

Speeding-upTurn

Use Case: Speeding-upTurn
Scope: ColtExpress
Level: Subfunction
Primary Actor: *Player*
Secondary Actor: *Player (who play the role of opponent)*
Main Success Scenario:

1. *System* informs players of possible actions:
   - Draw 3 cards
   - Play a card
2. *Player* informs *System* of the action he would like to take
3. *System* informs *Player* of the new game state.

Extensions:

1a. *Player* chooses to draw 3 cards from his deck

1a. 1. Cards are added to *Player's* hand. Use case continues at step 2.

1b. *Player* chooses to play a card from his hand

1b. 1. *System* adds *c*ard to the top of the Action Card Stack. If *Player's* character is Ghost and it is the first turn in the round, card is played face down, otherwise card is played face up. Use case continues at step 2

StealingPhase

Use Case: StealingPhase
Scope: ColtExpress
Level: Subfunction
Primary Actor: *Player*
Secondary Actor: *Player (who play the role of opponent)*
Main Success Scenario:

1. *System* turns the card on top of the Action Card Stack.
2. *Player* that the Action Card belongs to performs one of the following depending on the Action Card:
   - NoInteractionActionCard
   - InteractionActionCard
3. *System* informs *Player* the action card was returned to his deck.
   Step 1 and 2 are repeated until there is no more cards in the Action Card Stack
4. *System* executes the EndOfRoundEvent.
5. *System* informs all *Players* about the new game state.

NoInteractionActionCard

Use Case: NoInteractionActionCard
Scope: ColtExpress
Level: Subfunction
Primary Actor: *Player*
Secondary Actor: *Player (who play the role of opponent)*
Main Success Scenario:

1. *System* informs the *Players* about which no interaction action card was played. Possible no interaction action cards are:
   - Move
   - Change floor
   - Ride Action
   - Rob
2. *System* informs the *Player* about the choice that needs to be made.

- Move (data: the possible positions where the Bandit can move to)
- Ride Action (data: the possible positions where the Bandit can ride to)
- Rob (data: the loot present in the car the *Player*'s Bandit is situated in)

3. *Player* communicates choice to the *System.*
   - Move (data: the position where *Player* decided to move his Bandit)
   - Ride Action (data: the position where the Bandit jumped off the horse)
   - Rob (data: Which loot will be taken)
4. *System* informs *Players* of the new game state.

Extensions:

2a. *System* is handling the change floor no interaction action card.

2a.1. *System* changes the floor of the Bandit. Use Case continues at step 4.

2b. *System* informs *Player* there is no remaining loot in his Car. Use Case continues at step 4.

2c. *System* informs *Player* there is no horse at the level of his Car. Use Case continues at step 4.

4a. *Player* chooses to ride to the inside of the Stagecoach.

4a.1. S*ystem* informs *Player* about choices to be made.

4a.1.a. *Player* informs *System* about his chosen hostage. Use Case continues at step 4.

4a.1.b. *System* informs *Player* he already has a hostage. Use Case continues at step 4.

4b. *System* is handling the new position of the Bandit.

4b.1. *System* informs other *Players* which adjacent position the affected *Playe*r moved his Bandit to.

4b.2. If *Player* moved his Bandit in a forbidden position (in a car with the Marshal or on a roof with the Angry shotgun), *System* moves Bandit to an appropriate position and *System* adds a bullet to *Player's* deck. Use Case continues at step 4.

InteractionActionCard

UseCase: InteractionActionCard
Scope: ColtExpress
Level: Subfunction
Primary Actor: *Player*
Secondary Actors: *Player* (who play the role of opponent)
Main Success Scenario:

1. *System* informs the *Players* about which single interaction action card was played. Possible interaction action cards are:
   - Shoot
   - Punch
   - Marshal
2. *System* informs the *Player* about the choice that needs to be made.
   - Shoot (data: the possible opponent *Players* the *Player* can shoot)

- Punch (data: the possible opponent *Players* the *Player* can punch, the positions the opponent *Player* can be moved to, and the possible loot the opponent *Player* drops)
- Marshal (data: the possible positions the *Player* can move the Marshal to)

3. *Player* communicates choice to the *System*
    - Shoot (data: the opponent *Player* the *Player* chooses to shoot and the Bullet Card that *Player* places on top of the opponent *Player*'s deck)
    - Punch (data: the opponent player the *Player* chooses to punch, the position the opponent *Player* is moved to, and the loot the opponent *Player* drops)
    - Marshal (data: the position the *Player* chose to move the Marshal to)
4. *System* informs *Players* of the new game state.

Extensions:

3a. If *Player* has to choose between multiple Bandits when shooting or punching, *System* informs *Player* that he cannot choose the opponent *Player* who's Bandit is Belle as his victim. Use Case continues at step 3.

3b. If there is no opponent *Player* in the range of *Player*'s Bandit, System informs *Player* cannot shoot. Use Case continues at step 4.

3c. If there is no opponent *Player* in the same position than *Player*'s Bandit, System informs *Player* cannot punch. Use Case continues at step 4.

3d. If current *Player*'s Bandit is Tuco and the Action Card played is Shoot, *System* informs *Player he* can shoot an opponent *Player* on the other floor of the *Player*'s car. Use Case continues at step 3.

4a. System is handling Current *Player*'s Bandit.

4a.1a. If *Player*'s Bandit is Cheyenne, *Player* can take the purse the punched opponent *Player* lost. Use Case continues at step 4.

4a.1b. If *Player*'s Bandit is Django, *System* moves the shot opponent *Player's* Bandit to the same floor of an adjacent Car farthest from *Player*. Use Case continues at step 4.

EndOfRoundEvents

Use Case: EndOfRoundEvents
Scope: ColtExpress
Level: Subfunction
Primary Actor: *Player*
Secondary Actors: *Player (who play the role of opponent)*
Main Success Scenario:

1. *System* informs *Players* about which end of round event is happening. Possible end of round events:
    - Braking (data: each Bandit on a roof moves to the roof of the adjacent car closest to the locomotive)
    - Take It All (data: Place the second strongbox token in the Marshal's car.)
    - Swivel Arm (data: each Bandit on a roof moves to the roof of the caboose)
    - Angry Marshal (data: The Marshal shoots the Bandits who are on the roof of his car. Each Bandit on the roof of the Marshal's car takes 1 Neutral Bullet Card and places it facedown on his deck.)

- Panting horses (data: remove the corresponding amount of Horse(s) who are the closest to the caboose)
- A shot of Whisky for the Marshal
- Higher Speed (data: Move all Bandits who are on the roof of the train one Car towards the back of the train)
- The Shotgun's rage (data: All the Bandits who are standing in the Stagecoach or on its roof or in the Car in front of the Stagecoach or on its roof receive a Neutral Bullet Card on the top of their deck)
- Sharing the loot
- Passenger's rebellion (data: each Bandit inside a car takes 1 Neutral Bullet Card and places it face down on the deck)
- Marshal's revenge (data: Each Bandit on the roof of the Marshal's car places his least valuable loot on the roof of his car)
- Pickpocketing (data: Each Bandit with no other Bandits in his space may take 1 purse token in his space)
- Hostage Conductor (data: Each Bandit inside or on the roof of the locomotive takes 1 $250 purse token from outside the game)
- Escape (data: Players on the train lose the game)
- Mortal Bullet (data: Each Bandit loses $150 for each Bullet received during this Round)

2. *System* informs *Players* of new game state.

Extensions:

2a. End of round event was a shot of Whiskey for the Marshal.

Step 2a.1 is skipped if there is no Whiskey Flask in the Marshal's car. Use Case continues at step 2.

2a.1. System removes the Whiskey flask in the Marshal's car

2a.1a. If it is a classic Whiskey Flask, the *System* moves the Marshal to the adjacent Car, toward the back of the train. The *System* places 1 Neutral Bullet Card on the top of the deck of each *Player* inside the car the Marshal enters. Use Case continues at step 2.

2a.1b. If it is an old Whiskey Flask, the *System* moves the Marshal through the Cars until he reaches the caboose. Each time the Marshal meets a Bandit during his movement, the *System* moves the Bandit to the roof and places a Neutral Bullet Card on the top of the *Player's deck*. Use Case continues at step 2.

2b. End of round event was Sharing the loot

2b.1 End of the game

2b.1a If several Bandits are on the same Car and at the same level, and one of them owns a Strongbox, *System* shares its value between all the present Bandits. If needed, the result is rounded down. Use Case continues at step 2.

2b.1b System does nothing. Use Case continues at step 2.