# 1.0 Motivation

For our project, we wanted to create a familiar but fresh mobile game for people to have fun, destress, and connect with others. We also wanted to gain more experience with game development. Our game makes sense in a mobile factor because it allows the player to conveniently play anytime, anywhere, alone or with friends. People usually always carry their phones, so they do not need to make special preparations to play our game.

# 1.1 Description

Jailbreak Pong has a nostalgic appeal for anyone who has played the arcade classics Pong, Breakout, and Battleship. However, to make the game more strategic, party-friendly, and motivating for the player to win, a few twists have been added to the fusion.

Much like Pong, Jailbreak Pong involves two players using paddles to relay a ball across a shared board. However, with Jailbreak Pong, instead of losing when you fail to relay a ball back to the opponent, when the ball passes behind your paddle, it has a chance of destroying your prisons, freeing the opponent’s citizens. Please see “Split-screen Mode” Mockup in Section 5.0.

Each player is a colony, and some of their citizens are trapped in the opponent’s colony. Each player’s objective is to use the relayed ball to break into the opponent’s prisons to free their own citizens, and to set up their own prisons in a way that is hard for the opponent to break into. When a player has successfully destroyed all the opponent’s prisons, they win.

# 1.2 Game Setup

During the pre-game setup, players can create fake prisons that trap the ball, and build brick barriers to prevent prisons from being hit. Like Breakout, the bricks are only able to withstand a certain number of impacts. When destroyed, they may release powerups.

The size of each prison can also be varied so that the player can have smaller prisons, but more of them; OR have bigger prisons, but less of them.

# 1.3 Modes

Please see Section 5.0 for the mockups of the different game modes. In split-screen mode, the two players can see each other’s setup, and do not have to be physically together to play.

In party mode, each player (of up to four people, two teams) will only see their own setup on their screen. With the larger screen estate, players will have more room to setup their prisons and barriers, making the game also last longer. *We may not implement this mode depending on time constraints.*

In single-player mode, which they can play offline, the game essentially becomes the arcade classic Breakout, but with the prisons surrounded by bricks. In this mode, the player has a set number of lives, and loses one if they fail to relay the ball back up, towards the bricks.

**2.0 Functional Properties**

1. User Accounts - The application syncs the two User screens during gameplay

* Player
* Opponent

1. Paddle - Used by Players to deflect the ball

* Upgraded paddles - Paddles increase in size when the appropriate powerup is picked up

1. Prisons - Holds opponent’s citizens, who need to be freed.

* Fake Prisons - These Prisons do not free any prisoners when they’re destroyed. Instead, they trap and return the ball to the Player’s paddle.
* Real Prisons - These Prisons free the opponent’s citizens when destroyed.

1. Ball - Relayed to destroy Bricks and Prisons

* Upgraded ball - The Ball’s speed is increased.

1. Bricks - Obstacles to protect Player Prisons

* Pure blocking bricks - Their only function is to protect Prisons. They are destroyed when hit a certain number of times.
* Powerup releasing bricks - They release powerups when they are destroyed.

1. Power-ups - Provide upgrades when picked up by Paddles

* Randomly generated - Generated randomly as the game goes on.
* Stored in Bricks - These are stored in Bricks and released when the Brick is destroyed.
* Different Types
  + Restore Player Prison
  + Create a Fake Prison
  + Increase player paddle size
  + Decrease opponent paddle size
  + Slow down the opponent paddle

1. Board

* Timer - A simple countdown timer. A game would last for a custom amount of time.
* Scoreboard - Shows the freed prisoner count for both players

1. Home screen - A welcome screen that allows users to connect and start the game.

* Single player mode - Allows a Player to play class
* Multiplayer mode - Allows a Player to connect to another player, using WiFi, for a versus style game

# 3.0 User Scenarios

**3.1 Multiplayer Mode**

John and Adam just finished class and have some free time. They are looking for a simple yet engaging game to play against each other, so they decide to play the multiplayer mode. In the game, John uses his paddle to hit the ball causing it to hit one of Adam’s prisons. Adam’s prison gets destroyed and drops a powerup. John picks up the powerup which causes his paddle size to increase. The game timer then runs out and John wins because he has more undestroyed prisons remaining. The benefit in this scenario is the ability to enjoy the game by playing against another player.

**3.2 Single-Player Mode**

Sarah is on the train heading back from work. She reaches out for her phone to look for something to pass the time. She decides to play the single player mode against the computer because she does not have any company at the moment. She logs into her account in order to use her pre-set placement of the bricks and prisons. While playing, she uses her paddle to hit a brick that the computer had placed causing it to shatter. The ball then bounces back and Sarah uses her paddle to hit the ball directing it towards one of the computer’s prisons. The prison gets destroyed but turns out to be a fake prison that does not increase Sarah’s freed prisoner count.

**4.0 Non-Functional Properties**

1. Performance:

* Performance is one of the most critical non-functional properties for a game.
* To ensure optimal user experience, the game has to be extremely responsive

1. Efficiency:

* For developing games on mobile platforms, it is imperative that the games are efficient and use the least amount of resources like battery and bandwidth.
* For example, we would need efficient communication between the two User phones to ensure smooth performance.

1. Heterogeneity:

* Android games must be heterogeneous and serve a wide variety of phones with different hardware specifications. For instance, the game should look consistent on devices with different sized screens.

1. Scalability:

* With the growing demand of the mobile gaming industry, scalability is always a major factor for development decision making.
* Games have a tendency to go viral, resulting in exponential increases in user base. Therefore, they have to be able to scale to meet increasing network traffic.

# **5.0 Mockups & Inspiration**

