# Aa bail mujhay maar!

In this exciting bull fighting game, the matador has been placed in a rink with an energized bull who is determined to make the matador's life a living chaos. To survive, the matador has to finish the bull before it finishes him. The matador will be provided with knives to help him do so. If the bull hits the matador, he loses his lives until he ultimately loses.

The rules of this game are:

- The user will begin at the bull fighting arena with 3 lives and can move left, right, and jump up.
- As soon as the game starts, a timer will begin the countdown for the game for a certain amount of time.
- A bull will be shown on the screen with a certain number of lives and will be running across the screen from one end to another trying to hit the user.
- The user will have to dodge the bull by jumping over it.
- To win the game, the user will be provided with a limited number of knives which they can throw in the direction of the bull to reduce its lives.
- The timer, lives of the bull and user, and the number of knives would be shown above on the screen.
- If the bull hits the user, the user's lives would decrease by 1.
- If the knife hits the bull, the bull's lives would decrease by 1.
- A crate/ box will be displayed at one end of the screen over which the user can jump and avoid being hit by the bull.
  - However, the bull can not hit this crate or harm the user when they are on the crate.
  - The user can not harm the bull from the crate as any knives thrown from the crate would not end up hitting the bull.

- Throughout the game, the bull can be presented with an object that it can consume which would increment its lives by 1.

- To win the game, the user needs to reduce the bull's lives to 0 before the timer ends.

- The game is lost if the user runs out of lives, or the timer ends before the bull's lives finish.

UML is attatched below:

## Timer
- -seconds: int
- +update_seconds() void
- +display_seconds() void
- +drawobject() void

## Game Screen
- #width: int
- #height: int
- +init() bool
- +loadMedia() void
- +run(): void
- +close() void
- +loadTexture(): SDL Texture

## Bull Fighting
- #gRenderer: SDL Renderer
- #assets: SDL Texture
- +draw_obj() void
- +create_obj() void
- +delete_obj() void

## Crate
- -SDL_Texture: assets
- -SDL_Renderer: gRenderer
- +draw() void

## Music
- -x_loc: int
- -music_update:bool
- -y_loc: int
- +update_lives() void
- +draw()void
- +Music(SDL Render, SDL Texture)

## Drink (to increase bull's life)
- -SDL_Texture: assets
- -SDL_Renderer: gRenderer
- +draw() void
- +drop_object() void

## Bull Lives
- -lives_left int
- +update_lives() void

## Bull
- -loc_x; int
- -loc_y; int
- -direction: char
- +draw_obj() void
- +collision_obj() void
- +calculate_life() void

## Matador
- -loc_x; int
- -loc_y; int
- -direction: char
- +draw_obj() void
- +collision_obj() void
- +calculate_life() void
- +select_knife() void

## Matador Lives
- -lives_left int
- +update_lives() void

## Knives
- -x_loc: int
- -y_loc: int
- -SDL Renderer
- -SDL Texture
- +update_knives() void
- +draw() void

## game state
- -x_loc: int
- -lives_left: ibool
- -y_loc: int
- +update_lives() void
- -is_selected() bool= false
- +State(SDL Render, SDL Texture)

## Knife A
- -knives_left: int
- -x_loc: int
- -y_loc: int
- +update_knives() void
- -is_selected() bool= false
- +throw_kinfe() void
- +KnifeA(SDL Render, SDL Texture)

## Knife B
- -knives_left: int
- -x_loc: int
- -y_loc: int
- +update_knives() void
- -is_selected() bool= false
- +throw_kinfe() void
- +KnifeB(SDL Render, SDL Texture)

## Knife C
- -knives_left: int
- -x_loc: int
- -y_loc: int
- +update_knives() void
- -is_selected() bool= false
- +KnifeC(SDL Render, SDL Texture)
- +throw_kinfe() void