

San Francisco State University
CSC 667 Spring 2017

Term Project
Milestone 2: Web Application Entity Design

GitHub Repository:

https://github.com/SFSU-CSC-667/term-project-spring-2017-ivm_

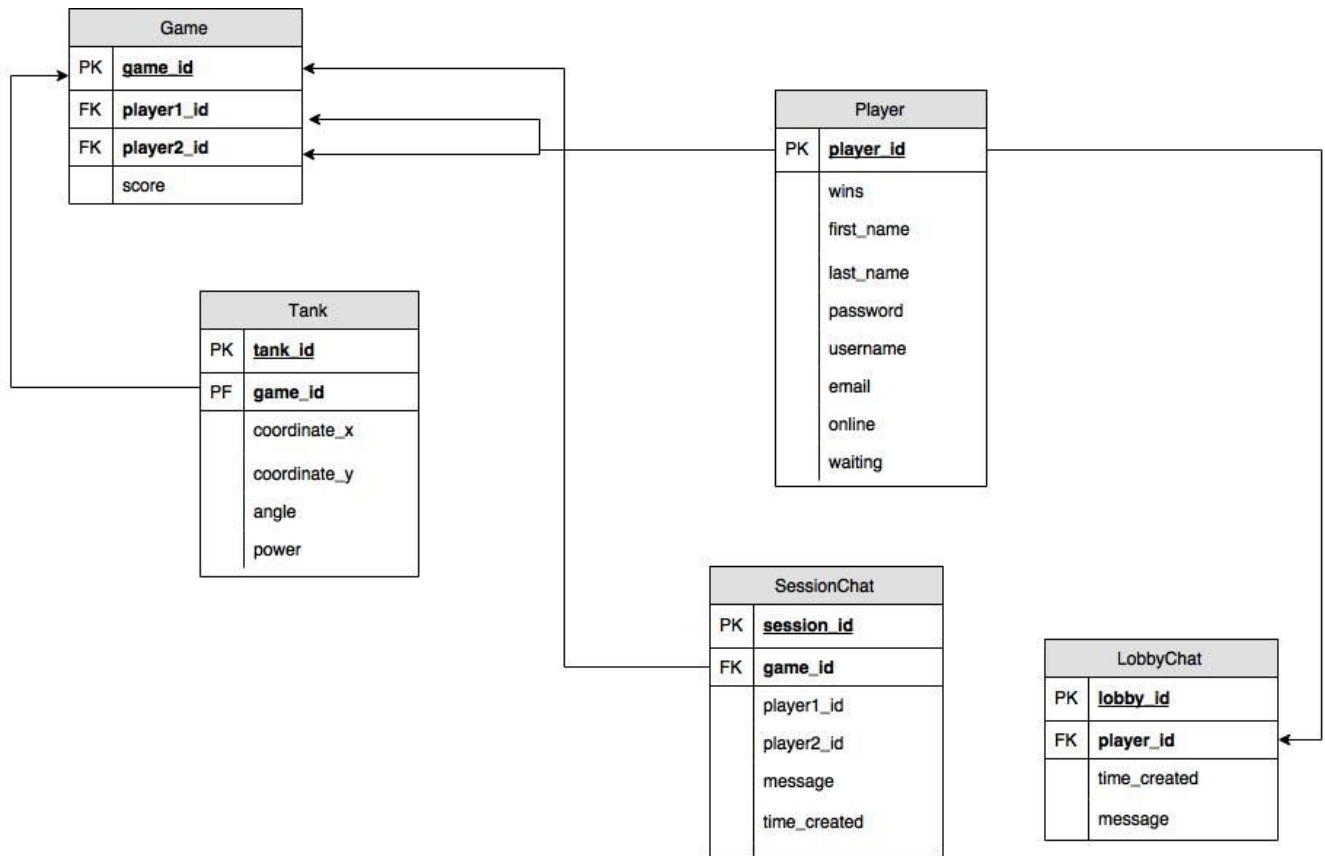
Team Members:

Alex Aichinger
Maroun Abi Ramia
Vivian Lee
Ivan Yu

Table of Contents

I.	Entity Diagram.....	3
II.	Entity Descriptions	
	• Game.....	4
	• Tank.....	4
	• Player.....	5
	• SessionChat.....	5
	• LobbyChat.....	6

I. Entity Diagram



II. Entity Descriptions

- Game

Game	
PK	<u>game_id</u>
FK	player1_id
FK	player2_id
	score

The Game table contains a unique identifier, `game_id`. The `player1_id` and `player2_id` are foreign keys that reference the Player table to show which two players are involved in each game. The Game table also keeps track of the score for each instance of a game. Since the only primary key of the Game table records is the `game_id`, multiple records in this table can have the same `player1_id` and `player2_id`. This table implementation allows two players to play with each other on another game on a different window or tab.

- Tank

Tank	
PK	<u>tank_id</u>
PF	game_id
	coordinate_x
	coordinate_y
	angle
	power

The Tank table has a one-to-one relationship with the Game table. For every game, one record of the Tank table is created and used. This record will have the information on the tank (the coordinates, angle and power) of the player who took the previous turn. Therefore, after at the end of each turn, this single tank record will be updated so that it has the information of the tank of the user who just ended the turn. During the next turn, the information contained in this tank record can be used by the player taking the turn, so that player can know the coordinates of the opposing tank on the screen, its angle, and power. For instance, the `coordinate_x` and `coordinate_y` are respectively the horizontal and vertical positions of the opposing player's tank after the end of the turn. The angle and power of the opposing player's tank can be used to animate the cannon ball launching in the current player's screen, after the opposing player's turn has ended.

- Player

Player	
PK	<u>player_id</u>
	wins
	first_name
	last_name
	password
	username
	email
	online
	waiting

The Player table stores each registered user's personal information, including details used during the Registration process, such as email. It also tracks wins, which is the total number of games that user has won since their registration. The online attribute indicates whether the player is currently signed-in. The waiting attribute indicates if the player has pressed the "Play" button and is waiting to join a game with a random player.

- SessionChat

SessionChat	
PK	<u>session_id</u>
FK	<u>game_id</u>
	player1_id
	player2_id
	message
	time_created

The SessionChat table will be used to store player messages when communicating during a game. The game_id is a primary and foreign key and this is how we will be able to query the chats associated between each user.

- LobbyChat

LobbyChat	
PK	<u>lobby_id</u>
FK	player_id
	time_created
	message

The LobbyChat table will be used to store all messages sent by players on the chat-room available for all online players. The table only has the lobby_id as its primary key, and this primary key makes all messages sent in the chat unique. This implementation allows players to send multiple messages, and the table will store each of these messages.