FootbagPix

The idea is to create a new version of the 80's game Footbag – California Games.
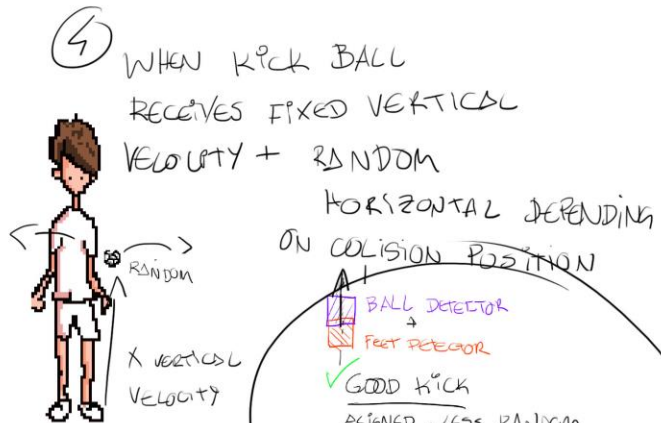
https://www.youtube.com/watch?v=_8Y86STNcd8

We also create a small "diagram" with some basics ideia for the functionality.

①

BALL

LEFT FEET

RIGHT FEET

②

GOOD KICK ONLY WHEN RED AREAS COLLIDES "
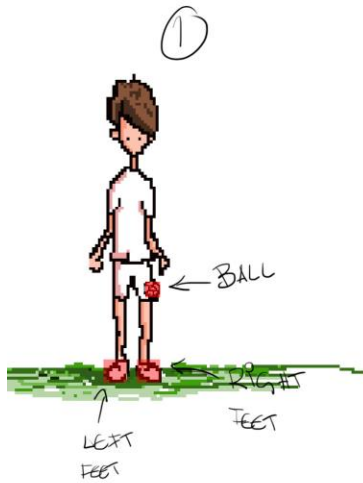
BALL:
HORIZONTAL Velocity = 0;
VERTICAL Velocity = 0;

③

④ WHEN KICK BALL RECEIVES FIXED VERTICAL VELOCITY + RANDOM HORIZONTAL DEPENDING ON COLISION POZITION

RANDOM

X VERTICAL VELOCITY

BALL DETECTOR
FEET DETECTOR
✓ GOOD KICK
ALIGNED, LESS RANDOM

X BAD KICK
BALL GOES MORE TO RIGHT

⑤ MOVES LEFT OR RIGHT AT FIX SPEED

⑥ PERFORM TRICKS TO INCREASE SCORE
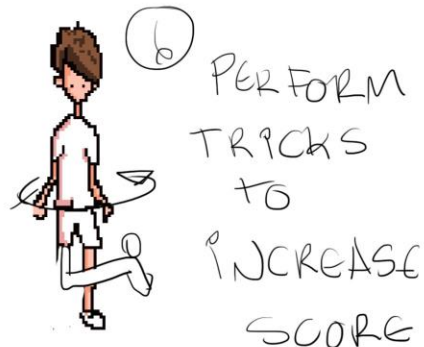
Game Design Document

Player:

Properties:
position (x value),
TurnBack (player is back)
isKicking(?),
kickingType (kicking with left or right foot, front or back, or with heel or knee(not sure), none - It will be used to perform the animation on the sprite)
kickableArea (left foot, right foot, left/right knee, left/right heel, head)

Move Right: Move player for fixed pixels per second when pressed right arrow key
Move Left: Move player for fixed pixels per second when pressed left arrow key
Turn: Turn player back/front (depending on actual position) when pressed down arrow key
Kick: Makes player perform a kick try when player press space. This should verify the actual ball position, and check what should be the closest try kick (left or right foot, knee, heel or head)

Ball:
Properties:
Position (x, y),
Velocity (x,y)
Gravity (game variable?)

When player press kick, checks if ball collides with some "kickable" area on the player (right foot, left foot, knees, heel or head). If yes, velocity Y receives a fixed positive value (start goes up). Also it is checked how centered was the kick, and then the velocity on X axis is set. Also the score increase, based on the *InSeriesTricks* (how many good kicks in serie were performed before).
If ball does not match with any kickable area nothing happen.

Score:
Properties:
ScoreValue,
InSeriesTricks (Array that store tricks without missing kick to multiply points, e.g. "triple kill, multiple hits etc"). If ball hits on the floor (position Y < ground), *InSeriesTricks* reset.

Timer:
Value: timer countdown

Use case diagram

Below you can find the use case diagram, which shows the different windows and the user's possible interactions with the system.
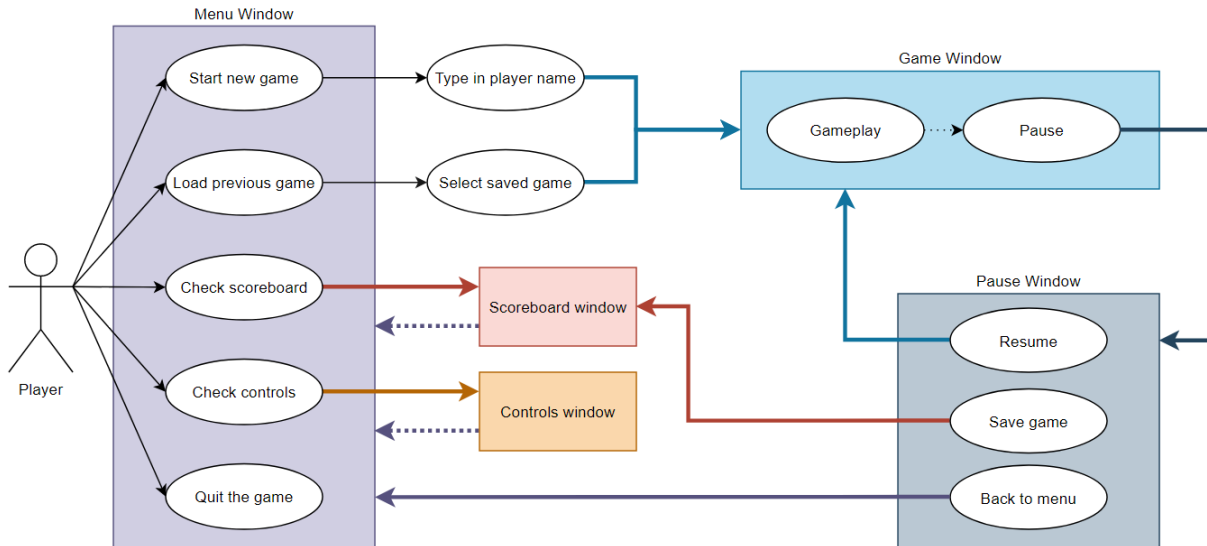
Wireframes

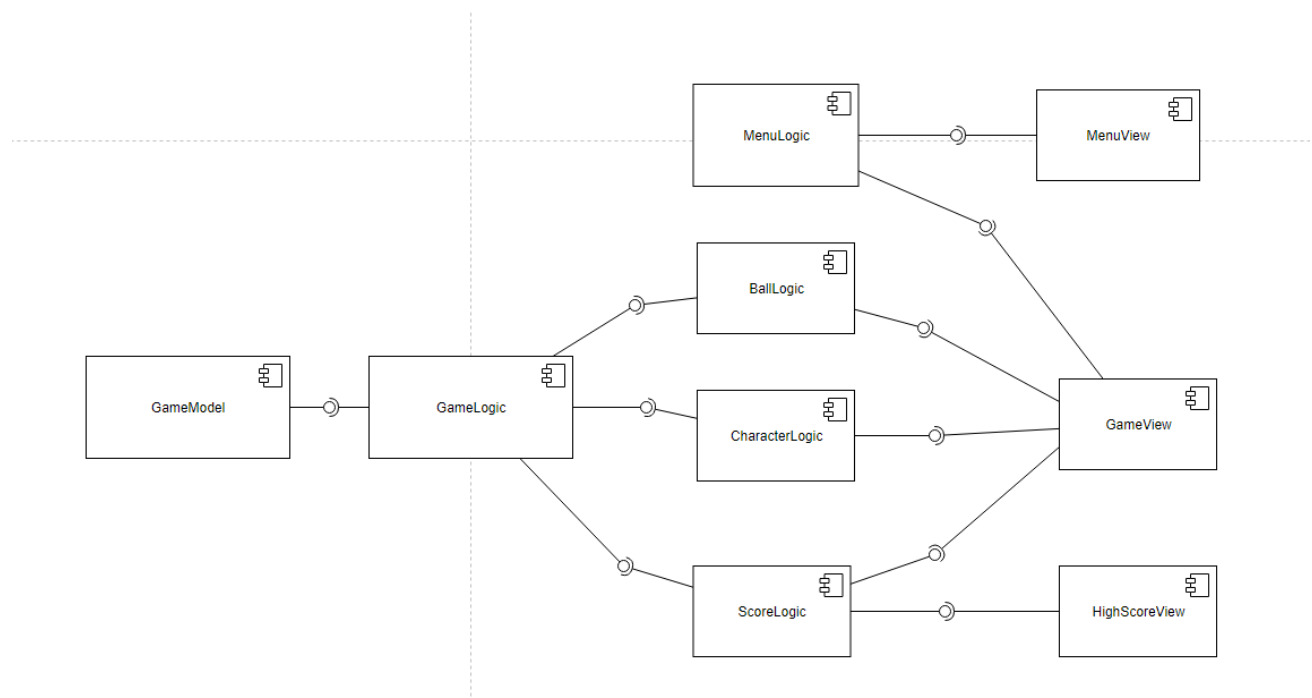**Main menu**: The planned main menu is a simple window with the title of the game, and a few different buttons.
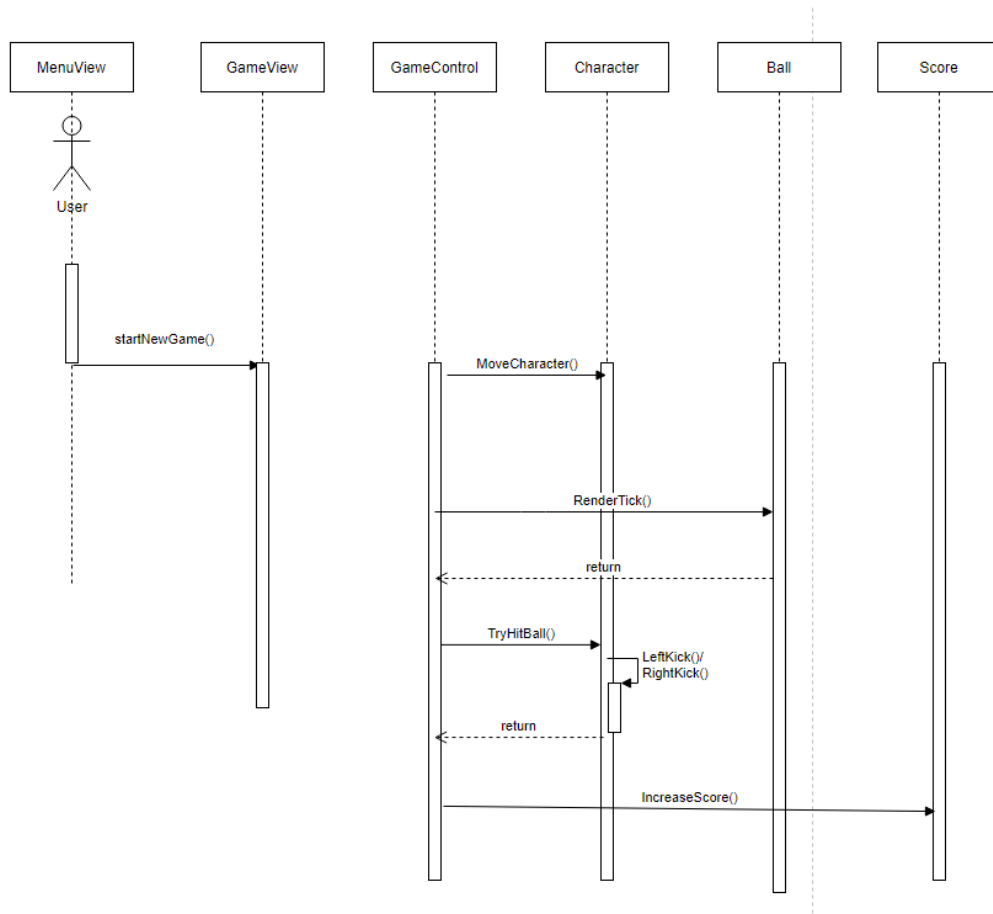
Ingame: During gameplay, we plan to use a clear and simplistic UI, so that the player doesn't get distracted from the gameplay. At the top of the screen, three different items will be visible, from left to right: The current player's name, the current score, and the time elapsed.

Component Diagram

Sequence Diagram

MenuView	GameView	GameControl	Character	Ball	Score

User

startNewGame()

MoveCharacter()

RenderTick()

return

TryHitBall()

LeftKick()/
RightKick()

return

IncreaseScore()

Class Diagram

GameItem

+ size : int (?)

+ PositionX : int (?)

+ PositionY : int (?)

+ Move: void

BallModel

+ VelocityX : double

+ VelocityY : double

FootModel

GameModel

+ character : Character

+ ball : Ball

+ playerName : string

+ currentScore : Score

+ elapsedTime : TimeSpan

CharacterModel

+ PositionX : int (?)

+ leftFoot : Foot

+ rightFoot : Foot

+ isTurned : bool

+ Move: void

+ Turn: void

+ Kick: void

CharacterLogic

+ Move() : void

+ TryHitBall() : void

+ KickLeft() : void

+ KickRight() : void

+ Turn() : void

BallLogic

+ SetPosition() : void

+ SetSpeed() : void

GameLogic

+ CheckCollision() : void

+ CheckBallOnFloor() : void

GameControl

+ gameModel: GameModel
+ gameLogic; GameLogic