

Finger Soccer

Flexible football Game Kit (v1.0)

Requires Unity3D 5.1.1+

Supports Android, iOS, WebPlayer, Windows and Mac

Dear Customer,

Thank you so much for purchasing this game kit. Here you can find the most important information on how to use this kit with maximum proficiency. All script and code assets are fully commented, but if you ever needed a hand on a block of code or anything else, feel free to contact me at <http://www.chupamobile.com> . I'll try my best to support you with your questions as soon as possible and till you are most satisfied.

Overview

Finger Soccer is a flexible football game kit that can be played as a single player (player vs AI) or 2-Players (two players on a same device). Players can select and drag on their units to adjust the power and direction of the shoot. There is also plenty of defensive and offensive formation available to play the game with tactics and strategy.

The finger soccer game kit is so flexible that you can use it as a strong base to prototype many other similar games like billiard, Air hockey, Space ball, etc...



The finger soccer kit uses real world physics to simulate and interact with the elements, so players can always expect a smooth and rational response of events in the game.

This game kit features a complete game flow with menu, shop, configuration and game scenes. The game is also ready to be integrated with the IAB of your own to accept all sorts of micro-transactions. You are minutes away from having a complete game that can generate you real money!

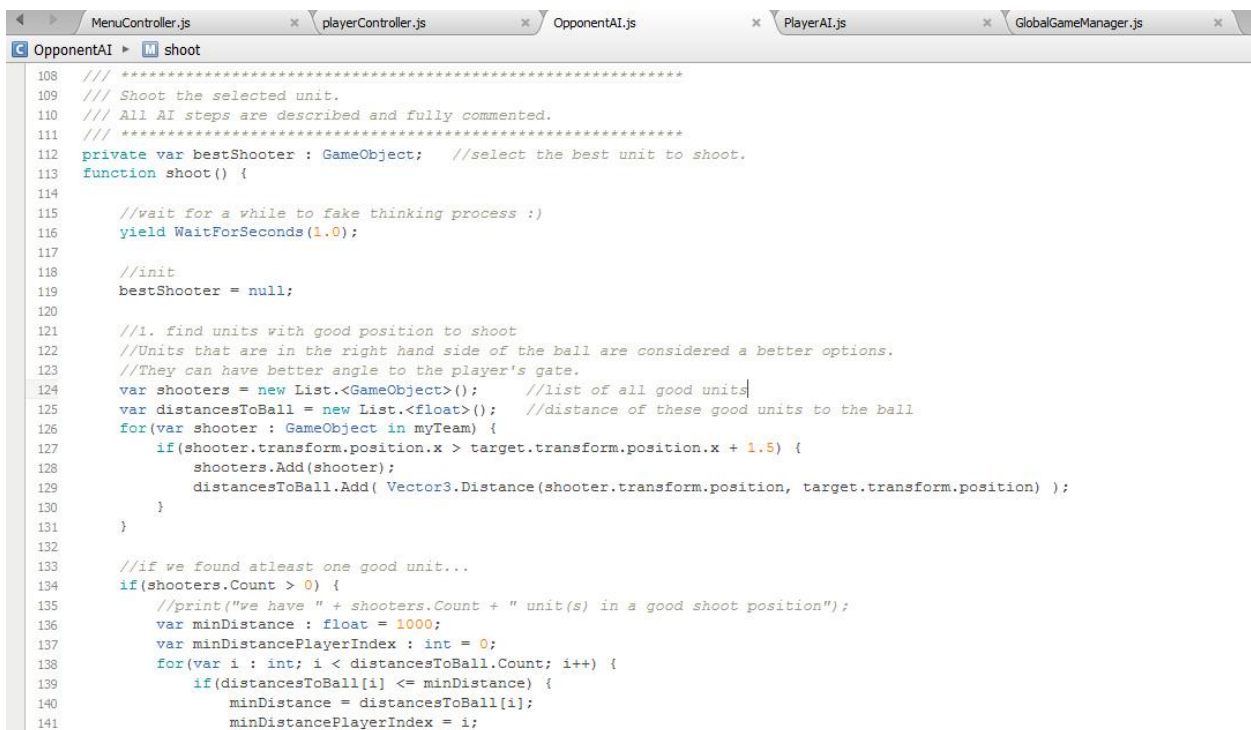
This project accepts both touch and mouse inputs, and thus, can be tested on **Android, iOS, WebPlayer** and **Stand-Alone** platforms simultaneously. This kit works flawlessly on both Unity3d Free and Pro.

!Important:

- When you load the kit for the first time, always add all scenes to the “Scenes in build” list via **file->BuildSettings**, to make sure you experience a smooth transition and game flow.
- The game requires these Tags to work:
 - Opponent
 - Border
 - mouseHelperBegin
 - helperArrow
 - ball
 - PlayerBasketCenter
 - mouseHelperEnd
 - playerGoalTrigger
 - opponentGoalTrigger
 - opponentAI
 - playerAI
 - Player_2
 - gatePost

Make sure to maintain them in your personalized game or replace them with proper equivalent.

- All elements must have a Z position set to -0.5 inside the game at all time. (Freeze Z Position)



```
108  /// *****
109  /// Shoot the selected unit.
110  /// All AI steps are described and fully commented.
111  /// *****
112  private var bestShooter : GameObject; //select the best unit to shoot.
113  function shoot() {
114
115      //wait for a while to fake thinking process :)
116      yield WaitForSeconds(1.0);
117
118      //init
119      bestShooter = null;
120
121      //1. find units with good position to shoot
122      //Units that are in the right hand side of the ball are considered a better options.
123      //They can have better angle to the player's gate.
124      var shooters = new List.<GameObject>(); //list of all good units
125      var distancesToBall = new List.<float>(); //distance of these good units to the ball
126      for(var shooter : GameObject in myTeam) {
127          if(shooter.transform.position.x > target.transform.position.x + 1.5) {
128              shooters.Add(shooter);
129              distancesToBall.Add( Vector3.Distance(shooter.transform.position, target.transform.position) );
130          }
131      }
132
133      //if we found atleast one good unit...
134      if(shooters.Count > 0) {
135          //print("we have " + shooters.Count + " unit(s) in a good shoot position");
136          var minDistance : float = 1000;
137          var minDistancePlayerIndex : int = 0;
138          for(var i : int; i < distancesToBall.Count; i++) {
139              if(distancesToBall[i] <= minDistance) {
140                  minDistance = distancesToBall[i];
141                  minDistancePlayerIndex = i;
```

Game Play

The goal of this game, like every other soccer game is to get the ball into the opponent's gate. The game-manager grants each player their turns to play. When you have the turn, you can select one of your units that you think has the best shoot angle from the ball to the opponent's gate. Then you can tap on the selected unit, hold your finger and drag it away to adjust the power and direction of the shoot.



The underlying processor works similar to billiard or any other similar games. It processes everything with real world physics calculation to deliver a rich and rational response. When testing the game inside editor, you can see all the debug lines that are rendered to help you see what is happening.

The game uses a simple formation system that positions the units on the field. You have full control over this system and can add/remove/modify unlimited number of formations to the game. You can make some of these special formations non-free and give them to your players as rewards or have them buy it from the game's shop with their money.

The game's time is also flexible and can be easily set on any value you wish. You can also provide additional options to your players and let them choose the time of the game on their own.

By default, the game is finished when the time runs out. By that, the player that lands more goals is considered the winner. But you can also configure the game to finish on special occasions like when a player reaches a certain amount of goals.

Introduction to Scripts, Classes and GameObjects

All scripts in this kit are fully commented and have short description about what they do. But here we take a look at some classes used in the kit.

ShopController

This class monitors player money and show them available products (3 new formations by default) that can be purchased for money. This is where you define your new items, formations, flags, etc to make them available for player. You can set the index and price of each item via it's own class "*ShopItemProperties*". When you want to implement your own IAB system, "*MoneyController*" class is where you should do it.

ConfigController

Just before you enter the game, you see a configuration scene that enables you to make some customizations. Here is where you provide those controllers and receive player's preferences. Please note that this class will not do anything on its own. It just get some information's from players and pass them to other managers to interpret. We introduce them later on.

GlobalGameManager

This is the main controller of the game. It manages rounds, turns, player names, game time, game modes, and special events like when a goal happens or when a turn is finished. This class is also responsible to provide UI information via 3d text objects.

LevelHelpers

Level helpers are a set of useful gameObjects that provide insight for AI controllers, to help them have better understanding about game field. They show the goal triggers, and indicate the center of the gates for controllers.

Borders

Borders are a set of predefined game objects that prevent the units and the ball to exit from game scene. The ramp-borders are used to prevent the ball to get stuck when it gets near the borders.

PlayerAI

Player AI manages all things related to player (player-1 or player-2, Human players in general) units, like setting them up for the game and setting their position in the field based on the selected formation.

PlayerController

PlayerController handles all things related to shooting, like selecting the units, dragging, powerHelperArrow rendering, Debug lines and required physics calculations.

OpponentAI

Opponent AI is just like the playerController, but it controls the CPU units. It use a sets of actions to find the best unit to shoot, calculate the best shoot angle, calculate the required power based on the shoot angle and the distance to player's gate, and performs the best shoot possible. You can modify the hardness of the game by tweaking the important parameters you have in this class. For example, you can make the game harder, by modifying the algorithm used to find the "*directionToTarget*" or multiply it with some random number to simulate the stochastic behavior of soccer.

Final Word

If you have any questions, feel free to ask me at <http://www.chupamobile.com/> and I will get back to you as soon as possible.