Действия, которые может выполнять футболист:

1. Бег

Для того, чтобы бежать нужно:

* Выбрать направление
* Определить скорость
* Выполнить действие

Скорость будет определять, какое количество точек пробежит игрок за единицу времени. П1 сам устанавливает какой будет скорость. Чем выше скорость, тем быстрее устает игрок. Количество потерянных единиц энергии необходимо рассчитать позже. Если при движении двух игроков они сталкиваются, то направление игрока с меньшим балансом тела отодвигается по перпендикуляру к движению с большим балансов на количество единиц равное разницы баланса тел игроков, при этом скорость игроков падает в 3 раза, а потом набирается. Если в момент бега игрок (неважно какой команды) пересекается с мячом, то движение игрока останавливается и размещение мяча будет зависеть от параметра «Прием мяча».

Скорость игрока при беге с мячом уменьшается в 1,5 раза.

**Параметр**: количество точек, которые может пробежать игрок за один ход.

**Скрыт:** никогда

1. Удар

Для того, чтобы ударить нужно:

* Определить место в воротах, куда будет производиться удар.
* Выбрать силу удара
* Выполнить действие

Чем выше сила удара, тем меньше точность и наоборот. Чем больше расстояние до ворот, тем больше силы нужно приложить, соответственно падает точность. П1 выбирает действие и указывает в какую точку в воротах он пытается пробить. Затем, определяет силу удара и выполняет действие. В момент удара, мяч может попасть в одного из игроков, и будет некоторым путем решено рекошет это или перехват.

**Параметр:** количество точек, которые удар будет точным. Расчет произвести, когда будет определена единица времени и размеры поля.

**Скрыт:** когда у игрока нету мяча

1. Пас

Для того, чтоб отдать пас нужно:

* Выбрать цель
* Определить силу, с которой будет отдан пас
* Выполнить действие

Сила будет определять, с какой скоростью будет лететь мяч к партнеру. ?Скорость должна быть поднимающей и спадающей?. Чем выше скорость при приеме мяча, тем тяжелее его принять. В момент полета мяча, мяч может быть перехвачен соперником.

При выполнении паса, П1 должен выбрать игрока, который будет отдавать пас. Далее необходимо выбрать направление, по которому будет происходить пас. После этого П1 выставляет силу, с которой мяч будет лететь и принимает действие. В этот момент П1 может использовать действие игрока, которому выполняется пас, например, приближаться к мячу. П2 может предположить кто и куда будет выполнять пас и выполнить действия по прессингу или перехвату или ожидания следующих событий. ? То, куда смотрит каждый игрок соперника, показывается некоторой областью около игрока (сектор градусов 120), чтобы можно было предположить куда будет выполняться действие.?

Пас может быть на ход другому игроку. Тогда П1 не должен указывать кому отдать пас, а должен просто выбрать подпункт действия пас как пас на ход и выбрать произвольную точку на поле, куда откатить мяч, а другим игроком подбежать к мячу.

**Параметр:** количество точек, которые пас пройдет точно. Расчет произвести, когда будет определена единица времени и размеры поля.

**Скрыт:** когда у игрока нету мяча

1. Навес

Для того, чтобы выполнить навес нужно:

* Определить направление мяча
* Выбрать силу передачи или навеса
* Выполнить навес

Навес очень похож на пас, только он может происходить по верху. В таком случае, мяч может быть перехвачен игроком другой команды, только если он находится на определенное количество единиц (рассчитать позже – зависит от роста игрока) от пасующего или целевой точки навеса.

**Параметр:** количество точек, которые навес пройдет точно. Расчет произвести, когда будет определена единица времени и размеры поля.

**Скрыт:** когда у игрока нету мяча

1. Подкат

Для того, чтобы выполнить подкат нужно:

* Определить цель для подката
* Определить момент выполнения подката
* Выполнить подкат

Подкат может помочь отобрать мяч. Подкат можно выполнить как при статическом состоянии так и при беге игрока. П1 задает цель для подката и определяет момент выполнения подката. Это означает, что как только игрок достигнет этого момента, то подкат сразу же выполняется. Момент может быть как координатным, событийным или условным. Когда игрок выполняется подкат, то он проходит дистанцию в 2 раза быстрее, но после завершения подката игрок заканчивает ход. Если в процессе подката, мяч был перехвачен, то рассчитывается точка приема мяча. Подкат будет также выполнятся на программном задании управления игрока.

Параметры, которые есть у футболиста (практически все от 0 – 100):

1. Все выше указанные
2. Точность коротких пасов – количество точек отклонения от цели ((100 – Х)/2).
3. Точность навеса (длинных пасов) – количество точек отклонения от цели (100 – Х)/1.5
4. Сила – количество точек, которые пролетит мяч за единицу времени (Х/2)
5. Скорость – количество точек, которые пробежит игрок за единицу времени (Х/5). Значение на которое делится должны быть рассчитано исходя из скоростей других сущностей.
6. Выносливость – количество единиц спадания энергии за единицу времени (100 – Х)/10. Этот параметр должен быть более точно рассчитан в зависимости от единиц времени.
7. Баланс тела – количество единиц (разница значений столкнувшихся игроков), на которое игрок отклонится в движении при столкновении 2 игроков.
8. Прием мяча – количество единиц, на которое мяч отлетит от игрока при приеме мяча (100 – Х)/10
9. Точность подката – вероятность того, что подкат будет успешным.

Дополнительные параметры:

1. Опыт
2. Уровень

Архитектура приложения

Приложение будет выполнено в вебе. Для этого будет использоваться серверная технология ASP.NET WebForms.

Процесс игры будет происходит следующим образом:

Некоторый игрок создает процесс игры. Этот процесс – это некоторый контейнер, который имеет свои параметры игры, свои условия, свое футбольное поле, игроков и т.д. Использовать это процесс может только 2 игрока. Этот процесс никак не взаимодействует с другим процессом, в котором тоже может существовать игра.

Когда процесс создан, он появляется в некотором списке процессов к которому может быть подключен другой игрок. Когда игрок подключается к процессу, то он исчезает из списка процессов и у него происходит старт.

При создании процесса необходимо настроить его:

* Имя процесса
* Создатель
* Дата создания
* Время хода
* Время матча
* Режим выбора команд
  1. Существующие команды
  2. Произвольное создание команд (в этом случае, каждому дается некоторое количество очок для создания игроков – количество задается при создании процесса)

После того, как два игрока вошли в процесс, он запускается. При этом, каждому из игроков дается возможность выбрать команду или же создать свою. После того, как команды сформированы или выбраны, происходит знакомство команд и реализация тактических схем (в будущем добавляться еще программные процессы для игроков). После этого начинается игра.

Для процесса игры будет некоторая страницы с определенным идентификатором(хешем). Для игрока, который зашел в этот процесс будет создаваться некоторая сессия игрового процесса, которая необходима для того, чтобы восстановить состоянии игрока в случае потери сети или случайного выхода из процесса. Будет также существовать некоторый тайм-аут, после которого, эта сессия будет очищаться. Сессия будет представленная в виде сериализованых данных, которые будут лежать в БД.

При старте игры создается игровое пространство общее для двух игроков и происходит расстановка игроков. Каждый игрок на поле, будет характеризоваться текущим положением, последним выполненным действием и действием, которое хочет совершить. Все эти данные будет представленны в JSON виде. Когда происходит ход, то оба игрока начинают выполнять действия игроков. Каждое действие будет представлено некоторым объектом. Когда игрока сделают некоторое процесс-действие, то они нажимают кнопку закончить ход (если до этого время не вышло) и на сервер поступает сигнал, что некоторый игрок закончил ход и в случае прихода обоих сигналов, процесс-действие начинает выполнятся. Процесс-действие – это некоторый объект, который представляет собой определенную эмуляцию действия за один ход. В нем есть набор действий каждого игрока и по этому набору будет генерироваться некоторый код. Этот код, будет передаваться на клиент будет происходит перерисовка на игровом пространстве. При анимации могут возникать дополнительные события, например, столкновение игроков, перехват мяча, аут и т.д. По этим события будет формироваться некоторый экземпляр действия и туда передаваться необходимые параметры по которым будет определено дальнейшее действие для процесса эмуляции. Обработка этих событий будет происходить прямо на клиенте(возможно нужно будет перенести все на сервер).

Начало работ

1. Начало разработки нового фреймворка для упрощения работы
2. Создание необходимых элементов управления