

# Introduction

This document collects principles for the design of BALTEK.

## General design

Let us decouple the various aspects of the software in order achieved the following objectives:

- Playing against an AI.
- Playing against a remote player.
- Playing either on a desktop computer or a smart phone.
- Changing the visual aspect of the software.

Let us get inspiration from the **Model View Presenter (MVP)**



Let us translate such diagram into a table of events:

Step	View	Presenter	Model
1	sends user event		
2		receives user event	
3		requests model change	
4			updates model
5			send state-change event
6		receives state-change event	
7		request view change	

The previous objectives implies the following cluster of classes:

## Cluster for the state of game (Model)

---

These classes represent:

- The boxes of the field: its geometry, the position for the goals.
- The blue and red players: their forces, their positions, their moves in the current turn.
- The position of the ball.
- The score.
- The usage of the sprint bonus for each team.
- The turn: blue team or red team.
- The remaining action points of the active team.

These classes know:

- How to initialize the state of the game.
- How to clone the state the game.
- How to move the players at their initial positions.
- The possible action of the active team: move of each player, move of the ball, reinitialize the players positions.
- How to note the played moves.

## Cluster for the Presenter

---

These classes manage:

- The overall initialization of the software.
- The coordination between all other clusters.
- The captures of events.

## Cluster for the View

---

These classes manage the drawing. Nothing else.

## Cluster for the AI

---

These classes represent a virtual player.

From a given state of the game, the AI provides moves. An advanced AI might also used the previous

states of the game in order to provides the next moves.

## Cluster for the remote interaction

---

These classes are responsible for:

- Initializing the remote connection.
- Transforming local event into remote event, and vice versa.

## The game phases

---

Hereafter are the phases that are relevant and useful for organizing the software

- 1 match is composed of  $n \geq 1$  rounds
- 1 round begins after each kickoff
- 1 round ends after a goal by one player
- 1 round is composed of  $n \geq 1$  turns
- 1 turn is devoted to a given player
- 1 turn is composed of  $n \geq 0$  moves
- 1 move is the smallest action that a player can decide

## Estimation of the number of possible moves

---

Let us determine a bound for the possible moves:

- 6 boxes, related to the six footballers are selectable.
- 3 kinds of moves are possible:
  - kick
  - run
  - sprint
- For a kick  $(5 \times 5 - 1) = 24$  destination boxes are possible.
- For a run  $(3 \times 3 - 1) = 8$  destination boxes are possible.
- For a sprint  $((5 \times 5 - 1) - (3 \times 3 - 1)) = (24 - 8) = 16$  destination boxes are possible.
- For a move,  $6(24 + 8 + 16) = 6 \times 48 = 288$  moves are possible, or maybe less, but never more.

# Ideas for cutting responsibilities amongst the classes

---

- One entity PlayerBlue
- One entity PlayerRed
- One entity RulesEngine
- The entities Players can query the RulesEngine
- The RulesEngine notifies the Players
- Each Player provides its move, or decision to end its turn to the RulesEngine
- The entity Presenter updates the Screen
- The RulesEngine notifies the Presenter of changes (moves of the Players)
- The Mouse notifies the Presenter, which in turn, updates the active Player
- Kinds of notifications of the RulesEngine towards the Players, and maybe the Presenter:
  - start/end of round
  - start/end of turn
  - start/end of move
  - etc.
- A move can be seen as a triple :
  - Source box: it is a box hosting a footballer from the team of the active Player
  - Kind of move:
    - run of the footballer
    - sprint of the footballer is the "sprint bonus" has not yet been used
    - kick of the ball if the ball is in the box, and if possible dribble can be paid
  - Destination box: either for the footballer (run or sprint) or for the ball (kick)
- The RulesEngine can help the Player for deciding its move using the following sequence:
  - Player requests the RulesEngine for a new Move entity.
  - Player queries the RulesEngine for the selectable source boxes
  - Player selects one item from the selectable sources, and records that choice within the Move entity
  - Player queries the RulesEngine for the selectable kinds of move related to the chosen source
  - Player selects one item from the selectable kinds, and records that choice within the Move entity

- Player queries the RulesEngine for the selectable destinations of move related to the chosen source and kind of move.
- Player selects one item from the selectable destinations, and records that choice within the Move entity
- Player either confirms/commits its move or cancels it.
- Player should also have the ability to unselect a chosen at one of the three steps: select/unselect the source of move; select/unselect the kind of move; select/unselect the destination of the move.
- Hint-1: let us imagine an Artificial Intelligence that queries the RulesEngine, makes its own assessment, makes its exploration of the tree of moves, makes its decision.
- Hint-2: the steps of the construction of the move should be also beneficial for interaction between the Presenter and the (Screen/ Mouse). Either the RulesEngine or the Move can notifies the Presenter of the steps of construction of the move.

## Idea for easily click on smartphone

---

- Only click a box, no attempt to click on the ball or the player inside the box.
- If the box only contains the footballer:
  - the first click selects the footballer for a "run"
  - the second click selects the footballer for a "sprint" if it is possible
  - the third click unselect the footballer.
- If the box only contains the footballer and the ball:
  - the first click selects the ball for a "kick" if it is possible
  - the second click selects the footballer for a "run"
  - the third click selects the footballer for a "sprint" if it is possible
  - the fourth click unselect the footballer.