Visual Planner

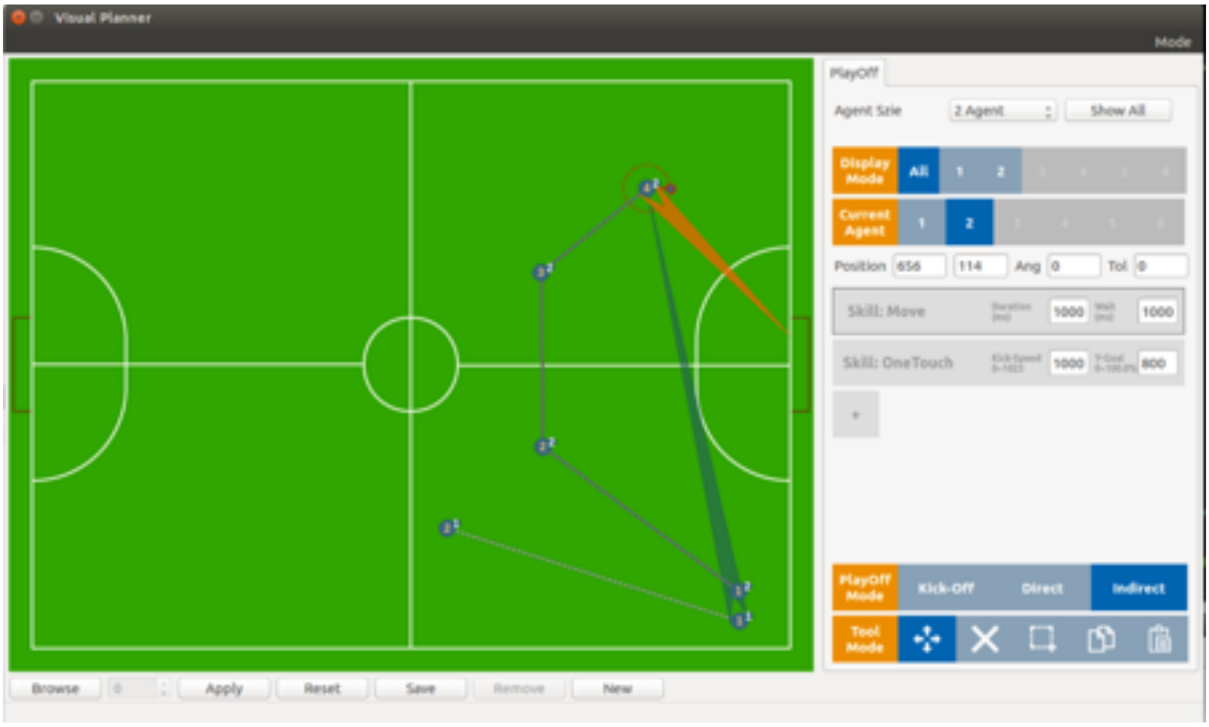
1.Summary

It has been always complex and time consuming for arranging a good plan for offensive roles, Visual Planner is an application for making complex plans so simple, by using graphical user interface.
For example assume you want your agent to send a pass to another agent near opponent goalie area, in old fashion way of planning you would probably use some scripts to limit the possibility among other plans, beside choosing skills and and other critical input of robots. But with help of Visual Planner you can simply do all those above in very shorter timer and lower complexity!

This application has two sides, PlayOff and PlayOn
PlayOff is for those moments that the ball is still untouched like Kick-Off, Direct and Indirect.
PlayOn is for those moments that the ball is moving and there is no fixed position for ball nor robots.

2.PlayOff (shortcut: use Ctrl+m)

Like explained before, PlayOff is for situations that robots' position are fixed and there is a fixed point for ball, In PlayOff it's very important to have plans that can be executed sequentially, so the main focus of Visual Planer in PlayOff mode is to have sequence, and events to make it easier to terminate and proceed each tasks.
Each plan will be categorize according to agent size, PlayOff mode (Kick-Off, Direct and Indirect kicks) and ball position (though it doesn't ask user to declare ball position directly)



Agent Szie: 2 Agent
Position 656 114 Ang 0 Tol 0
Skill: Move Duration (ms) 1000 Wait (ms) 1000
Skill: OneTouch Kick-Speed 0-1023 1000 Y-Goal 0-100.0% 800

2.1.Agent Size

Agent size can be declared in this combo box, the maximum agent size is 6.

2.2.Display Mode

With use of this feature you can choose to display each plan timeline individually, it also affects the choosing algorithm that can be very helpful, specially when there is more than one agent in a same point.

2.3.Current Agent

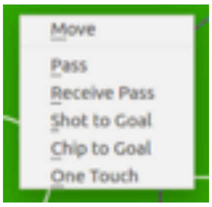
It declares which agent's timeline is being edited, for minimizing the human error, it can only increase to agent size value.

2.4.Agent Geometry

Although you can completely choose agent geometry using mouse, but for making it more accurate you can use this feature and set X, Y and Angle of agent directly, It also allows to have tolerance in the case you are not sure about the desire position and want the AI decides best position for you.

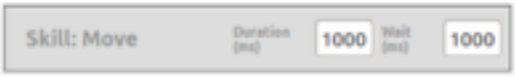
2.5.Skills Toolbar

Each states can have up to three skills, as default the first skill is Move. Available skills are: Move, Pass, Receive Pass, Shot on Goal, Chip to Goal and One touch.



2.5.1.Move

You can choose the duration of going from point A to B, and wait time in milliseconds. Wait time comes handy when you need to optimized your one touch skills or compensate latencies of fast plans.



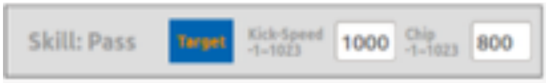
Skill: Move Duration (ms) 1000 Wait (ms) 1000

2.5.2.Pass

Pass skill have three inputs; Target, Kick Speed and Chip.

By clicking in target button you can choose the pass receiver agent, it is completely dynamic and you can move the receiver agent and see the the target updated.

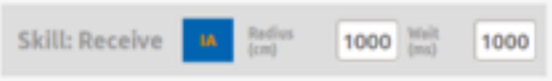
The reason that pass have both kick and chip speed is clear it will change the direct pass to chip pass if there will be obstacles in pass path, but for making it so reliable you can disable each of those by changing value to 0.



Skill: Pass Target Kick-Speed -1~1023 1000 Chip -1~1023 800

2.5.3.Receive Pass

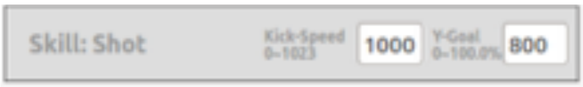
Receive Pass Skill has two input Receive Radius that clears the receiver agent movement limit from its position and wait time which explained. The IA stands for ignore angle, this feature doesn't force agent to look at ball.



Skill: Receive IA Radius (cm) 1000 Wait (ms) 1000

2.5.4.Shot on Goal

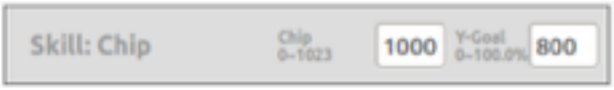
It has two inputs; kick speed and the vertical position of target to shot (0 means bottom and 1000 means top position of opponent goalie)



Skill: Shot Kick-Speed 0~1023 1000 Y-Goal 0~100.0% 800

2.5.5.Chip to Goal

Like Shot on Goal it has two input the only difference the way of kicking.



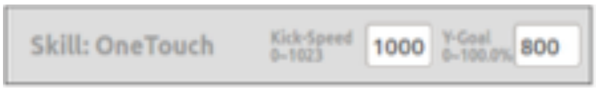
Skill: Chip

Chip 0–1023: 1000

Y-Goal 0–100.0%: 800

2.5.6.One Touch

It’s like Shot on Goal with minor difference it has receive skill in it as well.



Skill: OneTouch

Kick-Speed 0–1023: 1000

Y-Goal 0–100.0%: 800

*note: each skill can be removed by pressing right click on them

2.6.PlayOff mode

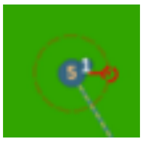
User must choose plan mode here
Options are: Kick-Off, Direct and Indirect

2.7.Panning Tools

For making it more user friendly we have implemented few basic tools, which are:

2.7.1.Place/Move

An agent can be placed simply by pressing left click mouse in field area.
User can move agents by drag and dropping it, also final angle can be set here

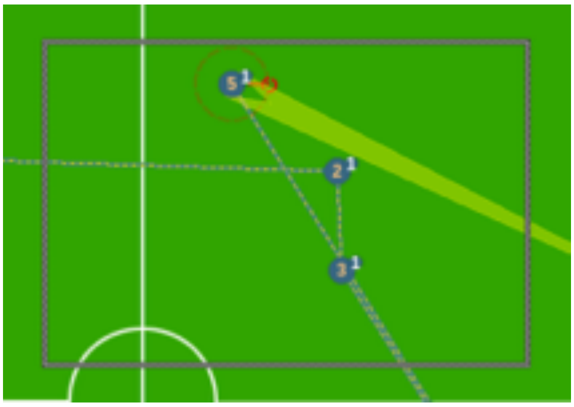


2.7.2.Remove

Both middle mouse button and remove icon can remove undesired agent.

2.7.3.Multi Agent Select

User can select multi agent with rubber band and use other tools, like moving, removing, copying and pasting a group of agents.
For removing a group of agents user need to click inside the rubber band rectangle.
Selected agents can be filtered using display mode option



2.7.4.Copy

User can copy a group of agents from any plans, but before that user need to select them using select tools

2.7.5.Paste

Anything in clipboard can be placed by clicking on paste button

2.8.PlayOff SQL file

For making our saving plan so simple we used SQL, SQL is so simple to use, fast and efficient.
In saved file we have two types of SQL;

2.8.1.Plan Essential Data SQL

For saving essential values like init positions for matching, agent size, PlayOff mode

SQL Structure (just for example)

id	pname	pmode	psize	pball	pagent1	pagent2	pagent3	pagent4	pagent5	pagent6
1000	plan1	1	3	255l34	12l544	65l77	na	na	na	na
1001	plan2	1	3	344l34	34l83	na	12l700	na	na	na
1002	plan3	1	3	12l57	32l456	123l54	na	na	na	na

Id obviously identifies each plan, and help SQL engine to find desired plan faster

Pname is the name of table that we want to save our plan in it, (all agents with all indexes)

Pmode defines PlayOff Mode

```
enum POMODE {
   KICKOFF = 1,
   DIRECT = 2,
   INDIRECT = 3
};
```

Psize defines agent size

Pball is first position for ball, it helps AI to match each plan faster, template is like this (XPosition of ball + “l” + YPosition of ball)

Pagent1~6 are like Pball, and the main use of these points are for matching agents

2.8.2.Sequential Plan SQL

This SQL stores each sequence geometry, tolerance, skills and etc…
Plan Structure:

There is an ID column the rest follows :
Agent1:

r1geo	r1tol	r1skill1	r1skill2	r1skill3	r1ptarget
X+”l”+Y +”l”+Angle	integer	POffSkills	POffSkills	POffSkills	Agent +”l”+Index

Agent2:

r2geo	r2tol	r2skill1	r2skill2	r2skill3	r2ptarget

Agent3:

r3geo	r3tol	r3skill1	r3skill2	r3skill3	r3ptarget

Agent4:

r4geo	r4tol	r4skill1	r4skill2	r4skill3	r4ptarget

Agent5:

r5geo	r5tol	r5skill1	r5skill2	r5skill3	r5ptarget

Agent6:

r6geo	r6tol	r6skill1	r6skill2	r6skill3	r6ptarget

```
enum POffSkills {
    NoSkill = 0,
    PassSkill = 1,
    ReceivePassSkill = 2,
    ShotToGoalSkill = 3,
    ChipToGoalSkill = 4,
    OneTouchSkill = 5,
    MoveSkill = 6
};
```

2.9.Arranging Plan

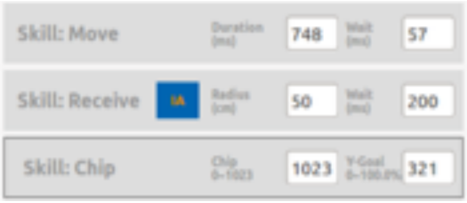
First of all you need to press on NEW button to have an empty project then choose how many agents you want for a plan (you have to so this to enable other options)

Then you have to choose between PlayOff modes which are Kick-Off, Direct and Indirect.

The next thing that really matters is ball position, we don’t declare ball position directly, actually we use skill data to find out where is the ball position, for example it is obvious that the agent with pass skill have to be nearest agent to ball, so we simply find out ball position.

Each states in plan has timing in order to have faster plan with better calibration.

You can have up to 3 skills for each state, it’s better to have move plan in first skill because other skills don't have duration and wait time options.

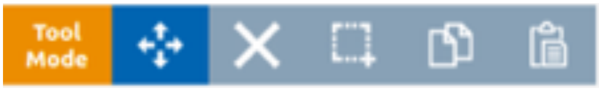


Skill: Move Duration (ms) 748 Wait (ms) 57

Skill: Receive IA Radius (cm) 50 Wait (ms) 200

Skill: Chip Chip 0-1023 1023 Y-Goal 0-100.0% 321

Then by using toolbar you can place and replace agents and arrange them for your plan



Then press on apply button to confirm the changes and hit on save button, use *.db or *.db3 files, and it’s done,

2.10.Encoding the SQL file

For making it faster you have to load all of your plans into generic arrays like QList or Vector

A simple example:

First you need to have a structure for each robots in each states

We use this structure:

```
struct playOffRobot {
    Vector2D pos;
    AngleDeg angle;
    double tolerance;
    int targetIndex;
    int targetAgent;
    POffSkills skill[3];
    int skillData[3][2];
    int skillSize; };
```

Then you have to have a list of this structure. we used QList, and finally there should be up to current agent size copy of that QList, and we added few variables to match best plan in SQL file.

```
struct planStruct {
   QList<playOffRobot> AgentPlan[6];
   POMODE planMode;
   int agentSize;
   POInitPos initPos;
};
```

3.PlayOn (shortcut: ctrl+N)
This part explained in our TDP

4.System requirement
Qt5 <
*It is recommended to run in Qt5.4.2
geom library
Code can be found in https://github.com/hamidrezakks/ssl-visual-planner