Sprite

* int pos[];
* int size[];
* \*SDL\_Surface Surface;
* Sprite();
* int getXY();
* int getSurface();
* void initSurface(int size[], int color[]);
* void drawSurface(SDL\_Surface dest, SDL\_Surface source, int xy[]);
* bool checkCollision(Sprite S1, Sprite S2);

Timer

* float lastTime;
* float currentTime;
* Timer();
* Float getDT();

Controller

* Player P;
* \*data type\* inputSource;
* double state[];
* Controller(Player P, \*data type\* inputSource);
* void movePlayer();

FrictionZone: Sprite

* int friction;
* FrictionZone(int friction);
* void slowSprite(Sprite S1);

Goals: Sprite

* bool team;
* int score;
* Goals(bool team);
* Void increaseScore(bool hasBall)

Ball: Sprite

* double speed;
* Player captivity;
* Ball(double speed);
* void moveFree();
* void getCaptured(Player capture);
* void moveCapture();
* void getShot(int theta);

Player: Sprite

* double speed;
* bool hasBall;
* bool team;
* Player (double speed, bool team);
* void move(double action[], double dt);
* void shootBall(Ball b);
* void captureBall();

Map

* \*SDL\_Surface;
* int color[];
* Map(int size[], int color[]);
* void updateMap(Sprite allSprites[]);
* void updateSprites(Ball b, Player p[], Goal g[], FrictionZone f[]);
* \*SDL\_Surface getMap();