TurtleField Quick Reference

11-JUN-2017, Y. Hayakawa (CITE, Tohoku Univ.)

Turtle Graphics

	C functions (all names in capital letters)	
connect to server	int CON(char *hostname)	TF uses TCP port 2222
clear screen	void CLR(void)	
home position	void HOME(void)	origin is the center of square
pen up	void PU(void)	
pen down	void PD(void)	
step forward	void FD(float step)	
step back	void BK(float step)	unit of length: pixel
jump	void JUMP(float x, float y)	
relative jump	void RJUMP(float dx, float dy)	
turn left	void LT(float angle)	
turn right	void RT(float angle)	unit of angle: degree
print string	void SAY(char *string)	
line mode	void LINE(void)	
brush mode	void BRUSH(void)	
filling mode	void FILL(void)	max # of vertices in filling mode: 6
display card	void CARD(int type, float x, float y)	type: 1..53
clear card	void CLRCARD(void)	
query curr. position	void Q_POS(float *x, float *y)	
query heading direction	void Q_DIR(float *angle)	
pen color	void COL(float r, float g, float b)	0.0 < r, g, b < 1.0
background color	void BGC(float r, float g, float b)	

Set callback functions for event handling

```
SET_HIT_BY_BULLET_HANDLER(func)
SET_RUN_INTO_TURTLE_HANDLER(func)
SET_RUN_INTO_DONUT_HANDLER(func)
SET_RUN_INTO_STONE_HANDLER(func)
SET_RUN_INTO_WALL_HANDLER(func)
SET_FOUND_COIN_HANDLER(func)
SET_DETECTED_BY_FINDER_HANDLER(func)
SET_DETECTED_BY_RADAR_HANDLER(func)
SET_GOT_MESSAGE_HANDLER(func)
```

Callback functions

```
(void) callback_func(unsigned int time)
```

time code is passed by caller

Scoring

initial score of each: 100
point to loose for each shoot: -1
hitting other turtle: +10
hit by bullet:-30
hitting donut within short distance: +10
when score < 0, disconnected

Regulations/limitations in battle (robot) mode

step size of FD & BK: up to 2 pixels
turning angle: up to 3 degrees
field is surrounded by rigid walls

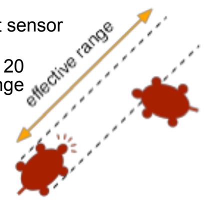
only single bullet can exist at a moment in the field for each turtle

Battles & robot simulations

	C functions	
switch to battle mode	void BMODE(void)	
switch to graphics mode	void GMODE(void)	
shoot a bullet	void FIRE(void)	
set nickname	void NM(char *name)	
set my team number	void TM(int team_id)	0<=team_id<=4
query number of objects	void Q_NT(int *n)	returns # of turtles and donuts
query to finder	void Q_FINDER(int *result)	returns code →
set range of finder	void RF(float range)	
query to angular radar	void Q_RADAR(float *angle)	returns angle →
query to sonar	void Q_SONAR(int *left, int *front, int *right)	↘
query my current score	void Q_SCORE(int *score)	
query team id of turtle met just before	void Q_TM(void)	use this call **after** Q_FINDER or Q_RADAR
put a robot turtle in the field	void ROBOT(void)	
scatter a donut in the field	void DONUT(void)	max 10 donuts for each
scatter a coin in the field	void COIN(void)	
borrow 10 coins from the TF owner	void BORROWCOIN(void)	max 300 coins in the field
drop a coin at current position	void DROPCOIN(void)	
pick one coin near current position	void PICKCOIN(void)	
check number of coin at near distance	void Q_COIN(int *ncoin)	
broadcast message to team members	void BCAS(char *msg)	
query team message	void Q_BCAS(int *etime, char *msg)	

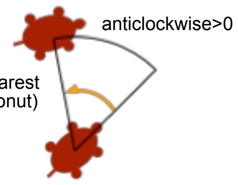
FINDER front sensor

default: 20
use RF() to change



result code = 0: no objects within effective range
1: turtle 2: donut 3: stone 4: wall found

RADAR angular sensor

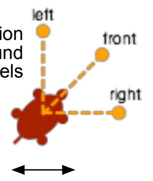


angle to the nearest object (turtle or donut)

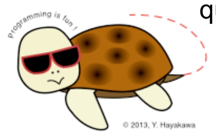
measurement error: ±20 deg

SONAR obstacle detector

for each direction
0: no obstacle 1: obstacle found within 20 pixels



size of turtle: 20 pixels in diameter



Java, Ruby, and Python APIs are available at TurtleField web site:
http://seaotter.cite.tohoku.ac.jp/coda/tfield/index.html

Usage of Turtle Field server

right click on screen to pop up menu

clear field	clear drawing
erase zombie	erase turtles that lost TCP connections
toggle graphics/battle mode	in battle mode, background texture changes
toggle private/public mode	in public mode connections from any addr. are accepted. window title bar changes
capture screen	save screen shot in home (UNIX) or desktop (Windows)
kill all turtles	
exit	

press capital **M**: enter maze mode
3: toggle b/w 2D and 3D modes

http port for Turtle Field Live: 2280