Artificial Intelligence used in a Battle Simulation

New Mexico
Supercomputing Challenge
Final Report
April 7, 2010

Team #73
Los Alamos Middle School

Team Members
George Barnum
Mohit Dubey
Ben Liu

Teacher
Clara Vigil

Mentor
Bob Robey

Artificial intelligence is the field of computer science, which programs machines to do human-like tasks. Through AI, scientists can solve problems or complete tasks that cannot be done by human intelligence alone. The program for the machines intelligence is usually based on an algorithm, a system of finite steps that solve problems.

AI was first discussed during a meeting at Dartmouth College in 1956. Many famous scientists were there including John McCarthy, Marvin Minsky, and Herbert Simon who helped pioneer AI. Throughout the next 40 years, the scientists working on AI had many breakthroughs and pitfalls. Finally, in the 1990s and 2000's, artificial intelligence achieved its greatest successes. Using modern day computers people could use intelligent machines to do new things and solve many unsolved problems.

Today, artificial intelligence is applied to very many things we do. One major on is computer science, where researchers use algorithms to simply run computers and give them the ability to run applications (Microsoft Word, Firefox, etc.) and do tasks (Spell check, run a search of computer documents, etc.). AI is also commonly used in video games to make them more fun and interactive. Banks use AI to invest in stocks, balance funds, and organize staff operations. Companies use intelligent machines to do tasks, which could be dangerous or time-consuming. Lastly, AI can be used to simulate battle to help plan strategies and decrease collateral damage, which we were trying to do with our program.

To model our battlefield we used C++ with a graphics library called Allegro to visualize the data. The battlefield was grid based and the movement of the units was made turn based because we were not very experienced in C++. We decided to try to simulate a relatively small number of units for the same reason. By altering the variables in the program we would create different situations and repeatedly test the AI we created. However, we could not finish our goal in the available time.

We first made a simple movement program that would move two units across the screen when the arrow keys were pressed and switch between the two with the space bar (labeled Red Side Movement loop in the code in the Appendix). Then we tried to make another two units move towards the red units and stop when they were adjacent to them (labeled Blue Side Movement loop), but when we tried to run the code with that, the program became unresponsive and crashed.

We had planned to have the artificial intelligence navigate around buildings and pick the best way through varying terrain. We were going to have it calculate each unit's path to the target with an A* search, an algorithm that compares the cost of moving to a space with the distance from that space to the goal, and run a test on each square that it would be going through to see if it would run into anything or get too close to an enemy unit and if it did it would make small changes to its path and recheck it for speed.

We would like to thank Bob Robey for helping us with our code problems and giving us advice.

Appendix

```
int rsx = 0;
int rsy = 0;
int rsxi = 0;
int rsyi = 1;
int rsm = 3;
int rsmi = 3;
int rsh = 3;
int rshi = 3;
int bsx = 25;
int bsy = 0;
int bsxi = 25;
int bsyi = 1;
int bsm = 3;
int bsmi = 3;
int bsh = 3;
int bshi = 3;
int tx = 0;
int ty = 0;
int txi = 0;
int tyi = 0;
int tt = 0;
int sol = 0;
int turn = 0;
int space_bar_down = 0;
int left_press = 0;
int right_press = 0;
int down_press = 0;
int up_press = 0;
int enter = 0;
volatile long speed_counter = 0;

void increment_speed_counter()
{
 speed_counter++;
};
END_OF_FUNCTION(increment_speed_counter);

int main(int argc, char *argv[]){
 FILE * fout;
 allegro_init(); // Initialize Allegro
 install_keyboard(); // Initialize keyboard routines
 install_timer(); // Initialize the timer routines
 install_mouse();
 LOCK_VARIABLE(speed_counter);
 LOCK_FUNCTION(increment_speed_counter);
 install_int_ex(increment_speed_counter, BPS_TO_TIMER(60));
 set_color_depth(16);
```

```
set_gfx_mode(GFX_AUTODETECT, 640,480,0,0);
BITMAP *buffer = create_bitmap(640,480);
BITMAP *redsoldier = NULL;
BITMAP *bluesoldier = NULL;
redsoldier = load_bitmap("redsoldier.bmp",NULL);
bluesoldier = load_bitmap("bluesoldier.bmp",NULL);
while(!key[KEY_ESC]){
 while(speed_counter > 0){
  if (turn == 0){                                              //Red Side ovement loop
   clear_keybuf();
   if(key[KEY_SPACE]){
    if(space_bar_down==0) {
     sol=(sol<1)? 1: 0;
     space_bar_down = 1;
    }
   }//key space
   if (sol==0){
    if (rsm>0){
     clear_keybuf();
     if(key[KEY_DOWN]){
      if(down_press==0){
       if(rsy < 18)rsy++;
       down_press = 1;
       rsm--;
      }//down_press
     }//KEY_DOWN
     clear_keybuf();
     if(key[KEY_UP]){
      if (up_press==0){
       if(rsy > 0)rsy--;
       up_press = 1;
       rsm--;
      }
     }
     clear_keybuf();
     if(key[KEY_RIGHT]){
      if (right_press == 0){
       if(rsx < 25)rsx++;
       right_press = 1;
       rsm--;
      }
     }
     clear_keybuf();
     if(key[KEY_LEFT]){
      if(left_press == 0){
       if(rsx > 0)rsx--;
       left_press = 1;
       rsm--;
      }//left_press
  }//KEY_LEFT
 }//rsm
}//sol
else{
 if (rsmi>0){
  clear_keybuf();
  if(key[KEY_DOWN]){
   if(down_press==0){
    if(rsyi < 18)rsyi++;
    down_press = 1;
    rsmi--;
   }//
  }//
  clear_keybuf();
  if(key[KEY_UP]){
   if (up_press==0){
    if(rsyi > 0)rsyi--;
    up_press = 1;
    rsmi--;
   }//up_press
  }//KEY_UP
  clear_keybuf();
  if(key[KEY_RIGHT]){
   if (right_press == 0){
    if(rsxi < 25)rsxi++;
    right_press = 1;
    rsmi--;
   }//right_press
  }//KEY_RIGHT
  clear_keybuf();
  if(key[KEY_LEFT]){
   if(left_press == 0){
    if(rsxi > 0)rsxi--;
    left_press = 1;
    rsmi--;
   }//left_press
  }//KEY_LEFT
 }//rsmi
}//else
if(!key[KEY_SPACE])space_bar_down = 0;
if(!key[KEY_DOWN])down_press = 0;
if(!key[KEY_UP])up_press = 0;
if(!key[KEY_RIGHT])right_press = 0;
if(!key[KEY_LEFT])left_press = 0;

if(key[KEY_ENTER]){
//if(enter = 0){
  turn=1;
  rsm=3;
```

```
    rsmi=3;
    enter = 1;
//}//enter
  }//KEY_ENTER
  if(!key[KEY_ENTER])enter = 0;
 }//turn                                                  //End Red Side Movement loop
 if (turn == 1){                                          //Blue Side Movement loop
  tx = rsx;
  ty = rsy;
  if (tt == 0){
   if (bsm > 0){
    if (bsx > (tx + 1)){
     bsx++;
     bsm--;
    }//tx
    if(bsx < (tx - 1)){
     bsx--;
     bsm--;
    }//tx
    if (bsy > (ty + 1)){
     bsy++;
     bsm--;
    }//ty
    if(bsy < (ty - 1)){
     bsy--;
     bsm--;
    }//ty
    if((bsx == (tx - 1)) or (bsx == (tx + 1))){
     tt = 0;
    }
   }//bsm
   else{//
    tt = 1;
   }
  }//tt
  else{
   bsm = 3;
   turn = 0;
  }//else tt
 }//turn                                                  //End Blue Side Movement loop
 speed_counter--;
}//speed_counter

show_mouse(NULL);
clear_to_color(buffer, makecol(0,255,0));
rectfill(buffer, 0, 324,640,480,makecol(0,0,0));
textout_ex(buffer, font, "Soldier 1", 70, 325,makecol(255, 255, 255), -1);
textout_ex(buffer, font, "Soldier 2", 70, 345,makecol(255, 255, 255), -1);
if (rsh > 0){
 rectfill(buffer, 0, 325,20,335,makecol(255,0,0));
 if (rsh > 1){
  rectfill(buffer, 22, 325,42,335,makecol(255,0,0));
  if (rsh > 2){
   rectfill(buffer, 44, 325,64,335,makecol(255,0,0));
  }
 }
}
if (rshi > 0){
 rectfill(buffer, 0, 345,20,355,makecol(255,0,0));
 if (rshi > 1){
  rectfill(buffer, 22, 345,42,355,makecol(255,0,0));
  if (rshi > 2){
   rectfill(buffer, 44, 345,64,355,makecol(255,0,0));
  }
 }
}
if (bsh > 0){
 rectfill(buffer, 640, 325,620,335,makecol(0,0,255));
 if (bsh > 1){
  rectfill(buffer, 618, 325,598,335,makecol(0,0,255));
  if (bsh > 2){
   rectfill(buffer, 596, 325,576,335,makecol(0,0,255));
  }
 }
}
if (bshi > 0){
 rectfill(buffer, 640, 345,620,355,makecol(0,0,255));
 if (bshi > 1){
  rectfill(buffer, 618, 345,598,355,makecol(0,0,255));
  if (bshi > 2){
   rectfill(buffer, 596, 345,576,355,makecol(0,0,255));
  }
 }
}
if (rsh > 0){
 masked_blit(redsoldier, buffer,0,0,(rsx*25),(rsy*18),25,18);
}
if (rshi > 0){
 masked_blit(redsoldier, buffer,0,0,(rsxi*25),(rsyi*18),25,18);
}
if (bsh > 0){
 masked_blit(bluesoldier, buffer,0,0,(bsx*25),(bsy*18),25,18);
}
if (bshi  > 0) {
 masked_blit(bluesoldier, buffer,0,0,(bsxi*25),(bsyi*18),25,18);
}
acquire_screen();
show_mouse(buffer);
```

```
  blit(buffer, screen,0,0,0,0,640,480);
  release_screen;

 }//KEY_ESC
 show_mouse(NULL);
 destroy_bitmap(redsoldier);
 destroy_bitmap(buffer);
 allegro_exit();
 return 0;
}
END_OF_MAIN();
```