Call To Power 2 API: Status Report
Ushhan Gundevia, Stephen Lee-Urban, Héctor Muñoz-Avila
Department of Computer Science & Engineering;
Lehigh University; Bethlehem, PA 18015; {udg204,sml3,hem4}@lehigh.edu

Abstract. This document describes the current status of the API that we developed for making possible to control Call To Power 2 from TIELT. The API description is as of March 17, 2006. Call To Power 2 is a turn-based strategy game that is targeted for evaluation of high level transfer tasks. We also developed a communication interface between Call To Power 2 and CTP2.

1. Introduction
The DARPA/IPTO Transfer Learning (TL) Program focuses on the development, implementation, and evaluation of TL technology in which performance benefits are observed for solving a target set of problems by first training on a source set of problems (where significant differences exist between these two problem types). Evaluations are conducted according to a methodology that is documented in the TL Broad Evaluation Plan (Fitzgerald & Aha, 2005).

Call To Power 2 (CTP2) is a strategy game, its code is available as open source. Lehigh University is developing an API for CTP2 and making it available from TIELT. CTP2 is targeted as a testbed for high-level transfer tasks by the NRL evaluation team. The main characteristics of CTP2 are:

- CTP2 is turn-based strategy game. Opponents alternate turns when playing the game. During one turn there is a finite set of actions that a player can make. Once a player makes his/her actions, he/she relinquish control to the other player to make his actions, and so forth.

- CTP2 is an imperfect information (partial observant) game. Players only see part of the world state. CTP2 implements the fog-of-war, whereby parts of the map that have not been explored cannot be seen. Also the player cannot see what military units and what types of buildings other players have in their cities.

- CTP2 is a chance game, as opposed to a deterministic game. That is, executing the same action on the same situation may have different outcomes. For example, when ordering a unit A to attack another unit B, there is an element of chance deciding if the unit B is hit. If the unit B is hit, there is another element of chance determining the amount of damage done to it.

- CTP2 allows for multi-scale decision making. It allows for strategic, operational, and tactical decisions. An example of a strategic decision is where to place a city. In CTP2, cities serve to produce units, as staging based, as sourced of income, and to control neighboring territories (called zones of influence). An example of an operational decision is the disposition of forces in a region and an example of a tactical decision is assigning tasks to units for assaulting an enemy’s city.
The appendix contains a game description form the perspective of game-play taken from http://apolyton.net/ctp2/, which also maintains the open source of the game.

2. API So far

The API as of March 17, 2006 concentrates on executing in-game actions. Not all possible actions are currently implemented. Rather we are working on obtaining a kernel API, and enhanced according to the requirements of the Transfer Learning project. The API has been divided by categories, as follows.

Creation of: settlers, workers, military units

API call:

CityBuild(const Player *player_ptr, const sint32 cityId, const sint32 unitType)

This function is used to build a unit (e.g. a warrior) from a city. The parameter cityId is the id of the city, unitType is the unit type that has to be built (refer to Units.txt in ctp2_data\default\gamedata\) and player_ptr is a pointer to the player. It returns true or false.

As it is currently implemented, it does check if the build queue is empty before adding the next unit. It’s not required; it’s just implemented it that way. The unit is added to the end of the build queue and not the beginning.

Creating a city

API call:

Settle(const Player *player_ptr, const sint32 settlerId)

This function is used to create a city. The parameter settlerId is the id of the settler (a special unit in CTP2 that can create cities) and player_ptr is a pointer to the player. It returns true or false. The settler builds the city at its current location.

Adding city improvements

API call:

CityImprove(const Player *player_ptr, const sint32 cityId, const sint32 improveType);

This function is used to build an improvement (e.g. Shrine, Granary etc.) in a city. The parameter cityId is the id of the city, improveType is the improvement type that has to be built (refer to Buildings.txt in ctp2_data\default\gamedata\) and player_ptr is a pointer to the player. It returns true or false. The improvement is added to the end of the build queue and not the beginning.

Note that the build queue for a city is same for both units and improvements.
Moving units to a given location.

There are two functions that that performs this action

API call:
    MoveArmy(const Player *player_ptr, const sint32 armyId, const int direction)

This function is used to move an army to the neighboring block in the direction specified. If movement is not possible in that direction, it chooses the next one in a cyclic order until it’s possible or movement is not possible in any direction. In CTP2, all units are referenced by adding them to an army. The parameter armyId is the id of the army which has to be moved, dir is the direction it is supposed to move in, player_ptr is a pointer to the player whose army has to be moved. The directions possible are: N, S, E, W, SE, SW, NW, NE

API call:
    MoveArmyTo(const Player *player_ptr, const sint32 armyId, const MapPoint newPos)

This function is used to move an army to the specified point. Here armyId is the id of the army which has to be moved, newPos is the position to move to. It returns if the movement was possible. It uses an existing path finding algorithm to find a path to the given location. Only if it finds a path, does it move.

Attack with army

API call:
    AttackPosWithArmy(const Player *player_ptr, const sint32 armyId, const MapPoint pos)

This function will be used to attack a position with an enemy unit or an enemy city. The parameter armyId is the id of the army which has to be moved, pos is the position to attack. It returns if the attack was possible.

Military unit takes a defensive stand

API call:
    ArmyToDefend(const Player *player_ptr, const sint32 armyId)

This function is used to assign an army with a defensive stance. The parameter armyId is the id of the army which has to go to a defensive stance. It returns if the action was
3. Upcoming API enhancements

Our next priority is to get the information about the state of the game back into TIELT. The API will include commands for the following actions:

- List of enemy units and their locations that a unit can see
- Closest enemy unit that a unit can see
- Diplomatic Stance with every tribe
- Government type
- Gold available
- Round
- Age
- Research goal
- Tax rate
- Science
- Trade Offers
- Advances
- Score
- All army list
- All city list

4. Video Demo

A video demo can be downloaded from:

http://www.cse.lehigh.edu/InSyTe/integrationDemo1.wmv

The video has no sound. Here is a description of what is seen in the video: TIELT and CTP2 start. Here is the outline of the movie that we are planning to make. There are 2 settlers and 2 military units from the player controlled by the API initially. The two military units go into a defensive stance on the second turn. The two settlers settle. The first city makes a settler, the second a military unit. The new settler then settles a new city, which starts making an improvement. The military unit generated moves and goes into a defensive stance. The initial two military units are in an adversary’s controlled area. The adversary sends a offer for truce that is rejected, which causes the enemy to atatek.
"Call to Power 2 is a turn based game of strategy and empire building. You have to nurture a fledgling nation into the most powerful empire in history. How you weigh your desires to explore the world, discover advances, conquer other nations, and maintain peace will mean the difference between victory and defeat.

You will need to build cities, establish populations, and manage their needs. You will have to master the forces of nature and use land to your advantage without upsetting the delicate balance of the ecology. You must fend off barbarian invasions and engage other nations of all stripes in various arenas-on the battlefield and at the negotiations table. And, perhaps most important, you will shepherd your people through the cultural and scientific advances that will enable your progress through time.

You will have a vast array of resources available. How you allocate these resources will dramatically affect your empire's viability and prosperity. You may choose to sacrifice scientific advancement in favor of a strong military. You may concentrate on building a few large cities and minimize your expansion. You may set out to explore the world, only to find out that a sudden invasion of your homeland forces you to protect your people. Above all, you must manage your objectives thoughtfully and temper your personal desires by accommodating the needs of your ever-changing empire.

A variety of factors internal and external to your empire will affect your success. Such things as your proximity to rival empires, access to natural resources and goods, the defensibility of the terrain you control, and your access to navigable waterways will influence your choices.

The paths to victory in Call to Power 2 are many. You can conquer the world by destroying your enemies, or forge an alliance with every nation to achieve world peace."