CS 194 Proposal: Supernova

Jason Block, Palani Eswaran, Alison Kohl

Description of the Project

Supernova will be a future-themed mobile board game inspired by chess, but with several rule and gameplay variations.

In brief: Players will move their individual pieces as often as they'd like without waiting for turns. After moving a piece, it cannot move for *X* number of seconds (defined at start of match). Because there are no turns, the game runs in real time. Pieces directly based off of the 6 standard chess pieces (King, Queen, Rook, Bishop, Knight, Pawn) will be available in the game, along with other new pieces. Supernova will also be played on a larger square board, with changing terrain that affects piece movement and strategy.

The specific details of the game are described below and are extremely tentative. We are confident in our basic idea described above, but several critical aspects such as immobility time, board size, terrain, piece selection, and piece prices will be tweaked and possibly heavily changed over time as we test the game and decide what makes the most sense for balance.

With that said, our initial plan for the game is as follows:

- The player can move pieces according to their specific movement abilities (like in chess) whenever they'd like - there is no waiting for turns. The only restriction is that after a player touches a piece, that piece cannot be touched for *X* seconds. Pieces that are frozen because they were just moved will be visually evident - we will do this either with a timer over the piece or by changing the piece's color while it is frozen. Our initial plan is to set *X* at 3 seconds, but we will potentially change this to better balance the game. We may also potentially implement multiple game modes that vary this time to change the game speed.

- When a player makes a legal move with a piece that lands it on a spot occupied by an opponent's piece, the opponent's piece will be eliminated and the eliminating piece will take its spot (as in chess).

- The board will also have a number of features that make it different than a standard chess board -- namely terrain and other obstacles. For example, a Pawn would be the only piece that would be able to cross a cell with mountains.
    - In order to accommodate the existence of these obstacles we would make the board bigger than the traditional 8x8 size.

    - Other terrain, such as forest, would take more time to cross and thus pieces would have a 6 second timer moving across them.
    - There will be uncrossable terrain, such as water.
    - These obstacles and terrains would be randomly generated at the beginning of each game, and the terrain would be mirrored across the diagonal so as to not give an unfair advantage to either player.

- Each player will have a King and eight Pawns on his team. The King and the Pawns can move in the same was as they would in normal chess.

- Each player then has 30 virtual dollars to spend on his or remaining pieces.

- The purchasing rules for normal chess pieces, from the outset, will reflect the "relative value" of real chess pieces. Thus, the equivalent of bishops and knights will cost $3, the equivalent of rooks will cost $5, and the equivalent of queens will cost $9. These pieces can move in the same way as they would in normal chess.

- Supernova will also include new pieces and powerups that are not used in traditional chess. Examples:

    - Supernova will include a piece called a Wall. This piece acts as a blocker - it can never be taken and it cannot take other pieces. As such, no other piece can ever land on the space it's occupying. This piece can be moved one space in any direction and follows the same rule . It must be purchased with your 30 virtual dollars and costs $5. Each player is capped at one of these pieces.

    - Double Move Power Up: if you take 2 of the other person's pieces in 6 seconds, you get a double move, which allows you to move a piece and then immediately move it again without having to wait.

- Once a player picks all of his or her pieces, the player can place them anywhere on the row closest to him or her.

- After the setup is complete the game begins and both players are free to move any pieces they wish.

- The game ends when one player knocks out the other player's King.

This game will be optimized for iPad, but it can also be played on any iOS device. Users will have the ability to play another person with both people using the same device, play another person through the internet, or playing against a computer player, which will utilize machine learning techniques and will have varying levels of difficulty.

Along with the game itself we will also create a tutorial that displays when the app is first open and available any time later on. The tutorial will describe the rules and walk the player through a basic example of the game.

Need for the product

Supernova's purpose is mostly recreational. It will be an inventive, fast-paced game that we hope will be fun for anyone who enjoys strategy games and puzzles. While this game has a lot of elements that build on traditional chess, we are confident that chess players will find the game to be a welcome, fun, and casual change-up to the traditional rules.

Potential Audience

The audience truly is limitless. Obviously fans of real-time strategy games will be the most likely group who would want to play Supernova. This would also be a fantastic game to have in classrooms because it involves strategy and critical thinking. Elderly people might enjoy it as a way to keep their brains active because of its fast-paced nature and because it will be relatively intuitive to learn because of its similarities to traditional chess.

Some research into the ideas we conceived show that there have existed attempts (some successful) to make "real-time chess" games in the past. In particular, a desktop Internet game named "Kung-Fu Chess" was developed in the early 2000s by developer Shizmoo Games. Up until its discontinuation in 2008, the game steadily built up a devoted community of players from all over the world. Most of these players were also chess enthusiasts who were captivated by the action and fun of the game. We believe that by re-designing the game for mobile and updating it to reflect modern technologies with many other new aspects, we can recapture this enthusiasm among chess players.

Discussion of Competing Products

The primary competitors for Supernova are real-time strategy gaming apps, such as *Autumn Dynasty*, *AmoeBattle*, *First Strike*, *Land Air Sea Warfare*, and *Line of Defense: Tactics*.

Major Technologies Used

Because we are building Supernova for iOS we will be programming in Swift. Instead of using XCode to develop the application we will be using SpriteBuilder, an open source game development tool. SpriteBuilder allows us to build our graphical interface and easily connect it to our rules and logic.

In order to establish the rules for our game we will either create the logic ourselves or build on an existing open-source chess application (Stock Fish - https://stockfishchess.org/) by tailoring its rules to our game.

To build out our Bluetooth functionality we will be using Apple's External Accessory Framework, which provides support for communicating with external hardware connected to an iOS-based device through the 30-pin dock connector or wirelessly using Bluetooth.

We will use Git for version control.

In order to create the artificial intelligence opponent, we will primarily utilize several machine learning techniques from CS221 (Artificial Intelligence).

Will will start with the minimax algorithm, which is a recursive algorithm that assigns a specific value for every single state of the game (a state being the specific pieces on the board and their positions at a given time). A heuristic evaluation function is used to indicate how valuable a given state is, and thus which of the players (human or AI) is "winning" the game, and by how much, at any given time. The human(s) are the "maximizing" players, and all moves that positively establish their board position and get them closer to victory are assigned positive value. Meanwhile, the AI are the "minimizing" players, and moves that make them more likely to win are assigned negative values. Thus, at any given time, the AI will analyze the heuristically-evaluated value of the game state, and make the legal move that has the greatest negative weight at that time.

A good game-theory artificial intelligence will also look 1 or more moves into the future (plies) when evaluating the game state and deciding which move has the most negative value. Because our game is real-time, the number of possible states is immense compared to simple turn-based games. It will be necessary for us to read further about combinatorial game theory to try to find different machine learning algorithms and techniques that are appropriate. We have been reading about *quiescence search* and the *horizon effect*, which are very applicable to this situation, and plan to research other related technologies as well.

Resource Requirements

One resource that we need is an iPad(s), which multiple team members have. However, when putting in some initial research into the development and deployment of our app from the computer to an iPad, we found that we need to enroll in the iOS Developer Program, which costs $99 a year. We would rather not have to pay almost $300 to develop our product, so we are wondering if we could get this license through the Stanford Computer Science Department.

Potential Approaches

As we develop our game there is a good chance that we will often tweak the rules. The aforementioned game setup is merely what we will start off with, but if we realize that we need to adjust our gameplay, we will do so. We anticipate that the artificial intelligence will be one of the most difficult aspects of the project, so we will likely proceed by first attempting to create a prototype with only the basic gameplay complete. Ideally, we would be able to implement the local-play Bluetooth technology quickly, so that we can begin testing real games between users and get their feedback on which aspects they like and dislike. After we are confident that the gameplay mechanics are solid, we will implement the artificial intelligence. Finally, we will focus on the design aspects and improving the graphics. Throughout the process and especially at the end, we will be getting a large amount of user feedback and using it to tweak Supernova to make both the visuals and gameplay more appealing.

Assessment of Risks

One key risk is that no one on our team has ever done iOS development, but we are all interested in learning it. Clearly there will be a bit of a learning curve at the beginning as we gain our bearings in iOS and the related technologies we're using - notably SpriteBuilder. We plan on using Swift.

We are also unsure about how balanced and strategically rigorous this game will be. None of us have ever heard of a game like this and so we will need to be flexible as we move forward and adjust the game's rules and components as necessary to make it a fun and complex game.

Another potential risk lies in the engineering of the artificial intelligence opponent. Because our game is complicated, there are numerous possible game states at any given time, and thus it will not be possible for the AI to simply analyze every single possible move. It will be a challenge to make algorithmic decisions to address this issue, while maintaining a computer opponent that can actually play to a legitimate level.

Next Steps

The first step for us will be creating the game environment, which will not only entail getting familiar with Swift and iOS but also establishing the movement rules for our pieces and locking down this movement functionality. Then we will have to focus on adding in the terrain, obstacles, and power-ups to make the game truly unique. The final steps will be the pre-game functionality in which the players select their pieces and set up customizable aspects of the gameplay.