

SUPERNATURAL SELECTION

Proposal by: Chip Hulseberg

Description:

Supernatural selection is a real-time strategy game (RTS) focusing on AI design and procedural model generation. The goal of the game is to conquer the unwitting army of an opposing God (2 players, possibly more) using the subtle mechanisms of breeding. To start, players are given an army of followers with randomized abilities and stats. These abilities and stats are chosen from a set of scales that affect the unit's behavior. Some examples of these scales are melee oriented to range oriented, high survivability to low survivability, and reckless to cautious. Using a large number of ranges (possibly allowing user's to code their own AI add-on ranges) will produce diverse armies of varying levels of effectiveness. Some units will start off strong, such as a high survivability melee class, while some will be doomed to continual failure such as a low survivability, reckless, melee fighter. It is the player's responsibility to breed and cull his forces (using his/her godly powers of divine conception and natural disasters) to produce an army capable of dominating the opposing player.

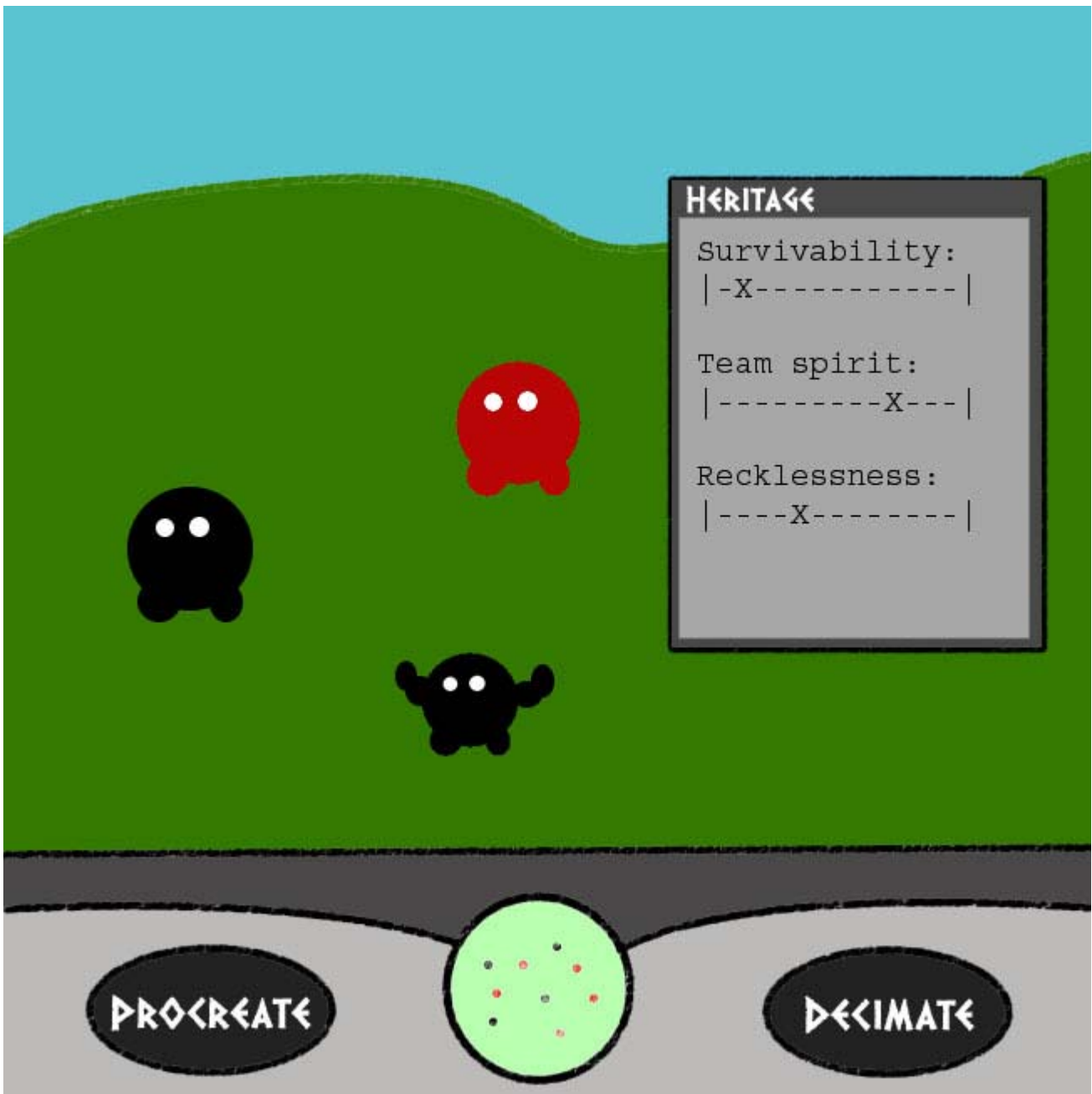
Specifically, breeding and culling mechanics are used to create new unit types. When two units are bred, they produce an offspring that combines the traits of the parents in a random way. Breeding is also affected by mutations, which can introduce traits not found in either of the parents. When a unit is culled from the population its life essence returns to the Gods to be forged into a new unit in the breeding process. Army size is limited by the total of number of souls that a God possesses, and the number of a God's units that have been slain. When an army is defeated, the game ends.

The focus of game play experience will be simple rules and controls that are nonetheless able to create a dynamic, emergent experience. The graphical design will echo these principles. Models will be constructed using rules instead of hard coded data. For example, a hardier unit will tend to be on a larger scale than a weaker unit. Some other possible model dimensions could be using a reddish tint to indicate recklessness, or shaking

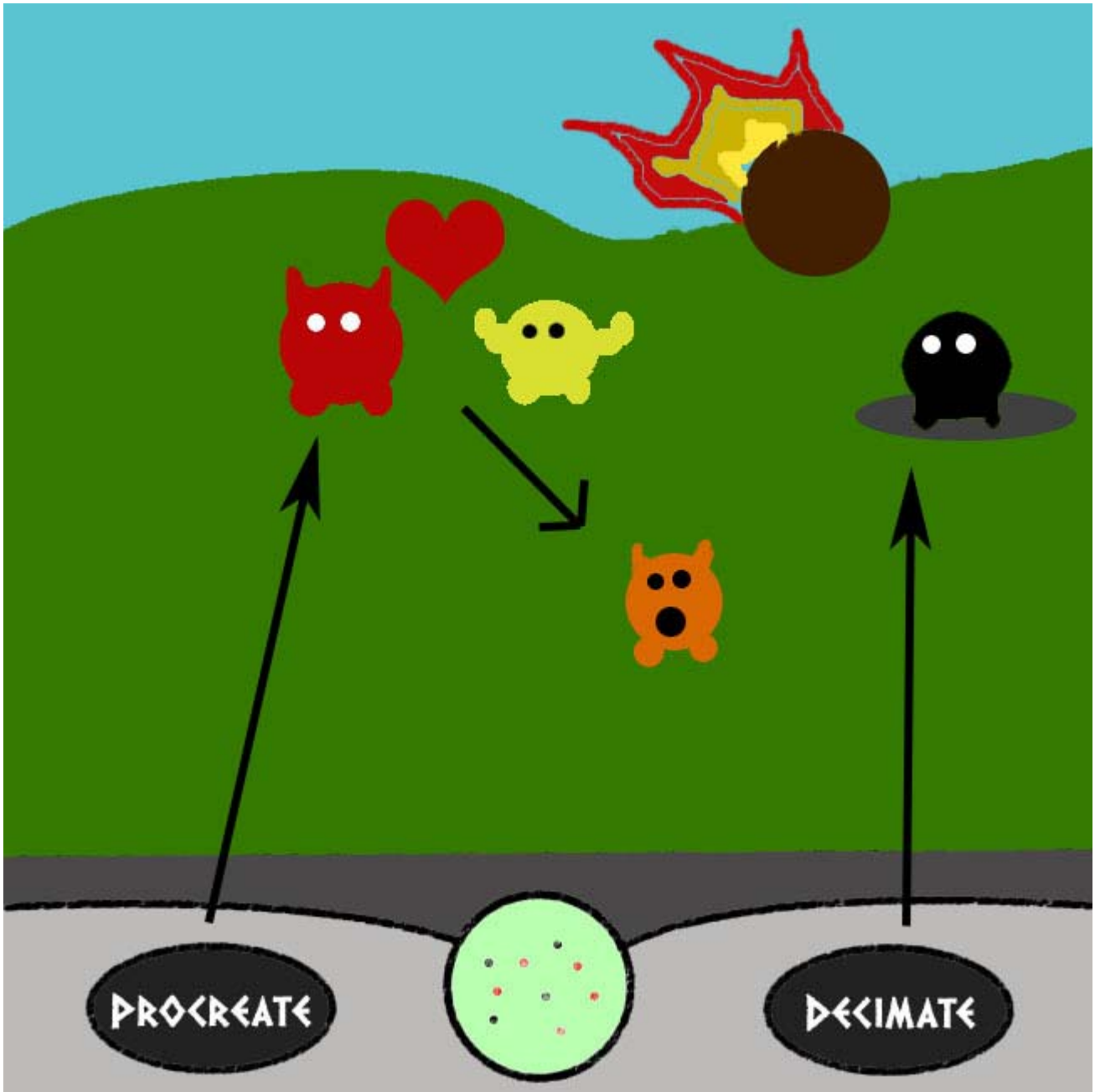
to indicate cowardice. The God's abilities will appropriately be on a large scale compared to the individual units. Using a rule based model will allow us to affect many creatures at once for dramatic affect.

Supernatural Selection benefits from the history of strategy games while adding something new at the same time. The concept of leading your army against an opponent's in a battle of wits has been around forever. However, *Supernatural Selection* will force you to not only fight your opponent, but your own units as well. This adds an inward looking creative component that is often missing in RTS. There are a million games where you can build an army out of the various provided units, but in what other game can you actually build the units? *Supernatural Selection* also boasts impressive replayability. Each time you play the game a different set of decisions or initial conditions can spiral into completely different circumstances. The focus is not on memorization of strategies or learning how to micro a huge number of units, but instead on adaption and creativity. Simplified controls are also a strength of *Supernatural Selection*. RTS is typically one of the most complex genres from the point of view of controls. Players are responsible for attacking with multiple armies, constantly repositioning them, building a base, and gathering resources simultaneously. *Supernatural Selection* on the other hand automates most of the attacking and positioning process, and instead produces complexity through an accumulation of small changes.

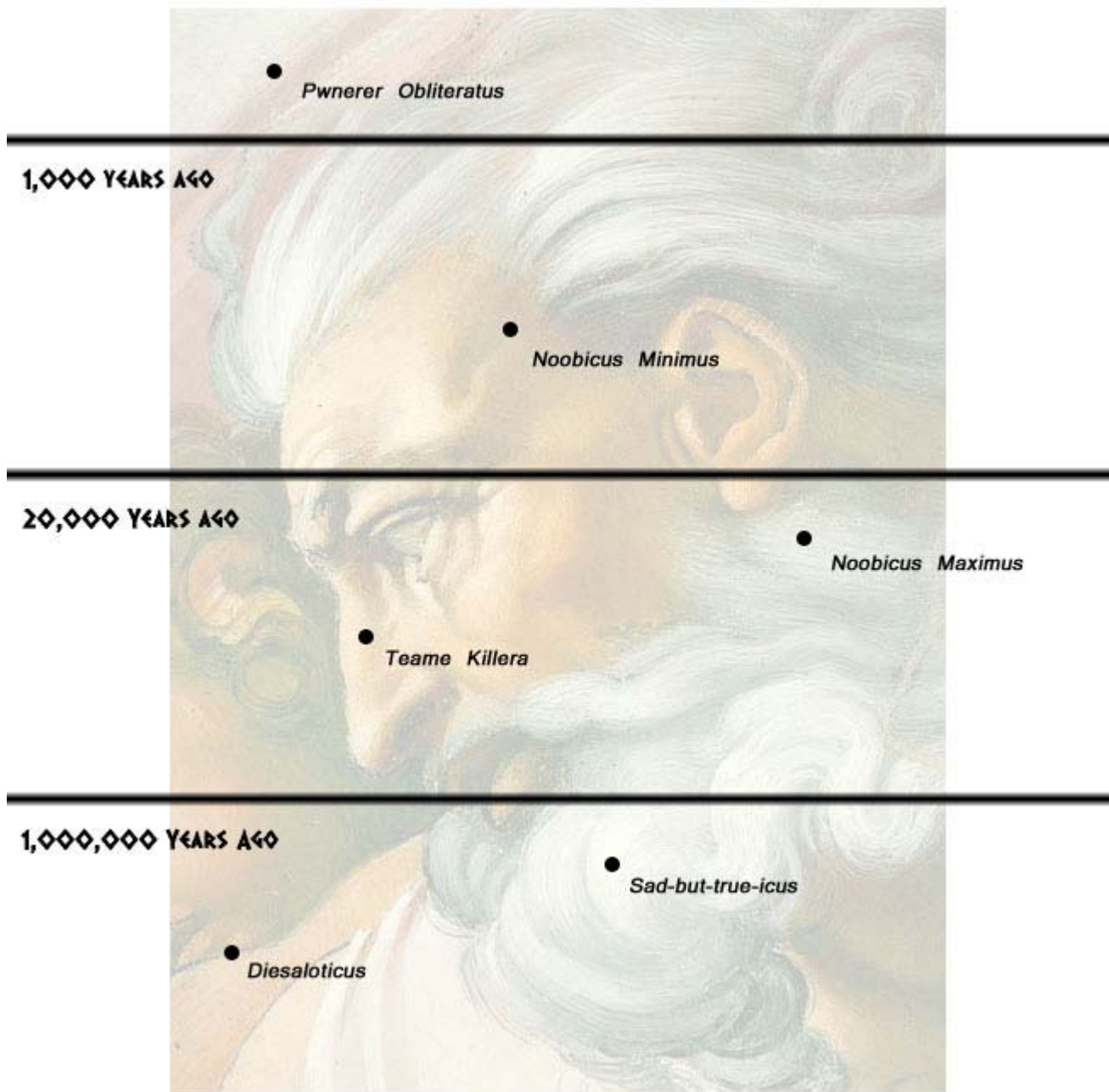
Because *Supernatural Selection* will focus on high level AI and model building concepts, we will most likely use a graphic engine such as OGRE to smooth over some of the low level graphics details. We will also use modeling software such as Maya to create the building blocks of the character models and the models for generic effects. We may also consider adding additional libraries to cover sound and basic networking.



Basic User Interface with Unit Info



Examples of the two breeding functions.



Exciting Advertisement!