**Process Model Report**

February 13, 2017

Team “Mango WEB”

Caleb Warren

Collin Epstein

Julian Bertini

**Introduction:**

This report describes our selection of a software process model for developing a 3D puzzle-based video game for mobile devices using the Unity game engine.

**Questions and Answers:**

**1.** **What primary process model does the team plan to use?**

The Team has decided that the Iterative Process Model is most worthy for a video game. We thought that building small components of our project to completion and combining them into a cohesive whole would be very useful. After completing these smaller tasks, we also thought it would be necessary to reevaluate the status of our game and add or remove future requirements as necessary. Both of these processes are the main operators for the Iterative Process Model.

**2.** **What secondary process model(s) does the team plan to use?**

We do not think a secondary process is necessary. If anything, we would use some pair programming from the Agile family of process models. At the current stage of the project, we do not see any issues with the Iterative Process Model that would require modification of the process. If we find after a certain iteration that we do need to modify our model, the Iterative Model will allow us the flexibility to do so.

**3.** **What project or team characteristics led to this decision?**

The primary function of a video game is that it is playable. We want our program to have a functional version at all times while adding discrete features. The success, failure and/or function of each added feature will determine what feature needs to be added next. The Iterative Process Model lends itself directly to creating software in this manner.

Additionally, this model will work best for our small team because we can independently work on coding individual features without necessarily being collocated with the rest of the team. This will increase our individual and collective flexibility while working on this project.

**4.** **Were any other alternatives considered and rejected? If so, why?**

We briefly considered the Spiral Process Model. However, upon further analysis of that model, the required rejection of each legacy prototype seemed wasteful for a video game project. We thought that creating complete and useful code during each iteration would be more productive than producing and discarding prototypes until a satisfactory program model is found.

We also considered the Waterfall Model at the beginning of brainstorming process models that would work well with a game. On further analysis of the Iterative Process, we realized that the Iterative Model takes the Waterfall Model and adds the availability of iterating on our software which, as described previously, seemed necessary in producing a video game.

**5.** **State the actual sequence of activities to be performed**

First, we will make a general outline of all desired requirements for the game. This would include gameplay mechanics as well as art and aesthetics. Then, an architecture will be fitted to our concept. Next we will need to design the game, focusing on what problems need to be solved and when. All of these activities will be decided on and produced as a team.

For each iteration, the team will select one or more requirements to implement and add to the current state of the game. If we select smaller, simpler features, each team member will each be assigned one feature to implement. In the case of larger, more complex and/or more essential features, team members will collaborate. Once a requirement is selected, the team member(s) in charge of that requirement will design and implement the necessary code. The team member(s) will then playtest the feature. If it functions as expected, other team members will verify the new feature by playtesting it as well. If the feature passes that round of testing, the feature will be integrated into the larger game, and all team members will playtest the new version of the game. Based on the results of playtesting, we will update our list of requirements and adjust the design of the game. In extreme cases, architecture revision and code refactoring may be necessary and will be addressed in the next iteration. These steps will be repeated until the final iteration is reached.

For the final iteration, we will verify all requirements necessary to the basic function of the game have been met. If we have additional time, we will include less essential requirements, such as including optional mechanics or creating an new aesthetic. If we reach our deadline and the game has met its basic requirements, we will release our final product.