

# CS 485 Group-Project Proposal

Due: February 26th 12:00pm, 2018

## OrigamiRunner



## Part I: Project Team and Allocated Roles

The team is composed of 4 people:

- **Gautier VAILLANT**: Developer, Game Designer
- **Maxime BACONNAIS**: Developer, Game Designer
- **Nicolas FEZ**: Sound Designer, Developer
- **Zackary BEAUGELIN**: 2D Artist, Animator, Developer

All the team members are responsible of game testing, document writing and project managing. The source code and the assets are available on the public Github repository: <https://github.com/Gysco/OrigamiRunner>

## Part II: High Concept Document

### Story Overview

The Game takes place in a paperish world. The cycle day/night has been perturbed by an anomaly which also engendered the enemies you encounter. The further the days past the longer the night is.

You are a character who lost the its soulmate to the anomaly. Your goal is to fight the anomaly's *monsters* and to survive for the longest time possible.

### Design goals

The game is designed to run on every hardware with slick graphics. We will do our best to make our game very simple to play and very fun!

### Features

The character will run during a sequence of 4 levels. Those levels will repeat until the player dies. When the fourth level is completed the speed is incremented and the player transition smoothly back to the first level. In each level he will have a tutorial (or adaption time to the new speed). The day is the tutorial (10% of each level, decreasing over reset) and the night is the part where the level has enemies.

### Target customer

The customers targeted by this game are arcade and casual gamers.

### Target hardware

As Unity offers a large panel of deployment targets, we will build our game for PC and Mac. We will also provide a WebGL version to play to the game in a web browser if it's possible. We will try to create an Android/iOS version, the gameplay could be adapted to smartphones easily.

### License

The game will be free and available to download on GitHub, source code and assets are licensed under the GNU General Public License v3.0.

## Part III: Game Treatment Document

### Gameplay

It's a side scrolling game, where the player has only few actions like: jump, shot and crouch. The player will learn a new action at each level:

- **Level 1 - Jump level:** In this level, there are holes and you will have to jump to avoid them, at night some monsters will appear, and you will have to avoid them by jumping over them.
- **Level 2 - Shot level:** What could be better than a super gun to kill anything that moves? Enemies will not have the slightest chance against your gun, be careful, they may be more aggressive!
- **Level 3 - Double jump level:** Platforms are out of reach, you'll need dexterity to reach them.
- **Level 4 - Crouch level:** Sometimes enemies fly, you'll have to be careful!

Each time the player completes the fourth level, he goes to level 1 again, but the speed is increased. This is an infinite run, and it finishes when the player dies, from an obstacle, the environment, or an enemy.

There are two parts in the gameplay: one for the day and one for the night, during the day it's like a tutorial, to test your new abilities and the speed of your character: there are no aggressive enemies, during the night the difficulty increases, and a lot of enemies appear.

### Level design

The levels are generated procedurally, that means this is a controlled random. This process ensures that the game is playable even if the level is different each time. The design of each element is a group choice, this means even if there will be many concept arts for some elements. If the group does not find an agreement the 2D artist will provide several other choices.



*Figure 1: Main Character*

## Technologies and software used

The game is built with Unity 3D and will use C# language for the *scripting*. The Unity engine allows the development of a game for many different platforms. Since the gameplay of the game is adaptable to the mobile devices, the game will be built originally for PC and Mac, then will be adapted to Android and iOS. We will then try to export the game with WebGL in order to create a web version.

The creation of the assets will be done with Adobe Photoshop and maybe with After Effects / Animate for some of the animations. The animator will then proceed to animate and make sure that his assets are compatible with Unity and will create the animation in the game engine.

The soundtrack will be composed with Adobe Audition and GarageBand. The sound effects will be also created with Adobe Audition and GarageBand.

The source code will be managed through the version controller Git on the website GitHub. Each member should be able to work on their own, without interfering with the development from each other. They will have to manage their own branch and merge correctly the project when they finished developing the features they were working on.