**Project Plan CECS 343**

**King of Tokyo**

Hunter Davis

Tymee Kong

Tanner Mindrum

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Authors |
| 9/27/2019 | 1.0 | Initial Project Plan | Hunter Davis  Tymee Kong  Tanner Mindrum |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

[**1. Overview**](#_dsyecct9nket) **4**

[**2. Goals and Scope**](#_q2e7vhpkywv1) **5**

[**2.1 Project Goals**](#_jj5tsjri2cs) **5**

[**2.2 Project Scope**](#_4xi23gkp8foi) **5**

[2.2.1 Included](#_rp0trnawgrl6) 5

[2.2.2 Excluded](#_aab43bb1gxlp) 6

[**3. Organization**](#_cwmd8ecssf7f) **6**

[**3.1 Organizational Boundaries and Interfaces**](#_scl96xrilshh) **6**

[3.1.1 Resource Owners](#_v3fni336tc28) 6

[3.1.2 Receivers](#_n51qgsjm5aru) 6

[3.1. Suppliers](#_32m53r5ezyri) 6

[**3.2 Project Organization**](#_6po819wvbph0) **6**

[3.2.1 Project Manager](#_g1cvw82hpoqu) 6

[3.2.2 Project-Internal Functions](#_fsca349rzk96) 7

[3.2.3 Project Team](#_rrajomscltfw) 7

[3.2.4 Steering Committee](#_b6f07eoltp84) 7

[**4. Schedule and Budget**](#_laqqnau4w49a) **7**

[**4.1 Work Breakdown Structure**](#_e5svlfmeesw3) **7**

[**4.2 Schedule and Milestones**](#_qtz7gxm5wgdb) **7**

[**4.3 Budget**](#_nimc808gvnzy) **8**

[**4.4 Development Process**](#_un1xwfts06cu) **8**

[**4.5 Development Environment**](#_xkm73guqx9re) **8**

[**4.6 Measurements Program**](#_k7lc37wjd1lb) **8**

[**5. Risk Management**](#_szeriwia69zu) **8**

**6**[**. Communication and Reporting**](#_c7imh922owre) **9**

**7**[**. Delivery Plan**](#_yxr60igqkvm) **9**

[**8.1 Deliverables and Receivers**](#_8zkpcqxu2sab) **9**

**8**[**. Quality Assurance**](#_f7zgqhtqgcrp) **9**

**9**[**. Security Aspects**](#_vfq5us4x4a37) **9**

**10**[**. Abbreviations and Definitions**](#_t1kb4hxfhe79) **9**

[**11. References**](#_3nwlrsfarx1r) **9**

[**14. Revision**](#_278gp5ult49b) **9**

# **1. Overview**

The motivation for this project is to fill the void of only being able to play King of Tokyo in a physical form. This project is being created to allow players to play it in more new ways and almost whenever they want using the digital version of the game. The customer is any person aged 8+ who wants to play King of Tokyo, especially those that want to play it in a digital form. The project will deliver a new digital version of the existing physical board game, King of Tokyo. The project will cost $0 USD to develop. It will take fourteen weeks to develop. There are no organizations involved, rather the project is being developed by three individuals and has one project manager. There are no other projects that depend on the result of the project nor are there any other projects that contribute their results to this project.

# 

# **2. Goals and Scope**

# 

## **2.1 Project Goals**

The project should function exactly like the physical version of King of Tokyo. It will have all of the components, assets, and follow the same rules and game mechanics. The project will be user-friendly and interactive, providing users with a visual to view while they play the game just as if they were viewing a physical game board. Our plan is to complete this project using a top-down development approach. We will begin at a high-level, creating strategic documents that outline the project and details of the game functions. Then, we will continually move to more specific documents with more details, such as diagrams, flowcharts, and a user manual. Finally, we will perform the actual software implementation at the end of the project lifecycle. As time goes on, technology advances very rapidly and it is easier for people to have access to computers and the internet. Since computers are so popular, there would be a wider audience for a PC platform of King of Tokyo. Creating a digital version of King of Tokyo would be convenient for people that do not have the ability to get together with their friends to play it. If they can play it on the PC it would be much easier for people to play with their friends. Technical wise, the digital board game will run on modern operating systems and will not require much disk space. It will be a fully-fleshed out program, meaning that it will replicate the actual physical board game to the fullest extent. The game will be playable on Windows, Mac, and Linux, meaning players can play the game using computer platforms.

## **2.2 Project Scope**

King of Tokyo is originally a tabletop game that supports between 2 - 6 players, where

they play a king of the hill style game. The entirety of the tabletop game and all of its rules will be ported to a digital version which includes dice rolls, power cards, spending energy cubes etc. Furthermore, this game will be free to play for ages 8+ and it will be convenient for players that want to play the game but do not have a physical version of King of Tokyo.

### 2.2.1 Included

This project will consist of a friendly UI for the user to easily understand. It will contain

all the game mechanics that are normally in the tabletop version of the game such as

rolling dice, attacking other players, and power cards etc.

### 2.2.2 Excluded

Physical, tangible items for the players to use while playing the game are excluded. The game will be solely digital, meaning any physical content will not be included with this product.

# **3. Organization**

The internal project organization affected by the project results are the three software developers involved in our team, and the project manager, Anthony Giacalone.

## **3.1 Organizational Boundaries and Interfaces**

### 3.1.1 Resource Owners

Resources owners are defined in 3.1.3

### 3.1.2 Receivers

Our one and only receiver will be our project manager, Anthony Giacalone. He will oversee the product and collect every deliverable until the project is complete.

### 3.1.3 Suppliers

|  |  |
| --- | --- |
| **Company: Product** | **Deliverable** |
| Atlassian: Trello | Project Management |
| Google: Google Docs | Documentation creation |
| Github Inc.: GitHub | Version Control |
| Microsoft: Microsoft Word | Documentation creation |
| JetBrains: IntelliJ | Code Creation |

## **3.2 Project Organization**

### 3.2.1 Project Manager

|  |  |
| --- | --- |
| **Role** | **Organization: Name** |
| Project Manager | CSULB: Anthony Giacolone |

### 3.2.2 Project-Internal Functions

|  |  |  |
| --- | --- | --- |
| **Function** | **Name** | **Comment** |
| Quality Assurance | Tymee Kong | N/A |
| Architect | Hunter Davis | N/A |
| UI/UX Designer | Tanner Mindrum | N/A |

### 3.2.3 Project Team

|  |  |  |
| --- | --- | --- |
| **Name** | **Availability** | **Comment** |
| Tanner Mindrum | 24/7 | UI/UX Designer |
| Hunter Davis | Tu/Th 2PM-4:15PM | Architect |
| Tymee Kong | Tu/Th 2pm - 4:15pm | QA |

### 3.2.4 Steering Committee

|  |  |  |
| --- | --- | --- |
| **Organization** | **Name** | **Comments** |
| HTT | Hunter Davis | N/A |
| HTT | Tymee Kong | N/A |
| HTT | Tanner Mindrum | N/A |

# **4. Schedule**

## **4.1 Work Breakdown Structure**

## 

## **4.2 Schedule and Milestones**

|  |  |  |  |
| --- | --- | --- | --- |
| **Milestones** | **Description** | **Milestone Criteria** | **Planned Date** |
| M0 | Start Project | Familiarize team with Diplomacy Rules  Solidify team | 9/10 |
| M1 | Planning  Utilities:  Google Docs  Trello  Github  Java IDE | Documentation:  Vision Document  Project Plan  UML Diagrams  Flow of game  User Manual | 9/20 |
| M2 | Development Milestone A | Minimum working product  Basic UI Implemented | 11/10 |
| M3 | Development Milestone B | Error catching, additional features,  Advanced UI Implemented | 12/10 |
| M4 | Complete Project | Final submitted project with all implemented features | 12/10 |

## **4.3 Budget**

N/A

## **4.4 Development Process**

In this project, we will use Scrum which is an Agile Software Development technique. This type of development process is essential to our type of project because it requires constant communication between team members to create different iterations of the project.

## **4.5 Development Environment**

|  |  |  |
| --- | --- | --- |
| Item | Applied for | Available by |
| Methods | | |
| Use Case | Requirement Capturing | M1 |
| Tools | | |
| Draw.io | Design | M1 |
| Google Docs | Documentation | M1 |
| Trello | Planning | M1 |
| Github | Source control | M1 |
| Discord | Communication | M1 |
| Visual Studio Code | Coding | M1 |
| IntelliJ | IDE | M1 |
| Languages | | |
| Java | Design | M2 |

## **4.6 Measurements Program**

|  |  |  |
| --- | --- | --- |
| **Type of data** | **Purpose** | **Responsible** |
| <# changed requirements> | Documents what requirements change throughout the development lifecycle. |  |
| <Bugs found> | Count bugs found to fix | Test lead |
| <Performance data> | To assess the achievement of project requirements | Test lead |

# **5. Risk Management**

**5.1 Project Risks**

* Lack of communication between team members
* Lack of communication between the client and the developers
* Failure to deliver the project on time
* Changes in requirements
* Unorganized code

|  |  |  |
| --- | --- | --- |
| Risks | Probability | Impact |
| Lack of communication between team members | 30% | A lack of communication between team members would prevent team members from dividing the tasks effectively. If tasks are not divided, then features will not be implemented as quickly as they should be. This type of problem would lead to a delay in the delivery of the final product. |
| Lack of communication between client and developers | 45% | This could lead to the developers delivering a product that the client does not want. If we do not keep constant communication the product could be completely different to what the client expects. |
| Failure to deliver project on time | 20% | Failure to deliver the project by the required deadline would start from the beginning of the project. If the team is not able to finish certain features by a certain deadline, then it would create delays within our project. Which could lead to our project not being delivered on time by the end of the semester. If the project is not delivered on time, then the group would lose points on the final project. |
| Changes in requirements | 30% | A change in requirements would lead to our team having to go back into the code and fixing our implementation of the project. If the changes in the requirement were minute, then implementing new code for those changes would not be difficult. However, if the requirements were changed drastically, the project may need to be delayed to implement the changes. |
| Unorganized code | 45% | Unorganized code would be difficult to manage. If the codebase is messy and difficult to read without documentation it could not be easily changed. People that are unfamiliar with the code would have to spend time deciphering what the code does. |

# **6. Communication and Reporting**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Type of Communication** | **Method/Tool** | **Frequency/Schedule** | **Information** | **Participants / Responsibilities** |
| **Internal Communication:** | | |  |  |
| Project Meetings | In-person | During dedicated lab session of class | Project status, next steps, problems, update on individual work, updated requirements | Software engineers, sometimes project manager |
| Sharing of project data | Project server (GitHub) | When requested or when necessary work is complete and needs to be pushed to server | Documentation, diagrams, and code | Software engineers |
| Code standups | In-person | Share code that was completed since last standup or meeting | Code written and problems solved | Software engineers |
| Milestone meetings | In-person | Discuss milestones | Project status | Software engineers, project manager |
| Final project meeting | In-person | Discuss presentation, overview of project capabilities | Share experiences, share completion of project | Software engineers, project manager |
| **External communication and Reporting:** | | |  |  |
| Project report | Trello | Weekly, sometimes daily | Project status, track tasks | Software engineers |
| External collaboration | Documentation, diagrams | Weekly, sometimes daily | Complete tasks | Software engineers |

# **7. Delivery Plan**

## **7.1 Deliverables and Receivers**

|  |  |  |  |
| --- | --- | --- | --- |
| **Ident.** | **Deliverable** | **Receiver** | **Planned Date** |
| D1 | Final Group Formation | Anthony Giacalone | 9/10 |
| D2 | Documentation | Anthony Giacalone | 10/15 |
| D3 | Prototype | Anthony Giacalone | 11/10 |
| D4 | Complete Project | Anthony Giacalone | 12/10 |
| D5 | Presentation | Anthony Giacalone | 12/10 |

# 

# **8. Quality Assurance**

Quality assurance is very important to the final product. The game should be free of bugs, input or process lag, and UI faults so that players can experience King of Tokyo in the same or better manner that they would if they were playing the physical game. Quality assurance is of top priority because the board game will have very detailed rules and functions. To keep close track of our design and development cycle, we utilize Trello to visually see the status of various tasks. Our team will perform standups that detail what we have worked on and how it contributes to the project. We will continually monitor our code and diagrams to make sure that our board game contains as little errors as possible.

# **9. Security Aspects**

**9.1 Source Management**

Source code will be kept on the King of Tokyo GitHub platform and should only be modified by the software engineers responsible for making the game. Any additional team member that joins the team must sign an NDA before gaining access to any company proprietary information. Specific team members will be responsible for the distribution of software and version control using GitHub. Any additional access should be read only through GitHub.

# **10. Abbreviations and Definitions**

NDA: Non-Disclosure Agreement

QA: Quality Assurance

UI: User Interface

UX: User Experience

# **11. References**

[Project Plan Template](https://trello-attachments.s3.amazonaws.com/5d7825d1741236750bee583c/5d8a91cad4681544322eddd7/3b1a8a6ee57dd81dd927f4b0f3ca8a54/Template_Project_Plan.pdf)

# **12. Revision**

TBD