Thoth Tech

# Project Documentation

Splashkit Games

## Background:

Thoth Tech was founded on February 22$^{nd}$, 2022. Its mission is to build, operate, and deploy world class education technology [1].
Splashkit is one of the technologies developed with the goal of making programming easier for beginners in the craft. It is an SDK developed in C++ with many libraries to assist in introductory programming education and can be used to create 2D games [1].
To showcasing Splashkit to the wider audience, arcade games is a project where students build games that can be downloaded to an arcade machine.

## Scope:

To deliver an authentication and authorization and games playground component for future integration to secure assets and build an arcade game that's compatible with a Raspberry Pi board for the arcade machine.

## Assumptions:

1. Solution for the backend server authentication is built with server to SSO broker relationship.
2. Games can also be built with the C++ language.
3. Games are secure from vulnerabilities.
4. Security checks are done for the games before playground entry.

## Technology platform/Architecture:

For the authentication and authorization component, loose coupling is the architecture and will be built with Splashkit SDK. Restful API call from the backend server will receive a Json Web Token issued by a Keycloak server. Backend server will be built in the C++ language.

The arcade games component will be built using Splashkit framework and the C# language. The game is compiled to run on a Raspberry Pi.

## Roles and Responsibilities:

Project Leader: Tanish Dhapola
Developers:

1. Himanshu
2. Mohamed Zirufaan
3. PHIL WILLIAMS
4. Riley Oussoren
5. Robert Osborne
6. Kiet Lam

Project Methodology:
Cleary defined milestones that will help to identify the progress of the applications.

## A draft project times schedule with milestones:

Team:

1. Project documentation
2. Technical documentations
3. Project hand over documentation

Authentication and Authorization component:

1. User stories

2. Use Case Diagram
3. Class Diagram
4. Deliver the binding relationship with calling server and SSO broker.
5. Playground games download solution.

Asteroid Game:

1. Build Games Application level 1.
2. Build Games Application level 2.
3. Build Games Application level 3.
4. Asteroid application technical documentation.

## Dependencies/Risks:

Authentication and Authorization component:

1. Deliver the binding relationship with calling server and SSO broker.
2. Keystore solution.
3. Storage at rest solution
4. X509 binding relationship
5. Low emission encryption and shuffler

Asteroid game:

1. Integration of each state of the game
2. Asteroid game bug testing.
3. Security checks of games before acceptance to the Games Playground.

## Acceptance Criteria:

Authentication and Authorization component:

1. Keystore solution: Secured by
2. Storage at rest solution
3. X509 binding relationship
4. Low emission encryption and scrambler
5. Playground games download solution.

Asteroid game:

4. Integration of each state of the game
5. Asteroid game bug testing.
6. Security checks of games before acceptance to the Games Playground.

## Change Management Process:

Authentication and Authorization component:

1. Keystore solution.
2. Storage at rest solution
3. X509 binding relationship
4. Low emission encryption and scrambler
5. Playground games download solution.

Asteroid game:

7. Integration of each state of the game
8. Asteroid game bug testing.
9. Security checks of games before acceptance to the Games Playground.