Diamond Miner, Final Report

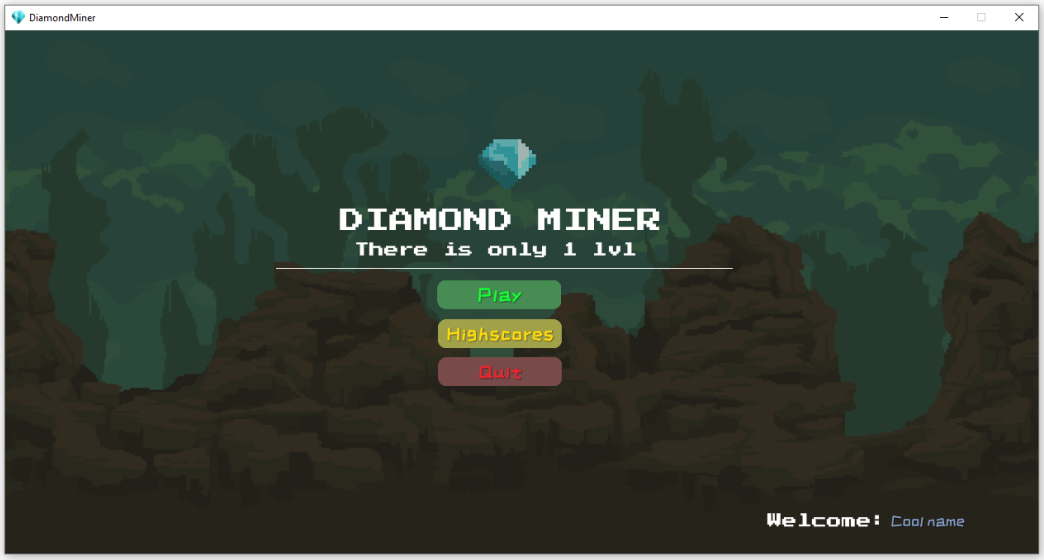
Computer Games Technologies, 2021

Group 8, Aarhus University

Peter Marcus Hoveling	au536878
Tobias Damm-Henrichsen	au553636
Anton Sakarias Rørbæk Sihm	au535993

Github: https://bit.ly/3E7x5RC

Assoc. Prof. Lukas Esterle
Due date: 21/12/2021



Department of engineering
Aarhus university
Denmark,
IHA.

1 Introduction

This report covers the work related to the course project in Computer Games Technology. In this course we have learned different genres of computer games, storytelling, technologies, user experience, alongside analysing the different stages of game development. This culminates in the planning, design and implementation of our own game. In our game Diamond Miner, the players will have to rely on their speed, creative thinking, and accurary against the game environment, to complete level after level of brain teasing madness to prove their worth against other players.

The game is developed using Unity and classic 2D platform microgame template[1]. This allowed our team to quickly gain a common base, from which the various features of the game could be developed. To further enhance our game various tutorials, video guides and reusable-assets[2] were used.

For development of our game idea inspiration was found in various previous played games. Examples include Super Mario Bros[3], the Rayman series[4], and This is the Only Level[5].

The report focuses on the games story, background and main mechanics alongside the more technical aspects and game development process. For further implementation details please see our GitHub repository[6].

1.1 Main contributions

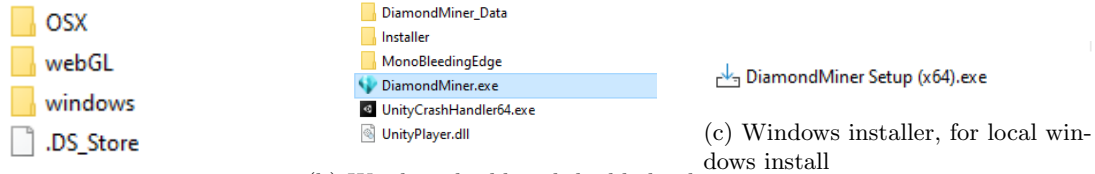
Work plan showing who was responsible for which part in the project. A large 'X' represents a lot of involvement in the given area, whereas a smaller 'x' indicates some form of participation.

Task	Anton	Tobias	Peter
Data Storage	X	-	-
Map design	X	X	X
Gameplay	X	X	X
Sound	-	X	x
Story development	-	X	-
Play testing	X	X	X
Animations	-	X	X
Controls	-	X	X
Menu	X	-	x
Score system	X	-	x
Visual design	x	x	X

[1]Microgame template - `https://learn.unity.com/project/2d-platformer-template?signup=true`
[2]Asset reuse - `https://assetstore.unity.com/packages/2d/environments/rocky-world-platformer-150009`
[3]Super Mario Bros - `https://da.wikipedia.org/wiki/Super_Mario_Bros.`
[4]Rayman series - `https://da.wikipedia.org/wiki/Rayman.`
[5]This is the Only Level - `https://da.wikipedia.org/wiki/Rayman.`
[6]GitHub repo - `https://github.com/TobiasDJ/Diamond-Miner`

1.2 How to play

Attached to our handin follows a zipped folder containing the compiled builds for windows, osx and webGL. In each of the compiled builds, a `README.txt` file is located for explanation on how to run. From the windows build however, is a two way installation and/or method of running. An installer is placed in the Installer folder, from which the game can be installed in its entirety. If this is not prefered, the `DiamondMinder.exe` can be launched but requires the attached files, *se figure 1.*



(a) Folder structure

(b) Windows build with highlighted *.exe file to quick play

(c) Windows installer, for local windows install

Figure 1: Compiled builds for OSX, Web and Windows

The web version is a proof of concept, and can be found on the following link: `https://diamondminer.pmh-unraid.dk/`. This version comes with some critical bugs and is therefor not recommended.

2 Game story and background

2.1 The customized TEN-pager

Following the discussion from lecture two the game is described using the TEN-pager design methodology. Our game title, see figure 2, is Diamond Miner - inspired by the waiting journey for our hero "Grabby", see figure 3. The intended platforms are PC, Mac, and Linux for ages 12 and up. Diamond Miner shows mild forms of violence towards human-like characters – and comes with a PEGI 12 rating[7]. The game is free-to-play with a projected shipment date set 21-12-2021.

Figure 2: Diamond Miner welcome

Diamond Miner features a single level that must played repeatedly. Each time the player progresses, aspects of the level change. Incorporated in the classic 2D platformer feel are challenging puzzles alongside speed-running possibilities based on time and deaths. When opening the game, players are greeted with intense rock music and the main menu. From here players can view the high scores or

[7]PEGI ratings - `https://pegi.info/what-do-the-labels-mean`

choose to play the game and test their skills. Upon finish the final score is calculated and top performances get listed in the high scores.

Figure 3: Main character - Grabby

Fighting his way through dark, deep, and forgotten mines in search of treasures - our hero delved too greedily and too deep. Earning him his nickname "Grabby". Awoke in the deep darkness he finds shadows, monsters, and challenges beyond his imagination. Controllable from the keyboard our hero relies on and abides instructions from the players using the W-key for jump, A-key for movement to the left, and D-key for movement to the right. Optionally the user can use left+right arrow keys and W-key for jump.

Figure 4: Concept art sprite sheet

In short Diamond Miner is an uncompromising speed-runner full of different brain-teasing stages.

You play as "Grabby" a greedy old diamond miner who in his search for treasures has been trapped in deep, dangerous, and forgotten mines. To escape this mysterious shadow filled world, you must learn to master the changing game level alongside the multiple challenges hidden as you progress. Enter a strange and gloomy world full of hazards where "Grabby" relies on your every move, if he ever hopes to escape and make is way back to the surface.

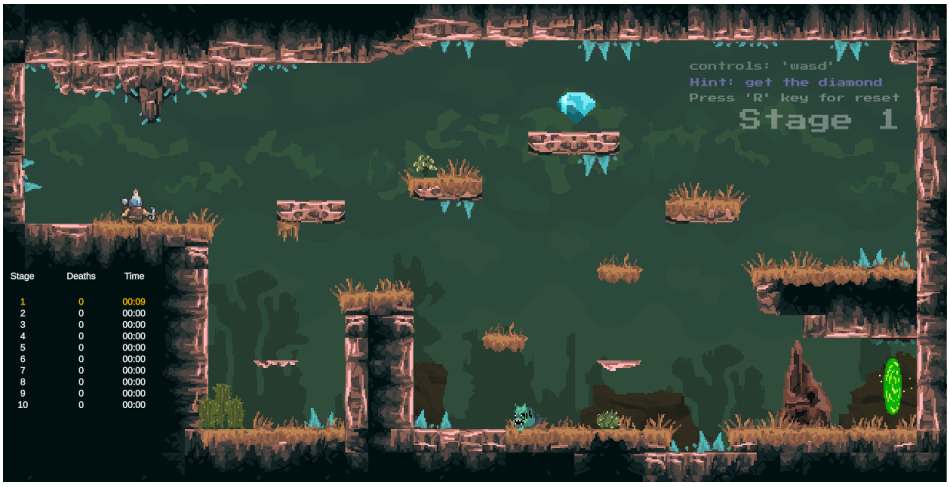


Figure 5: Level map

As shown in figure 5 “Grabby” finds himself in a dark gloomy place. His only option is to progress by collecting his beloved diamonds whilst avoiding the numerous dangers throughout the stage. In the bottom left corner player progress is shown alongside deaths and time spent in each stage.

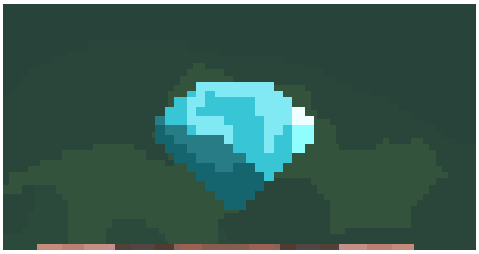
The overall end-to-end experience should leave the players frustrated all the while wanting more through improvements and the external glory available in the high scores. At first glance the game seems simple giving players an easy introduction to the gameplay. However, as the player progresses levels change resulting in a “easy to play – hard to master” game experience. To convey the game’s feel various sounds and music are used to support the gameplay.

(a) Portal - to progress to next stage

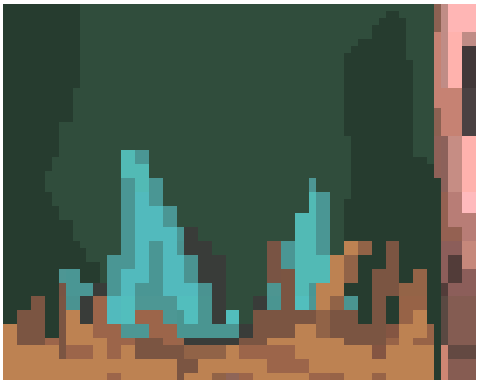
(b) Stone - blocking progress

(c) Monster - avoid or die

(d) Monster - avoid or die



(e) Diamond - collect by all means



(f) Spikes - death on contact

Figure 6: Gameplay mechanics and hazards

Figure 6 shows an overview of important mechanics essential in the completion of each stage. The various mechanics, hazards and enemies are described in table below as a list. For a closer description of each stage and its associated tasks and solution please see section 2.3.

Table 1: Gameplay mechanics explanation

Item	Id	Description
Portal	6a	"Grabby" must reach this portal in order to progress
Stone	6b	In order to reach the portal various tasks unique to each stage must be completed - maybe removing the stone
Monster	6c	Twisted creatures living in the darkness - do not get to close!
Monster	6d	Twisted creatures living in the darkness - do not get to close!
Diamond	6e	Collect but at what cost?
Spikes	6f	Numerous and comes in many variants throughout the levels - instant death if touched

2.2 Competitive aspect - Hiscore

The game is designed to be semi-multiplayer in the way that the players can compete with each other. The faster the ten stages is the completed the better score is archived. Do note that amount of deaths greatly increase the score. Having the lowest score will get you highest on the online rankings. The highscore is a MongoDB hosted on premise, which serves all players an updated all-time highscore, which is seen on figure 7.

Figure 7: highscore

When a player clear the final stage, their highscore is updated on the remote storage server. Their highscore is calculated by the formula in the underling section. The formulated penalties are calculated by time and deaths. Each death by a factor of 100 meaning that surviving 10 seconds without dying is the same score as one death.

$$score = (deaths * 100) + (totalTime * 10) \tag{1}$$

The highscore encourages the players to compete with their friends. Beating a friend's highscore, keeps the hype going, since the other friend wants come back and reclaim the rank 1 highscore for prestige.

Throughout the different stages the player will have a live overview of the number of deaths and the current time spent to complete each stage, as seen in figure 8. It gives the player an overview on which maps are critical for improvements, to hopefully achieve a better score.

Figure 8: Highscore metrics

2.3 Stages

In this section the various stages of Diamond Miner is described alongside their proposed solution.

Table 2: Stages of Diamond Miner

Stage	Title	Solution
1	Welcome	Collect the diamond, and then proceed to the portal previously blocked by the stone
2	Not straight forward	Solution is similar to stage 1. However, the controls have been inverted
3	Think before doing	No need to collect the diamond. Simply proceed to the portal previously unlocked by collecting the diamond. Collecting the diamond blocks the portal
4	No turning back	"Grabby" can only move to the right
5	Invisibility	Camera zoomed in on "Grabby" and stage features an invisible wall blocking access to diamond
6	What does the cat chase?	Teleport "Grabby" where ever the player clicks
7	Ahh! Can't see!	The platforms and spikes are not visible - but still very much there
8	Bouncy boy	A bouncy version of stage 1
9	Traverse reverse	Camera flipped and new monsters added
10	Only for champs!	Reversed controls and more monsters

3 Development process

As can be seen in section 1.1, all our team members covered multiple/overlapping roles, which we normally consider associated with a game development process. Similar to what we know from general software development processes – games are heavily a team effort. We all had tasks reassembling those from the character sheet of a programmer[8]. We also needed more specialized character sheets such as level designers, system designers, scripters, artists, testers, composers/SFX and writers. Being our first game and our first assignment together as a team – roles and tasks had to be created and shifted "on the fly". More on this in section 4. To support this we began the development creating a high-level overview development plan.

[8]Slides week 2 brightspace – Lukas Esterle

Development plan - ordered by priority.

- **Story development** - Creation of story and background. Gives a common understanding of the features needed to realize overall goals.
- **Initial level design** - Simple game mechanisms, user input, and prototype. Crucial to align understanding of requirements and enable parallelization of programming tasks. From here more specialized tasks could begin.
- **Level building** - As stated in section 2 we wanted to include game-changing variations - challenging and puzzling players. We also expected that this would allow us to experiment with a larger variety of game development aspects.
- **Character design and animations** - Extending the visual effects and animations to the previously designed core game features.
- **Multiplayer/Highscores** - Backend development to enable shared highscores between players.
- **Level design** - Development/improvements to game mechanisms
- **Game testing** - Not only by the development team, but include external parties to improve game experience.
- **Sound effects** - Tying user input, story, and game experience together more closely together.

Some of the produced artifacts are described below, where a discussion of the development plan is described further in section 4.

The first things agreed upon was that we wanted: 2D game developed in Unity utilizing the large developer base. Game should be inspired by speed-running - "simple to learn and hard to master". The initial wireframe - see figure 9.

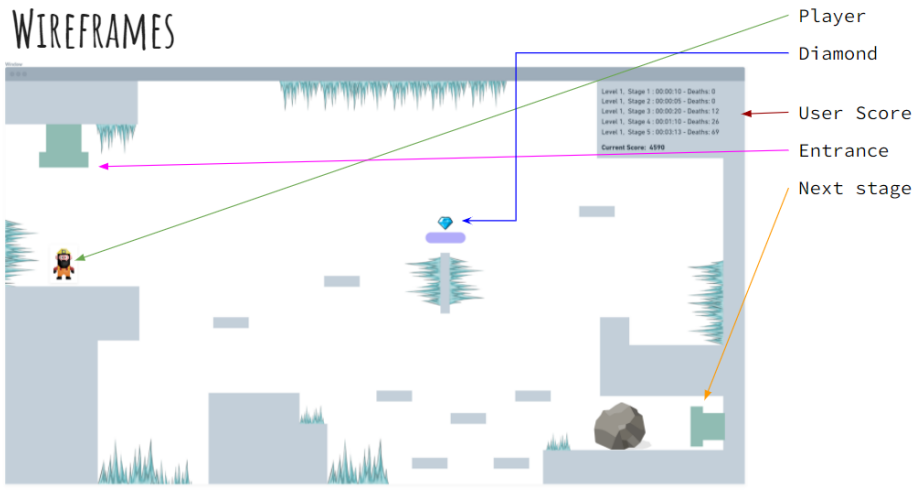


Figure 9: Initial wireframe

From here the initial level design could begin - where more specialized tasks could be integrated.

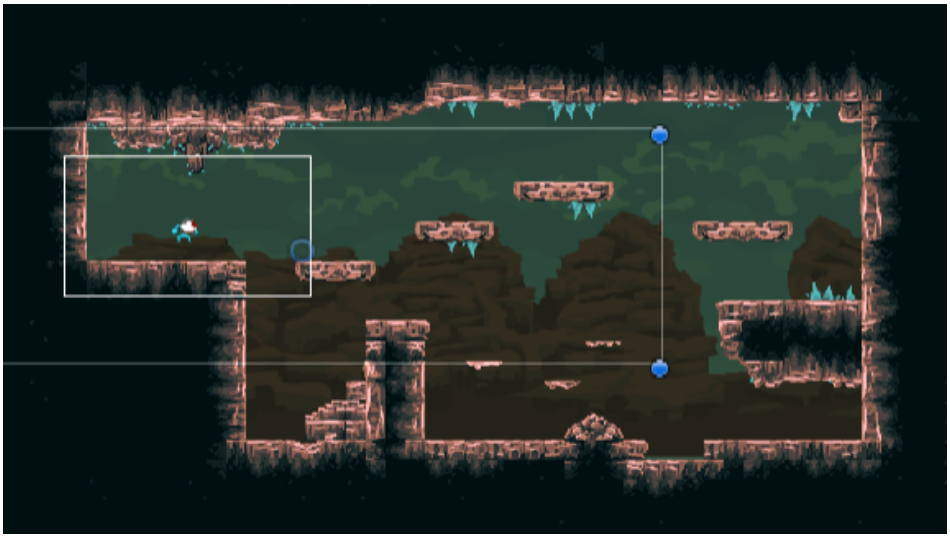


Figure 10: Initial level

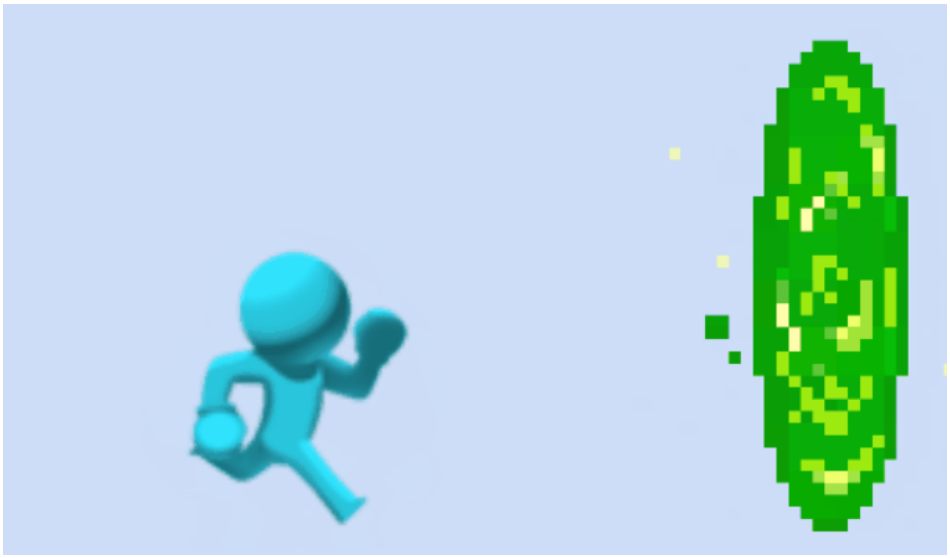


Figure 11: Initial level switch through portal

3.1 Alpha version - Play testers

In order to improve the game, and make the game more fun to play, we had people play an early prototype of the game, in order to get early feedback and inspiration for additional features regarding game play and map design. Some of the play testers, also reported a few bugs during play testing.

- **Features requests:**
  - One map where camera is flipped (Implemented after)
  - A map with 2 diamonds (Not implemented)
  - Invert map(Implemented after)
  - Reset button so player can restart if they had a bad run (Implemented after)
  - Highlight of current map (Implemented after)
  - Highlight of top 3 leader board (Implemented after)
- **Reported Bugs**:

- Highscore was wrongly spelled (Fixed after)
- Bug in stage 3 where sometimes the stage could not be completed (Fixed after))
- Highscore only fetched on awake (requires scene switch to update) (Not fixed)

4 Discussion and results

Going into to the development process it was unclear how far we would get, what challenges we would encounter, and the effort needed for a playable prototype. With aspects such as identifying roles, understanding development steps, needed features, associated challenges/common mistakes, learning the technologies, bug fixing, and so on - all relatively unknown domain, we created a naive development plan. In hindsight the roles, allocated tasks, and plan could have been optimized. E.g., with more specialized roles. However, this resulted in all members involved in close to all aspects of the game. This fact also contributed in a fun and learning development process - especially compared to previous projects.

When choosing our game idea, design, and gameplay - we maybe limited ourselves and could perhaps have been more ambitious - however, we managed to produce a playable version and enjoyed the end-to-end perspective. To conclude upon this it resulted in clear goals - where we going forward would like to explore more advanced game ideas/genres and their associated challenges.

For future iterations a large variety of aspects would make sense to include, such as a advanced combat system, multiple maps with different stages, co-op functionality, or desirable achievements. Various new enemies, combinations of these, and new environments could also help ensuring engagement - as covered in the Game AI lecture.

All in all it was a fun and learning experience - where we got to design and implement our own game. A valuable lesson was learned about the effort required in making great computer games, alongside the importance of a clear development plan that is aligned with the steps we learned in the lecture about game development processes. Furthermore, we are proud of our playable version - where a lot of the external game testers also gave great feedback to support this.