

Colonizing Bastards!

Ribin Chalumattu, Irina Huwiler, Tobias Krebs, David Krummenacher, Christoph Maurhofer

Game Description

Background Story

Towards the end of the 19th century, at the heights of the European expansion, there were only few white spots left on the maps of the colonial powers and trading companies. To close these gaps, a small number of brave explorers were sent to uncover the mysteries and unbury the magnificent treasures expected there.

Game Mechanics

The game is designed to be a 2D single-screen competitive multiplayer experience, seen from the side. Each player takes on the role of an explorer, fighting against the harsh environment of the jungle and his or her competitors to find and secure the biggest pile of treasures.

Match Setup and Environment

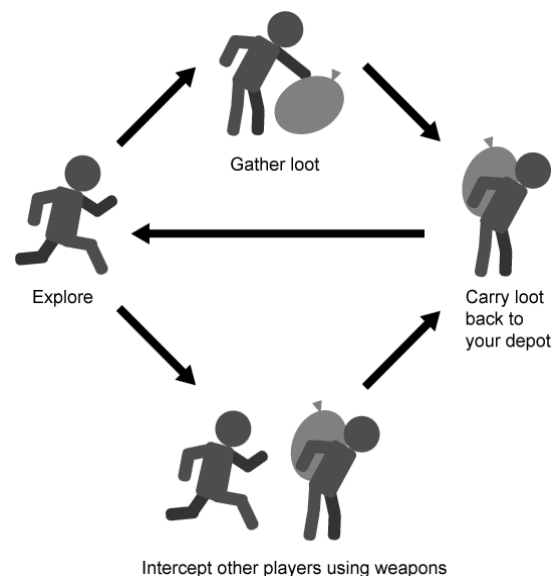
At the beginning of a game round, the two to four players are placed at a starting location (base camp) in the match environment (arena). This is the same place, where they need to deliver their loot to secure it. The arena itself is a part of an overgrown jungle with possibilities of for moving horizontally, but also using the environment for climbing up- and downwards or hiding in the shrubbery. Of course, the setting offers the possibility of creating additional maps, set in other regions of the world and not only the jungle. As described in the section 'goals', the players need to collect treasures, while keeping their competitors at bay. Treasures and weapons are scattered throughout the whole arena, but their value and strength increases with the distance to the base camp. At the beginning, all the loot is hidden away in the vegetation (or maybe also in some containers in a ruin, like pots). The initially visible paths are kept to a minimum, so that the players need to explore the environment. Towards the base camp, those lanes merge into one or a few only, to create a tense spot for fighting over valuables. Moving further away from base camp, they branch into puzzling pathways, forcing the players to find their way through the maze and carefully plan their extraction route.

An attractive addition to the game would be to have the game environment be more dynamic. For example by events like a part of it burning down, or having the ground flooded in some areas. Also, the environment may be hostile to the player by occasionally spawning dangerous animals, that hunt the players and cause them to drop their valuables or influencing their movement.

Goals

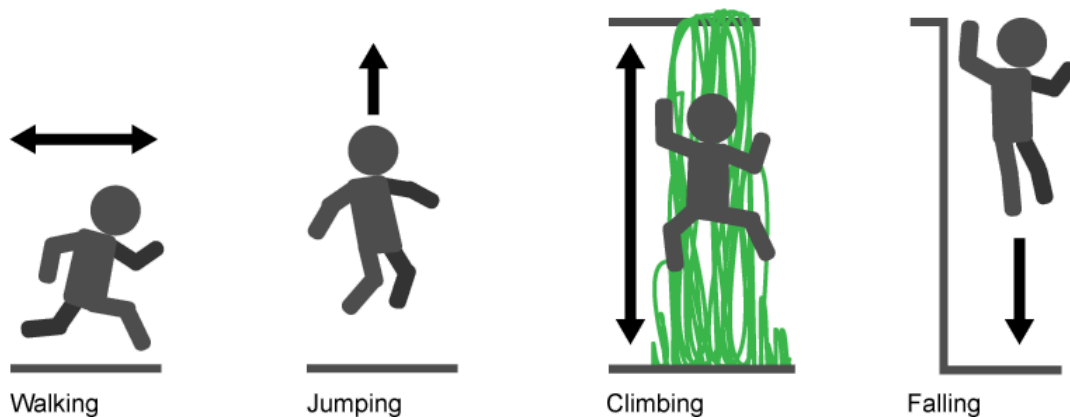
The goal of the players is to find and secure treasures. The collection with the largest value wins. The condition for ending a match could be: every treasure found and secured, a time limit (e.g. 5min) or a point limit with regards to the collected valuables.

In addition to the points collected by securing valuables, certain play styles could be rewarded. For example: most friendly player (robbed least number of other players), most aggressive player (robbed most other players), pathfinder (exposed highest number of new paths, e.g. by cutting down vegetation) and so on. These could give some incentive to perform some less lucrative actions or provoke skirmishes in a match.

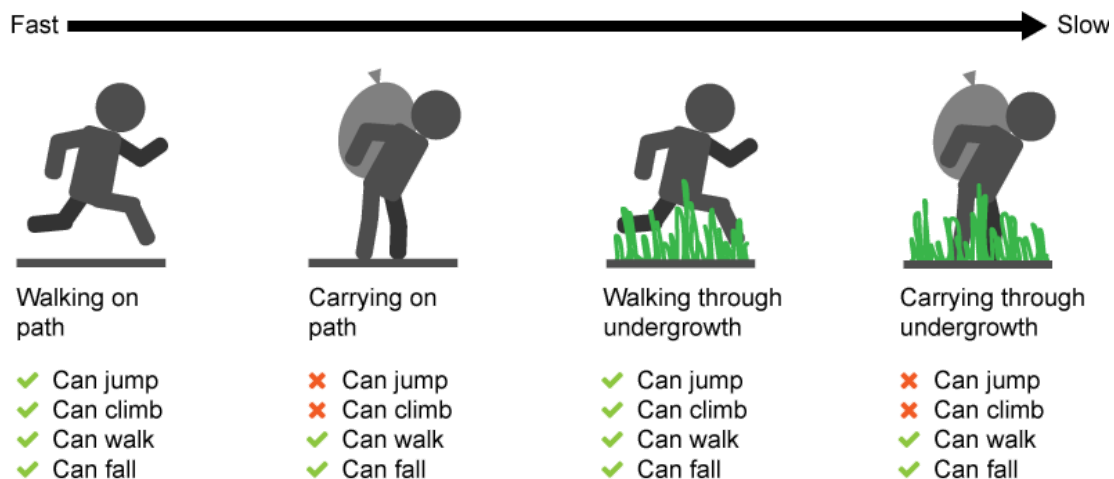


Player Movement

The players move on a horizontal and a vertical axis through the game environment.



The possibilities of movement and the maximum movement speed are constrained by the player carrying loot and the surrounding environment.



Player Abilities

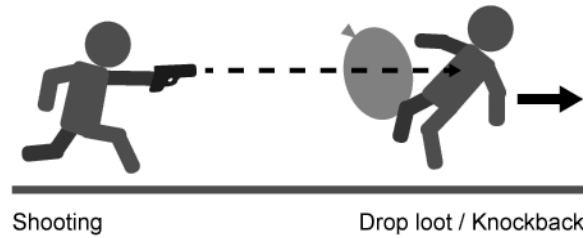
All players share the basic abilities: removing vegetation, hitting other players and shooting found weapons. Their number is kept low and the actions simple to offer a gentle introduction to the basic gameplay and enable intuitive controls of the characters, even in hectic situations.

Removing vegetation: In the game environment, there is shrubbery blocking paths and undergrowth lowering the movement speed. Additionally, vegetation may hide weapons or treasures. A player can use his or her machete to cut down these obstacles, revealing paths, loot or possibly some dangers like wild animals or traps.



Cut undergrowth with machete

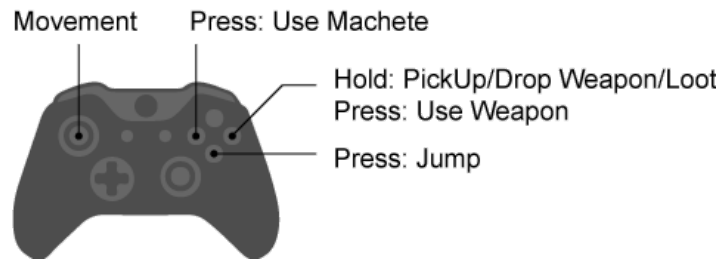
Hitting/shooting other players: When a competitor is hit or shot, he or she is knocked back and immobilized for a short amount of time. If he or she was carrying a treasure, the valuables are dropped. This gives the attacking player the possibility to steal the treasure.



An extension to those abilities could be allowing the players, to craft simple weapons or even traps to attack the competitors. Further, it could be interesting to equip the individual characters with a distinct skill (e.g. holds on to treasure when hit, able to “track” treasures in the near environment, ...). The origin countries of the explorers could give some inspiration.

Player Interaction

The player’s abilities are planned to be mapped onto the controller in a similar fashion to more popular games for intuitive playing (mapping below is not final).



Artstyle

Inspiration

As the game will be in 2d the art style will be primarily based on the works of Eyvind Earle. Particularly his work on Disney’s Sleeping Beauty. Its use of color and shapes will facilitate the platformer mechanics and the overall readability of the game.

Moodboard



Styleguide



Composition

The image will be composed by a foreground that frames the whole image, the middle ground where the game takes place and a background.



Character shape

Characters will consist of flat shapes, but due to the game being single screen will be more condensed than in Eyvind Earle's work.



Shapes

The dominant shapes of the environment will be abstracted into rectangular shapes to visualize the game's platforming mechanics.



Highlights

To further separate the characters from the background a rim light is added.



Fog

Desaturation between multiple layers will create the illusion of depth in the background and help make the background visually less important for the players.



Details

To break the flatness of the character shapes details are added through additional shapes in a single color (see the skirt).

Color palette

Environment

The foreground will be in very dark colors, the background in very desaturated greens. Gameplay elements will use colors to indicate their functionality.

Characters

Each character will have his own primary color to differentiate between players more easily (yellow, red, blue, magenta). Gameplay elements bound to a specific player character will share its primary color to better signal their relatedness.

Technical Achievement

Technical achievements are possible in many aspects of the proposed game.

- Proper and aesthetically pleasing multi-layer 2D graphics
- Physics system to accurately simulate the player's interaction with the environment (gravity)
- Procedural arena set-up (distribution of the treasures, weapons and dangers). Could even be extended to procedurally change the map by creating different paths and obstacles.
- Particle systems for more appealing effects and a more diverse game environment

Surely, not all of the points above completely lie in the scope of this course. But they provide interesting extensions for the upper layers and later development for further improvement of the features in a feasible state..

“Big Idea” Bullseye

Core Idea

Wrangling for loot.

Player goals and abilities, as well as spatial restrictions facilitates conflict between players.

Technical Innovation

See ‘Technical Achievement’.

Development Schedule

Development Layers

As recommended, the development of the game is split into multiple layers, designed to be completed from the minimum to the extras.

Functional minimum

Graphics: basic ability to display/move 2D sprites, simple camera

Physics: basic system to enable vertical gameplay

Game Mechanics: simple menu, evaluate input possibilities, one type of character, simple movement and implementation of the basic abilities, simple vegetation and treasures

Assets: single character sprite, first simple game environment

Pipeline: ability to load game environment from an external tool

Low Target

Graphics: multi-layer 2D graphics system, improved quality

Physics: system with more advanced collision detection (polygonal collision meshes)

Game Mechanics: improved movement, more diverse vegetation, more diverse treasures

Assets: more detailed game environment with additional assets, multiple character representations

Sound: simple sound design to explore possibilities and give basic feedback

Desirable Target

Graphics: support for weapons and for effects like fog, basic procedural map setup

Game Mechanics: weapons and a simple crafting system for creating simple weapons and a trap

Assets: assets for the weapons and the crafting system

Sound: some background music, improved sound-effects

High Target

Graphics: basic particle system for better effects

Game Mechanics: wild animals and traps in the game environment, individual skills for the characters

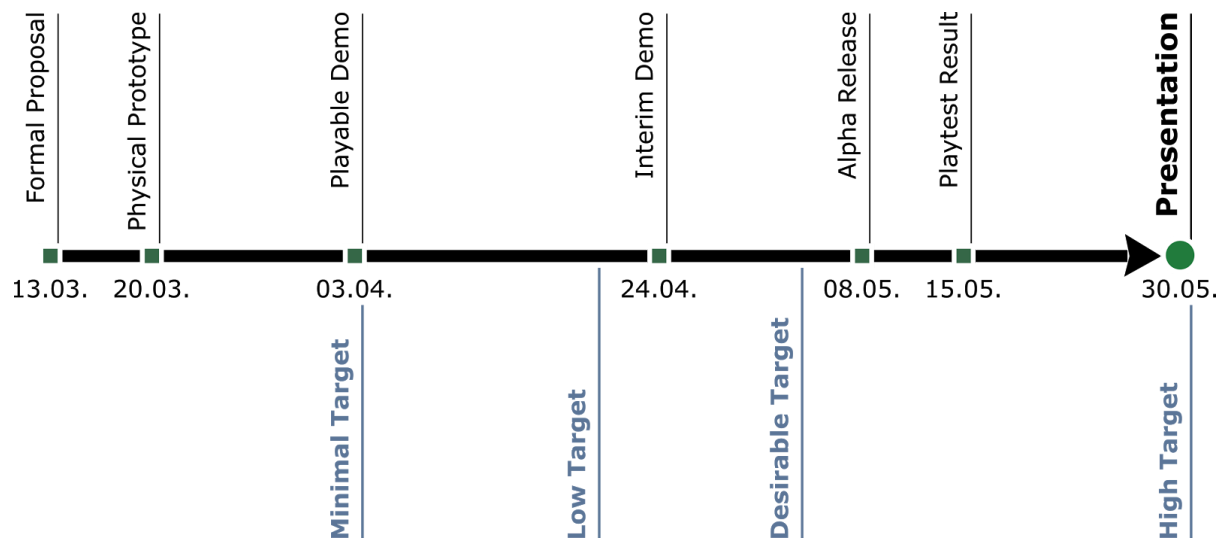
Assets: a second map in a different environment, animal sprites, trap sprites

Sound: additional sounds

Extras

Riddles (e.g. in a ruin) to reach more valuable treasures, dynamic arena (e.g. flooding or fire), adaptive music system, character voicing for certain situations

Timeline



Gantt Chart

https://docs.google.com/spreadsheets/d/1djJuGmrCx6J4rrYNVmk8c8VWfb6lrbJKONT4kw_rvxY/edit#gid=0

Stages/Milestone	Start	End	Required	Assigned
Minimal Target/Playable Demo (due: 03.04)				
Define asset styles	07.03.17	27.03.17	8h	Irina
Create physical/digital gameplay prototype	07.03.17	20.03.17	8h	David
Evaluate gameplay based on prototypes	18.03.17	20.03.17	2h per person	All
Define level design rules	21.03.17	28.03.17	2h per person	All
Static asset creation	28.03.17	03.04.17	8h per person	Irina, David
Level asset creation	28.03.17	24.04.17	8h per person	Irina, David
Character assets creation	28.03.17	24.04.17	8h per person	Irina, David
Character animation (assets)	04.03.17	01.05.17	8h per person	Irina, David
Game architecture	14.03.17	21.03.17	4h	Ribin, Tobias, Christoph (ETH)
Asset import and display	14.03.17	21.03.17	4h	ETH
Loading game map/environment architecture	14.03.17	21.03.17	2h	ETH

Basic camera system	14.03.17	21.03.17	2h	ETH
Simple physics management	14.03.17	21.03.17	3h	ETH
Simple menu and HUD system	21.03.17	28.03.17	6h	ETH
Basic character (input, movement)	21.03.17	03.04.17	10h	ETH
Sprite animation (code)	21.03.17	28.03.17	4h	ETH
Simple vegetation system	21.03.17	28.03.17	4h	ETH
Simple item system	21.03.17	28.03.17	4h	ETH
Basic ability/interaction system	28.03.17	03.04.17	8h	ETH
Low Target/Interim Demo (due: 24.04)				
Advanced collision detection	03.04.17	10.04.17	10h	Ribin
Advanced menu and HUD design	03.04.17	10.04.17	10h	ETH
Simple weapons and trap system	03.04.17	10.04.17	10h	Christoph
More diverse character movement	10.04.17	17.04.17	10h	Ribin
Advanced treasure system	10.04.17	17.04.17	10h	Christoph
Simple soundeffect + sounddesign	10.04.17	17.04.17	10h	Tobias
Advanced environment	17.04.17	24.04.17	10h	Ribin
Multiple characters	17.04.17	24.04.17	10h	Tobias
Advanced camera system	17.04.17	24.04.17	10h	Christoph
Desirable Target/Alpha Release (due: 08.05)				
Advanced item system	24.04.17	01.05.17	8h	Christoph
Advanced trap system	01.05.17	08.05.17	6h	Christoph
Simple AI (wild animals)	24.04.17	08.05.17	8h	ETH
Character skill system	01.05.17	08.05.17	10h	Tobias
Simple particle system: fire/rain/fog effect)	24.04.17	08.05.17	20h	ETH
Advanced soundeffect/ background music	24.04.17	08.05.17	8h	Tobias
Organize playtesting session	02.05.17	08.05.17	1h per person	All

Playtesting session	02.05.17	15.05.17	3h per person	All
Crafting system	24.04.17	01.05.17	10h	Tobias
High Target/Presentation (due: 30.05)				
Procedural map setup/generation	08.05.17	30.05.17	24h	ETH
Advanced weapons crafting system	08.05.17	30.05.17	12h	ETH
Advanced particle effects/ dynamic Env.	08.05.17	30.05.17	24h	ETH
Advanced AI system	08.05.17	30.05.17	10h	Ribin
Evaluate playtesting	13.05.17	15.05.17	2h per person	All
Refine gameplay	16.05.17	29.05.17	2h per person	All
Refine assets	16.05.17	29.05.17	6h per person	Irina, David
Prepare presentation	23.05.17	29.05.17	4h per person	All
Code improvement	08.05.17	30.05.17	infinite	ETH

Roles

Producer: Tobias Krebs

Co-Producer: All

Programmer: Ribin Chalumattu, Tobias Krebs, Christoph Maurhofer

Game-Designer: All

Asset Creation: Irina Huwiler, David Krummenacher, (Ribin Chalumattu, Tobias Krebs, Christoph Maurhofer)

Sound Designer: Tobias Krebs

Quality Assurance: all and external

Assessment

The game's core mechanics are designed to make it a competitive experience with the potential of provoking hot tempers in a peaceful manner. The controls are kept simple enough to allow casual players to quickly understand and play the game. By keeping the matches short and intense, larger groups than four players can easily take turns on the available player slots.

The setting of the game is rather neutral and no specific group of people is targeted. The 2D art-style surely is an eye-catcher and a feature which may also please and attract people not playing many games.

Colonizing Bastards! - Prototyping

Ribin Chalumattu, Irina Huwiler, Tobias Krebs, David Krummenacher, Christoph Maurhofer

Physical Prototype

With our physical prototype, we tried to simulate the mechanics of our game accurately. Even though fun is a priority, the development of this prototype is of limited use, when the rules, environment or the experience differ too much of what we are trying to achieve with our final game. For this, we created the following board game:

Rules

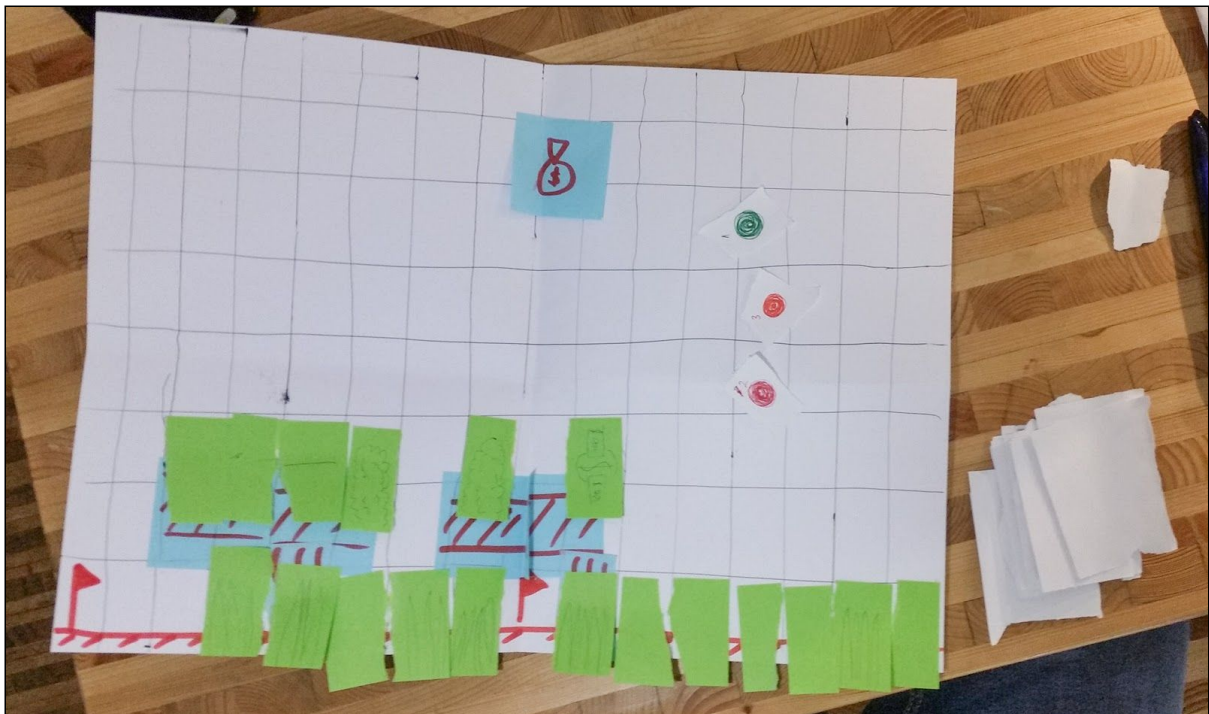
Goal

The players must find as many treasures as possible in the game environment and secure them by returning them to the base camp.

Gameboard

The game environment is a 2D-grid in which certain tiles on different height-levels are solid, so that they can be walked upon. As it represents the jungle, those different platforms can be reached by climbing up lianes, connecting them. The starting point for all players is placed in a central spot. This base camp is also the place, where found treasures have to be delivered.

The whole environment is overgrown by plants (green post-its), hiding the treasures beneath them (blue post-its).



Setup

Each of the players has a set of cards: two movement-cards and three action-cards.

Movement: Moving **one/two** fields in an arbitrary direction. One of each per player.

Action: Describe an action taken by the player at his/her current position. If the action can't be executed (e.g. no player to hit), it is discarded. Each player receives one of the following:

Cut: Cuts down the vegetation at the current position (only overgrowth, not the lianes).

Collect: The player grabs the treasure (if available) at the current position.

Hit: The executing player hits another player in the same tile, who drops the treasure he/she is carrying (if any).

Every player's pawn is placed at the base camp at the beginning of the game. The treasures are randomly distributed in the environment beneath the shrubbery (by a non-participating person, if possible). All tiles (or as many as possible) are covered with overgrowth.

Gameplay

The game is played in rounds. At the beginning of every round, each player chooses one or two movement cards and at most one action card. A player can decide to play fewer cards or none at all (player has to announce, which turn he/she skips). The chosen cards are placed on a stack (first card to play on top) face down in front of each player. Starting with a previously defined player, each card of each player's stack is uncovered and executed in turns. So the first player executes his/her first card only, then the second player (to the right of the first), and so on. When all cards have been played, a new round starts.

The game ends when all treasures are found or a player-defined goal/limit has been achieved/reached. The player who collected and returned most treasures to the base camp wins.

Gameplay Experience



We played this game with three players and one person setting it up and guiding it. The simulation of the realtime interaction with the action cards, as well as the core mechanics of the game do work. The rules are simple and with a little practice, the game can be played in a much faster pace. Even though some situations in the game are fun, the main problem is, that it is too slow. In some way, it's simply simulating the game we have planned, by computing everything by hand. As the fast gameplay

defines the core our planned game experience and most definitely makes out most of the challenge and fun, it seems to us, that a board game is not really fitting to prototype our ideas. With physical games being more organized and in some or another way turn-based, it's hard to provide the perks of virtual games, namely the possibilities for fast and interactive gameplay.

Results

As already described in the previous section, we were able to reproduce the core mechanics of the game, but failed to capture the majority of the fun, resulting from the fast paced interaction between the players. Still, the prototype was a possibility to discover some details in our idea, that have not explicitly been discussed before and generally allowed the coarse testing and reshaping of some mechanics. It gave an overview that also showed problems of certain configurations of our game.

An example is the placement of the player's base camps. Either each player has his/her own base (less conflict, but that's not what we want), or there is only one base camp for all (creates conflict zone, but enables players to barricade and steal treasures gathered by others). And if there is only one base, where should it be placed.

Digital Prototype

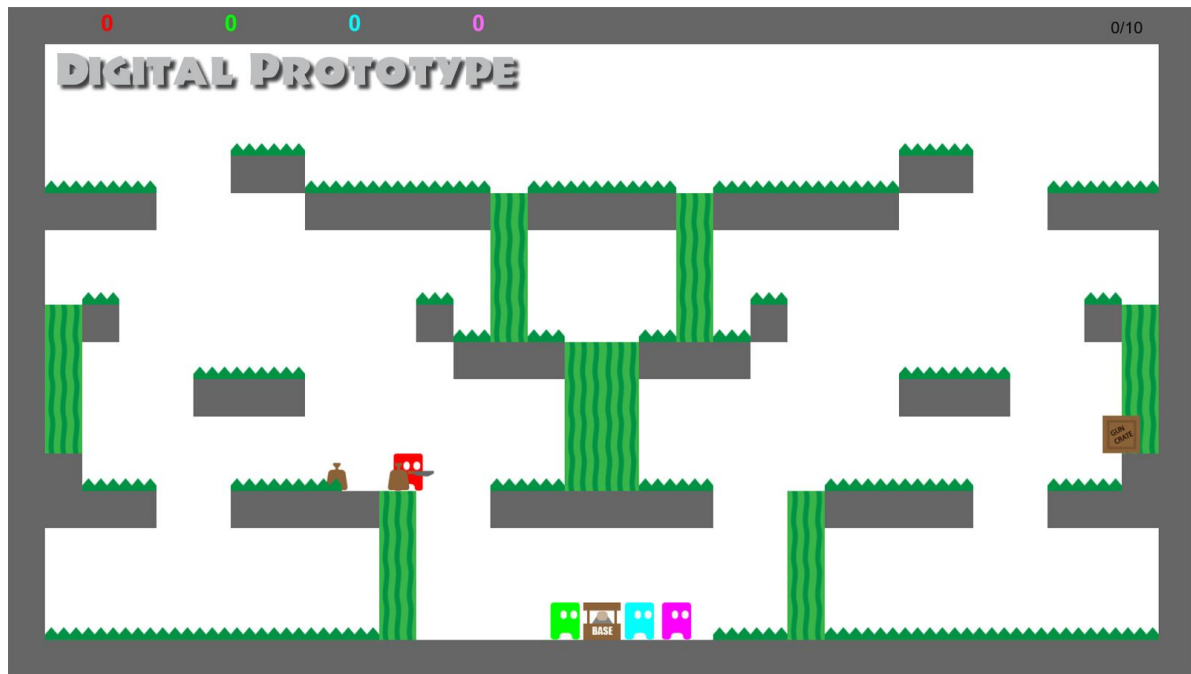
Due to the perceived limitations of the board game above, we implemented a digital prototype. This digital version allowed us to simulate the core gameplay mechanics in a realtime environment. Furthermore potential issues and game experiences could be observed early on, before the actual implementation in MonoGame.

Features

The digital prototype was implemented using *Construct 2*, a game engine that allows for a fast development of 2D games. In the picture below we depicted a sample version of the game.

The main mechanics of the prototype are as follows:

- Goal: The main aim is again for each player to collect as many treasures as possible and bring them safely to the base camp. Furthermore the gameplay should enforce confrontation between the players when wrangling for loot.
- Number of players: 2 to 4 players can play simultaneously.
- Player abilities:
 - Movement in all possible directions.
 - Cut down vegetation using a machete. Find loot or weapons hidden within the undergrowth.
 - Find guns and stun other players with it.
 - Carry and steal loot from other players.
 - Player can defend himself while carrying loot by using weapons.



Gameplay Experience

The gameplay experience was positive, since the mechanics work and are fun to play with in real time. It was especially satisfying to stun and steal loot from other players (To be a real bastard).

We could also test the game with players, with no previous knowledge of the game. One problem was, that the players were confused about what to do in the beginning and how the controls worked. But after a couple of tries, they grasped the main concepts quickly and the multiplayer aspect of the game allowed for variety of different game situations.

Results and Possibilities

From the digital prototype we could make several observations, which could help us to make the gameplay even more intuitive and fun. Some of the main things we noticed were the following:

- The movement of the player feels good with somewhat exaggerated physics.
- The stun time of a player is perceived too long. Vary stun time depending on the type of weapon
- Bullet speed of the gun was perceived to be slow. Increase speed slightly/vary speed for different weapons
- A player could just barricade himself at the base camp and try to steal the loot from other players. Allow the player carrying loot to defend himself using weapons. But this camping behaviour at the base camp helps the game's competitive feel.
- Current level size was too big for 2 players. Change the level size according the amount of players/ have a dynamic level size with varying amount of undergrowth, treasures and platforms
- We observed different parts of the level, which were interesting for level design. A player could have different jumping heights based on the carried loot and undergrowth. This allows for different movement strategies for the player which could make the gameplay even more interesting.
- Players were initially confused about the main goal and controls of the game. Mechanics and winning conditions should be communicated early on by using intuitive level design(Make base camp more prominent, make cutable undergrowth more visible and maybe add instructional screens)

Colonizing Bastards! - Interim Report

Ribin Chalumattu, Irina Huwiler, Tobias Krebs, David Krummenacher, Christoph Maurhofer

State of affairs

Having finished the goals of layer two before the scheduled date, the features of layer three are now implemented. The base game stands and the corner stone can now be improved and be refined. Alongside ensuring the proper functioning of the core, the technical achievements are now tackled more directly. Work on the particle system for enhanced visual quality and possibly some game influencing elements like fire has started and the way is paved for an AI for the wildlife (hindering the player from collecting treasure) or for experimenting with procedurally generating parts of the game environment. As these features have a large impact on the gaming experience, their specific implementations are still the subject of discussions.

Challenges and Decisions in Game Design

During the development, our game ideas are constantly reassessed. For example, the following aspect have been discussed:

A dynamic merchant positioning (base to deliver the treasures to) to prevent camping, as well as the addition of a tiger as a dynamic element, as it chases the players and stuns them.

Also, the implications of a procedural generation of parts of the level is difficult to estimate. Instead of a straightforward implementation, the process will be more iterative to experiment with different approaches. If none works satisfactorily, as single, fully designed environment might still be the better choice.

Coding

Base Engine

Progress

Most of the planned aspects for the game's foundation have been implemented. The proper distribution of different aspects such as the menu and the game itself and a solid foundation for the logical and the graphical aspects have been created, to allow much simpler extensions to the game-logic or additional features. The work on these parts are mostly concluded. Still, sometimes some changes to this basic architecture are necessary to fix some unforeseen problems or necessities of new implementations.

Challenges

Polygonal Collision Detection

Since our levels consist mostly of uneven ground, we decided to implement a polygonal collision detection. Performing an accurate collision detection has proven to be harder than expected. During the early phases of the implementation we encountered various kinds of glitches, since not all of the special cases have been taken into account. One of the main problem was, that the character was moving very slowly while walking on slopes. This problem was circumvented by performing a raycast from the player's hitbox to the ground, which allowed us to adjust the speed of the character according to the steepness of the slope.

Smooth Character Movement

The speed of the character varies depending on the whether he/she is walking on undergrowth, carrying treasure or doing both. Adjusting those different speeds, such that the movement doesn't feel too slow or too fast was also difficult. To partially solve this problem we linearly blended the old speed with the new one, to get smoother transitions. We are still experimenting with those speed parameters to get a good movement

Game Implementation

Progress

The basic mechanics of the game are implemented. Characters can pick up treasure, shoot others, get stunned and lose the treasure carried when getting hit, climb vines and cut undergrowth to reveal treasure potentially hidden therein. By delivering treasure to the base, players earn points, shown in a hud at the top of the screen. There is no time limit or point limit imposed yet.

By having implemented an event-driven framework, we can easily implement script-like events, which are triggered by collision or specific player-driven actions. This allows us to quickly change mechanics and reuse certain events (such as the treasure-drop event invoked when cutting grass or getting hit by a bullet while carrying treasure). Using this technique, we will be able to quickly implement future features such as the tiger or getting hit on touching fire.

Challenges

The main challenge was refining movement and implementing a dynamic animation system that allows for certain layers to be hidden or shown individually (such as the backpack shown when treasure is collected). Integrating everything in the architecture has proven to be easy, as it proved to be very versatile.

Assets

Characters

Progress

All player characters have been designed and animated. They were exported into spritesheets and corresponding xml data was created to describe the animation. We use different layers to animate the arms independently from the rest of the body (eg. shooting while running). The characters had to be bulkier to increase their presence on the screen. A tiger that attacks the players is still missing.



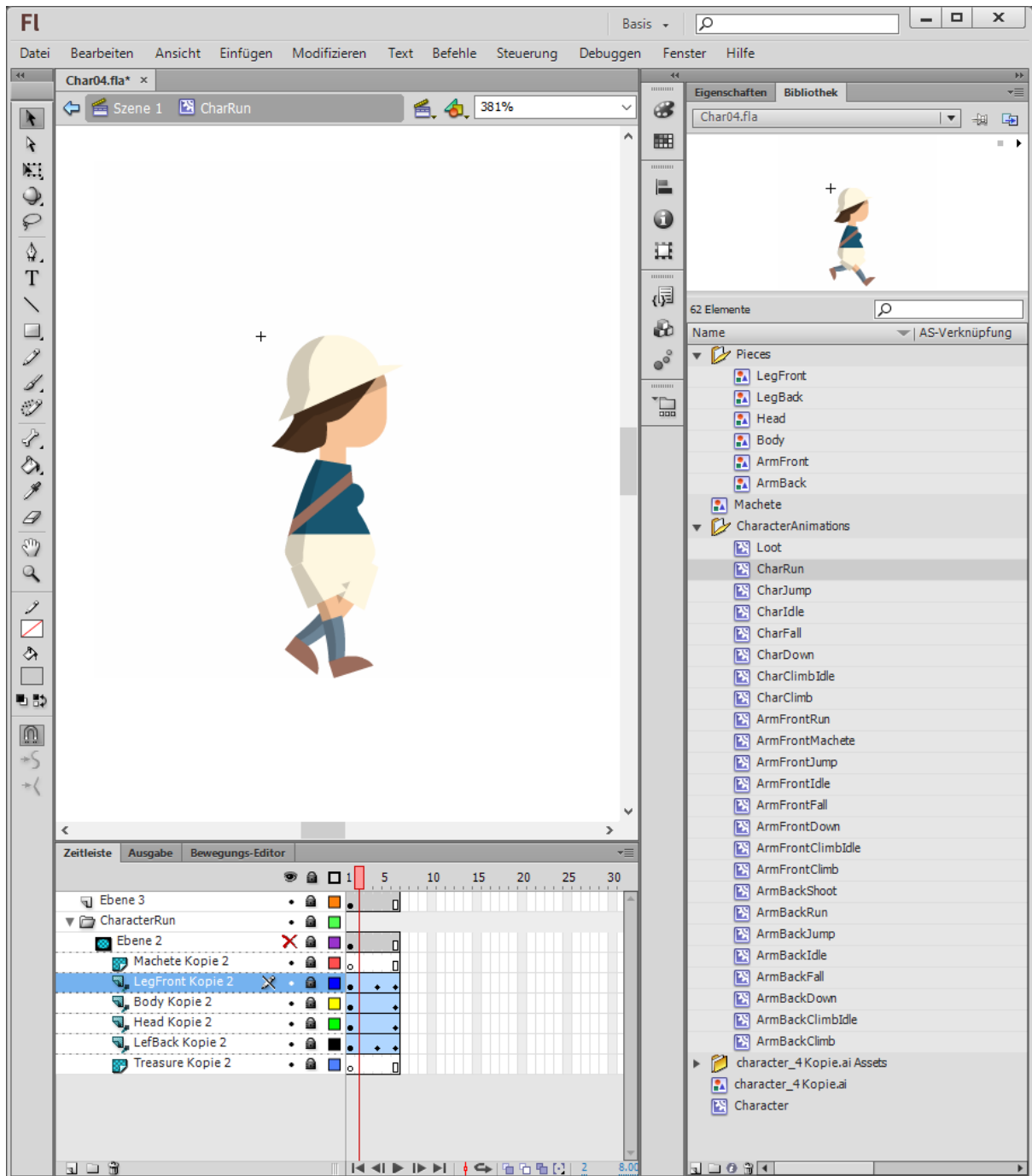
Challenges

The main challenge was to create a suitable asset production pipeline particularly for the generation of animation spritesheets.

We decided to use Adobe Illustrator to create layered vector graphics of the characters in a basic stance. The choice was made so we would suit our art style and could easily scale the characters later on and no detail would be lost. The layers were chosen in a way that we could easily import them into our animation software and were ready to animate.



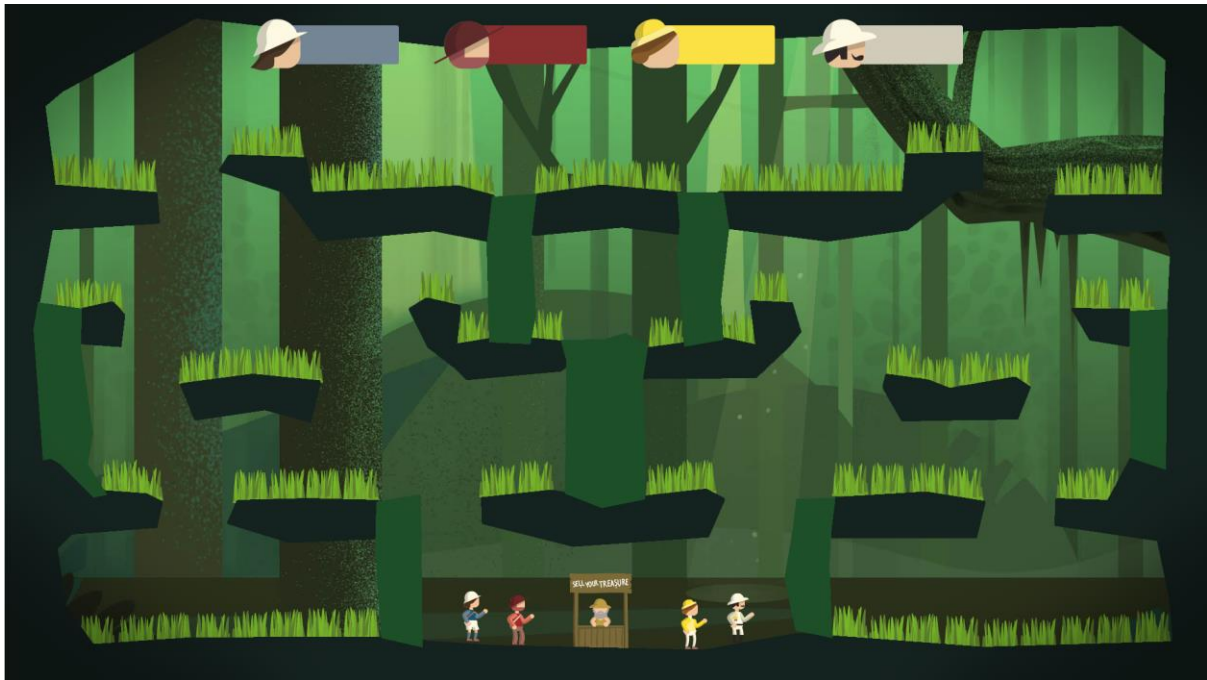
To animate the characters we first made some tests within Adobe After Effects but the spritesheet export was too limited even when using Plugins like SheetaH. We ended up using Adobe Flash to animate the characters. Since we had to manually edit the animation xml data, the animations of all characters are the same so we had to create the data just once. We used symbols within Flash to easily swap the animated body parts to another characters.



Level

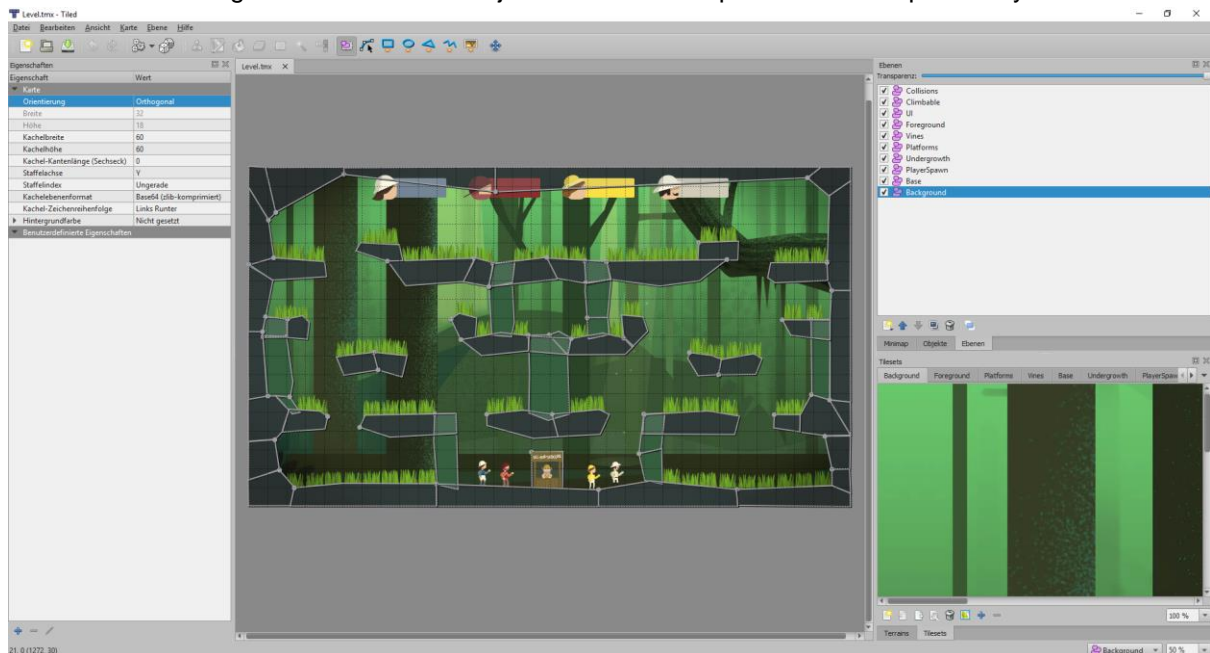
Progress

We created a background that now suits our defined art style. We added a simple GUI to display the score, some new grass sprites and a merchant for the home base. The merchant still needs to be animated and position changes need to be implemented to prevent players from camping next to him. The level assets and a display for the remaining amount of treasures are still missing. Grass animations and particle sprites for walking and cutting grass are missing as well. The level was put together within Tiled editor.



Challenges

The challenge inside the level is to connect the interactable are with the background visually without lower the readability for the players. Another challenge comes from the chosen level editor that does not allow for binding collision masks to objects so we had to put them in a separate layer.



Additional Screens

Progress

The menu screen is finished and allows players to join and start the game. The winning screen is still missing.



Challenges

The main challenge was to create the amount of interaction to start a game at a minimum. We chose to allow players to join the game by pressing a button and starting the game when 2 or more players joined. The assets for the menu screen were partially created in Illustrator and composed inside Photoshop. The composition was later brought into the Tiled editor for loading it into the game.



Sound

Progress

We implemented a basic sound engine, accessing a library consisting of sound-effects and music. The current implementation is basic and does not offer more fine-grained control, but should be enough for our project since features such as 3D-positioning do not make a lot of sense to spend a lot of time on. As for the source of our audio assets, we are using royalty free assets from the internet, while working on our custom soundtrack. We have planned to record sound-effects ourselves later on in the future, since we have some background on sound design and recording. Once these assets are finished to our satisfaction, we embed them in our project.

Challenges

Generating high-quality assets is a task not to be underestimated. Equalization, compression and volume normalization will be the main challenges for this task. Should our current sound engine not be powerful enough, we will have to improve it, maybe using the XACT framework.

Colonizing Bastards! - Alpha Release Report

Ribin Chalumattu, Irina Huwiler, Tobias Krebs, David Krummenacher, Christoph Maurhofer

Programming

Base Engine

Challenges

Collision Detection

Since we did not use any third-party library and implemented the collision detection system from scratch, we have encountered various glitches during the initial playtesting sessions. Further improvements had to be made to handle special cases and improve the character movement while climbing the liana. Some issues could be fixed by rounding the bounding box of the character and the bounding polygons of the collidable objects. Even though collision detection has proven to be tough nut to crack, it was educationally very insightful to implement it ourselves.

Game Implementation

Progress

Particle System

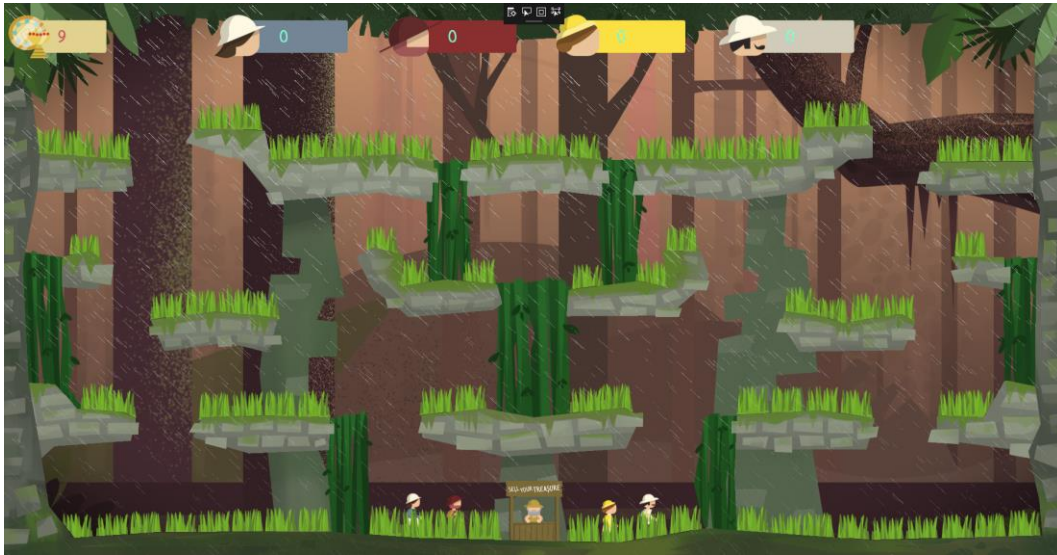
Several kinds of particle systems have been implemented in order to further improve the visual style and atmosphere of the game. To simulate a multitude of different effects we first implemented basic particle system framework on which we could build upon. Below we list some of the effects implemented:



- Footstep dust:



- Grass cutting:



- Rain:

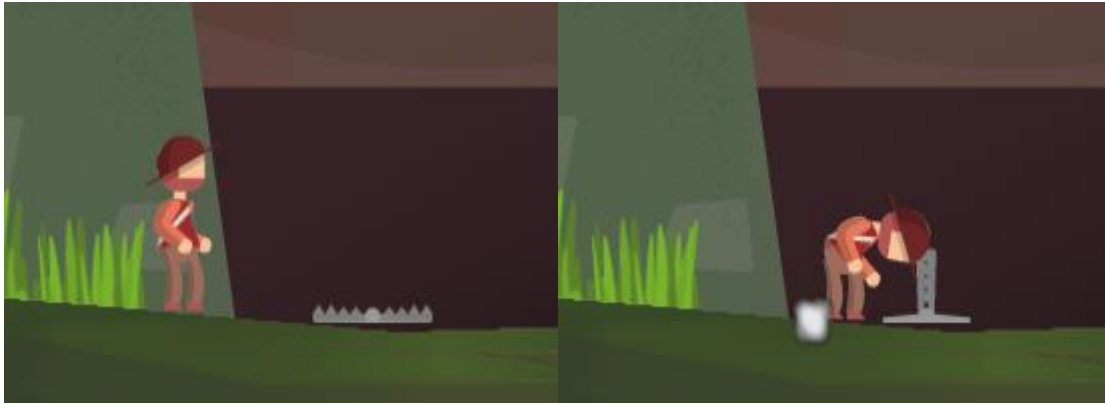
To make the game world more immersive a weather system was implemented which uses the rain particle effect mentioned above and additionally lightning bolts. The weather system then randomly creates the rain effect. The lightning bolts are created also at random intervals during the rain.



- Lightning:

Trap System

Traps can be placed by holding the collect-button for 3 seconds. This will probably change in the future to be a separate control. At the moment, all players have one trap at the beginning of the game and cannot actuate their own traps. We will probably allow players to pick up activated traps in the future.



Touching a trap laid by another player stuns you and makes you drop any treasure you're currently carrying. We will have to test different stun durations in order to give opponents enough time to steal the treasure dropped while not frustrating the trapped player by too extreme durations.

AI System

As an adversary to the players, we are currently implementing an indigenous character (instead of the tiger from our proposal), that is controlled by an AI. Those characters are also spawned by an event, when a player cuts down some grass. They then start to hunt the closest players, stunning them with a hit or by shooting them with an arrow from a blow tube.

At the moment, we are finishing the implementation of a navigational system, that allows the AI to do pathfinding across the whole map. Still, the system does not fully work at the moment and the indigenous people are therefore not activated for the alpha release. They will be added as fast as possible.

Ammo Crates

To hinder players from deadlocking by stunning each other repeatedly, we added ammo crates which have to be discovered in the undergrowth and be picked up to gain ammo. At the moment, each crate supplies players with three shots. This also imposes a strategic conflict for players as to decide when to use the bullets.



Challenges

Particle System

The main challenge for the particle system was to find the correct parameters (like scale, velocity, spawn points etc.), such that the resulting effects looked visually pleasing. Further fine tuning is still needed to improve upon the existing effects.

AI

The implementation of a system, that automatically builds a navigation graph from the given environment and the creation of an AI, which is able to move thereon, is not an easy task. The main

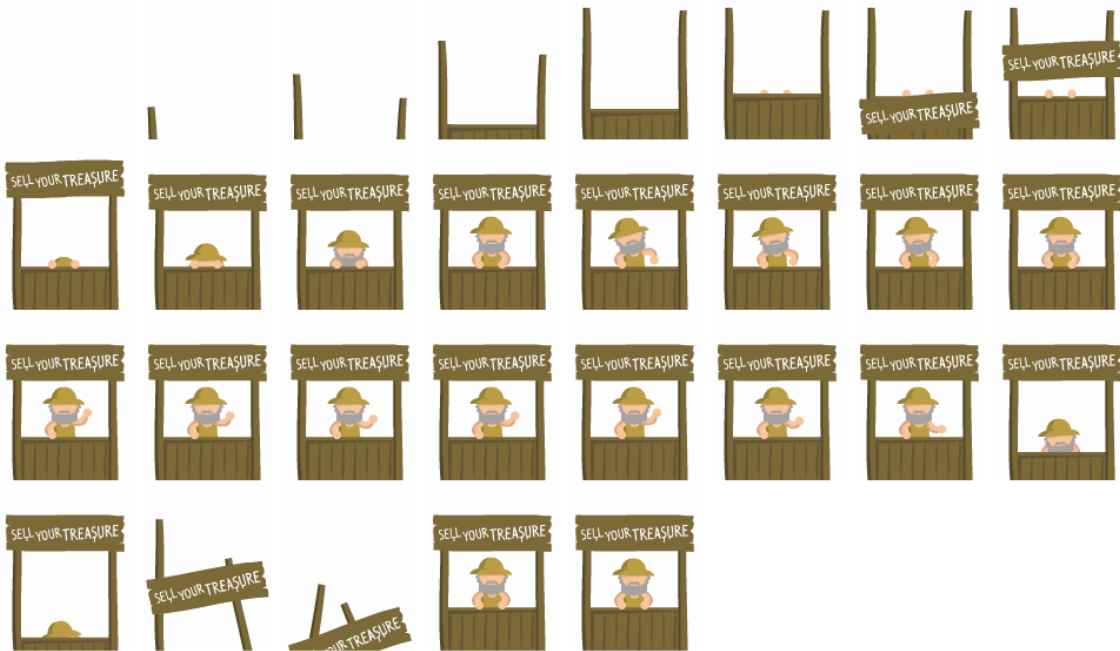
challenges are to compute reliable information, find an efficient path to arbitrary targets, having a character with fluid movement and a plausible behavior. As this task takes more time than anticipated (respectively, the goal has been extended in the scope), we are currently reducing the priority of the procedural level generation and are concentrating on the AI at the moment.

Assets

Characters

Progress

The shopkeepers appearance and disappearance was animated to enable dynamic positioning. But the idea was scrapped and only the animations for the idle state and receiving treasure are used.





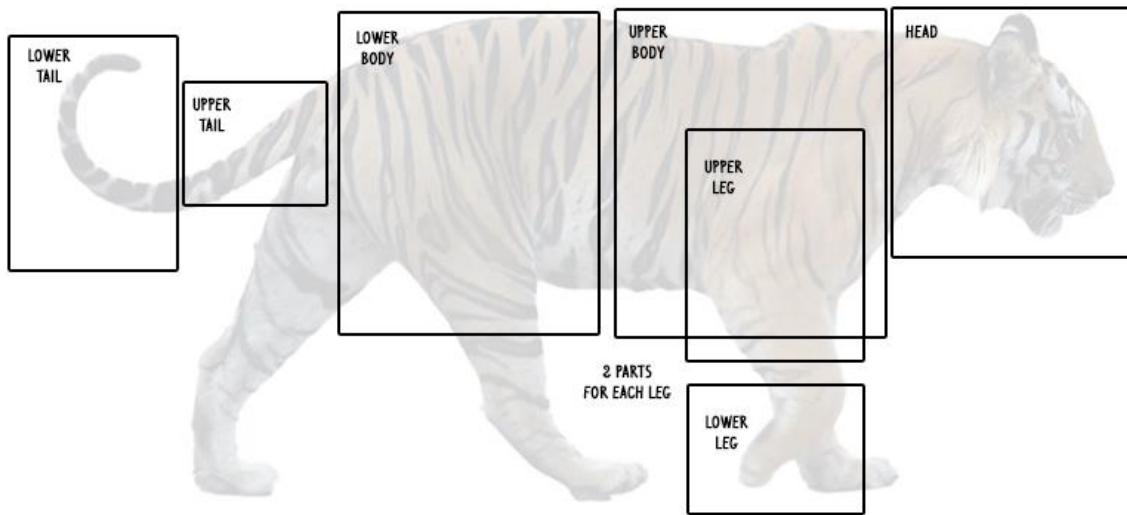
We finished the assets for the indigenous person. The animations are based on the same system the player characters use.

The snake as a dynamic element was animated.



Challenges

The proposed tiger as the dynamic challenge turned out to require too much articulation for current animation approach to work sufficiently. So we decided to introduce an indigenous person instead.



Level

Progress

The level assets were finalized and an additional display for the remaining treasure was added.



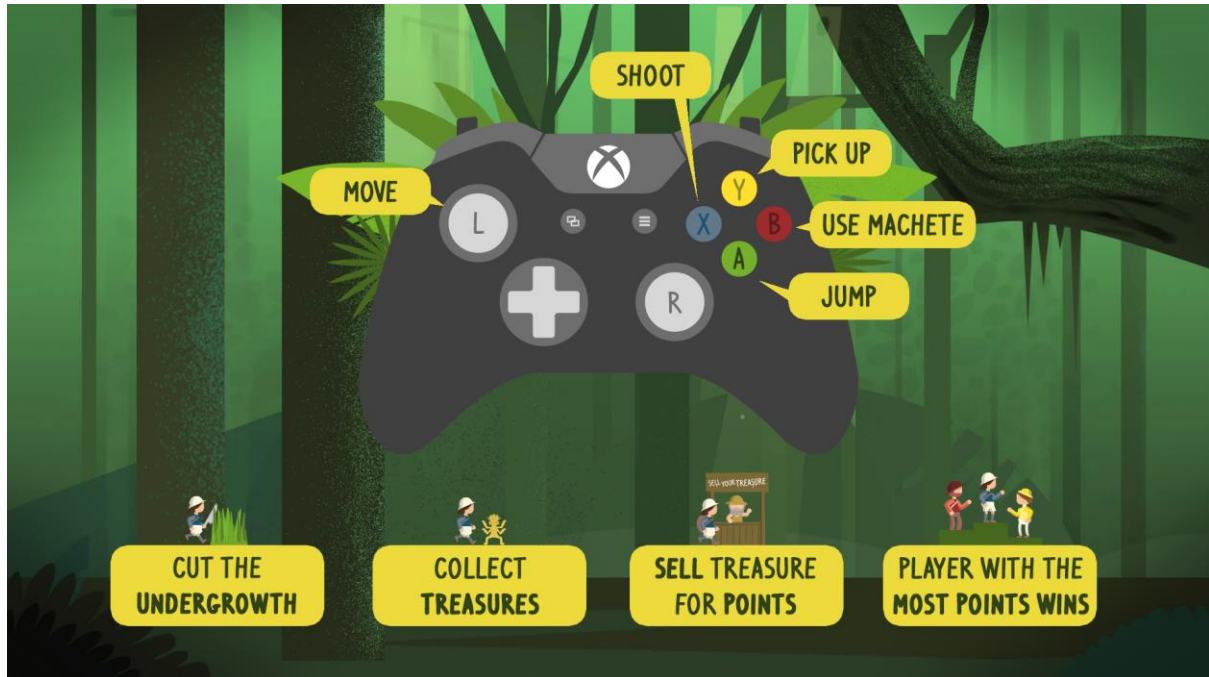
Challenges

The main challenge was to visually separate the background from the playing area. We chose to use color contrast to achieve it.

Additional Screens

Progress

A winning screen and an instruction screen were made. The control instructions still have to be changed according to our final control scheme.



Challenges

There were no extraordinary challenges. You just have to do it.

Sound

Progress

We added sounds for the trap activating, a winning screen fanfare, rain and lightning. In order to get a continuous sound effect for the rain, we played around with crossfading. Fortunately, since rain contains a lot of random noise, this didn't prove to be too challenging. We are still working on a custom soundtrack.

Challenges

The challenges did not change since the last report.

Colonizing Bastards! - Playtesting Report

Ribin Chalumattu, Irina Huwiler, Tobias Krebs, David Krummenacher, Christoph Maurhofer

Playtesting Results from 09.05.17

Participants Feedback

Around 6-7 people participated at the playtesting held at the ZHdK from which we could gather some useful insights from the participants. Some of the changes proposed were the following:

- Increase the pacing of the game (increase the running speed of the character, increase the speed while climbing etc.)
- Allow the character to fall down when climbing down (by button press)
- Allow the character to climb out of the liane more easily(when climbing up)
- Make the character movement feel more natural/ smooth
- Allow the character to decelerate faster when it comes in contact with foliage
- Gather loot/treasure in the game automatically (instead of button press)
- Make the foliage burn, if it is hit by lightning
- Make shooting bullets feel more dramatic (by using particle effects to show smoke coming out of the gun, fire effect behind the bullets etc.)

Some of the suggestions made, were features we already wanted to implement, but didn't get the time to do so:

- Indicate the ammo count for each character
- Limit the total ammo each player can carry (max. 3 ammo per player)
- Allow the game to be paused and restarted during gameplay
- Decrease the number of treasures depending on the amount of players (to further increase conflict between the players)
- Show the final score of the players in the winning screen, after the game has ended.

Changes Made

From the suggestions above, we decided to make the following changes to the game:

- Increase the speed of the character when climbing
- Increase the speed of the character slightly while running
- Make some of the foliages move to indicate a possible treasure or a snake, that can stun the player
- Allow the character to fall down while climbing by pressing a button (increases the pacing of the game further)
- Make the foliage burn if it is struck by lightning (should not happen too often, so that the gameplay is not disturbed)
- Decelerate the character faster, when it it comes into contact with the foliage
- Disable cutting foliage if the character is carrying treasure (player needs to strategically cut the foliage beforehand)

Playtesting Results from 13.05.17

A second playtesting took place with some friends (all of them being gamers). Additionally the more positive comments, we received the following feedback:

The more important feedback:

- show points and ranking for each player in the winning screen
- optimizing collidable layer to make movement more fluid/expected
- jump in the lower left corner too difficult
- leaving climbing vines on top is difficult

Points with lower priority:

- animation or progress bar indicating you're laying a trap (and making sure you're not moving and not stunned while doing so)
- jumping or dropping while climbing
- enable players to move diagonally while climbing
- add fall damage and lower falling speed slightly
- adjust total count of treasures hidden according to player count
- floaty controls (especially on uppermost platforms while carrying loot)
- lower frequency of rain slightly

Bugs noticed during the playtesting session:

- limit ammo count correctly
- fix controls for traps
- exception in hud when some players are not ready (but their controllers connected) and someone hits start
- bullets traverse through players
- cutting animations is not properly executed
- jumping over shopkeeper doesn't trigger sell (wanted?)

Proposed changes in gameplay:

- enable players to hit each other (e.g. to stun)
- multiple traps/picking up traps
- making undergrowth more sturdy (some requiring 2+ hits to be cut)

Changes adapted as a result:

The ammo bug has been fixed and the trap mechanic has been overhauled. Players can now pick up any closed trap (carrying no more than one trap at any time) to re-use. Players are stunned while deploying a trap such that players can see their character is performing an action. The cutting animation has been fixed as well and is now visible.

We tested making some foliage harder to cut but did not end up implementing the change. There is no clear indication currently for players trying to cut grass, and, as a result, gameplay felt more glitched or buggy which is against our intention of optimizing gameflow and feeling.

Because of the tight time constraints we could not test players hitting one another to stun them. We believe that this mechanic would change the core gameplay too much and make players explore in order to gather ammunition less appealing and reward players camping at the shopkeeper even more

rewarding. Especially at the end of the rounds, when only few treasures remain, and exploring is less lucrative. This is not our intention.

We plan on fixing many of the other points of criticism, as far as the remaining time allows it. Additionally we are hard at work to have the artificial intelligence on a game-ready early enough, to be able to test its impact on the gameplay and do the fine-tuning.

Colonizing Bastards! - Final Report

Ribin Chalumattu, Irina Huwiler, Tobias Krebs, David Krummenacher, Christoph Maurhofer

Changes After Alpha Release

Adversarial AI System

In order to give our game story more depth, we introduced indigenous people into the game, who want to protect their treasures against the invading explorers. We implemented an AI system, which controls the movement of the indigenous character. The AI works roughly as follows:

1. Precompute the Navigational Graph, which indicates the connections between each of the platforms. The Graph is computed dynamically, hence it works for any 2D map, which uses polygons for collision. The orange lines in the figure below, indicates the connections between each of the walkable platforms.
2. Place the indigenous character randomly throughout the map behind one of the foliagees. If one of these foliagees is destroyed by a player, the adversarial character will be spawned.
3. If the character has been spawned, it chooses one of the players as a target. Players who carry treasures are targeted with a higher probability. On the other hand, player which are already stunned are avoided.
4. Now the AI system computes the shortest path between the indigenous character and the target player using the aforementioned Navigational Graph. This path is updated each time as the target player changes position. The white line in the following figure indicates the current path computed towards the target player.
5. The indigenous character will remain on the map, until it has acquired a treasure from one of the players.



Burning Foliage

The lightning in the game was mainly used for visual purposes and did not contribute to the gameplay. Hence we added the feature, that the foliagees can be set on fire, if they are hit by lightning. This can reveal potential treasure or ammo crates hidden behind the grass or it could also spawn the indigenous character. This feature adds an additional level of unpredictability to the game.



Experiences Throughout The Course

1. *How well did the initial design ideas materialize into the final game?*

After we settled on our game idea, we had a very clear picture on how the gameplay should work, hence we could incorporate almost all the main ideas from the initial design into our game. The basic gameplay works exactly as we had envisioned it. The only features we didn't implement into our final game was the weapons crafting system and the procedural map generation.

The original idea behind the weapons crafting was that the player could create their own weapons, but since this would unnecessarily slow down the gameplay and one gun was sufficient, we omitted the idea. The procedural map generation was not included in the final project due to the focus on the AI system.

2. *Was there any deviation from the development schedule?*

We implemented all the features according to the schedule and there were only some slight deviations due to the complexity of the collision detection or the AI system, where the implementation took longer than expected.

3. *How did the elements of the project structure affect our progress?*

The different elements were a great help and contributed a lot to the evolution of our game. Especially the development schedule and initial prototyping were essential, since we could create a solid foundation from which we could build upon. Due to the digital prototype, corrections could be made early on in the game. The playtesting sessions were also quite insightful, since it helped to fine-tune parameters of the game and add additional features.

Personal Impression

1. *What was the biggest technical difficulty during the project?*

The AI system and the polygonal based collision detection were the biggest challenges of our project.

2. *What was your impression of working with the theme?*

The theme helped us to find a couple of good ideas for the game. It didn't hinder us in the creative process, since we could all interpret the idea of the jungle our own way.

3. *Do you think the theme enhanced your game, or would you have been happier with total freedom?*

Since we didn't have total freedom it is difficult to predict, whether it would have produced a better game or not.

4. *What would you do differently in your next game project?*

For the development process of the game we wouldn't change anything, since we followed our initial development schedule rather closely. For the next game project we would definitely make again a digital prototype, since it has proven to be very useful to analyse the core game mechanics and receive early feedback.

5. *What was your greatest success during the project?*

Our greatest success was perfecting the AI system. It was really a joy to see the AI character follow the player around anywhere on the map. Thanks to the flexibility of the AI code, we can use it for any new map, without adjusting the existing code base.

6. *Are you happy with the final result of your project?*

The game works well as a multiplayer game and it is fun to compete against other players. Our only regret was that we could not create more maps using procedural map generation, due to the time constraints. Otherwise we are very happy with what we achieved in this short amount of time.

7. *Do you consider the project a success?*

Yes.

8. *To what extend did you meet your project plan and milestones (not at all, partly, mostly, always)?*

Most of the time we could follow the project plan and implement the relevant features rather quickly.

9. *What improvements would you suggest for the course organization? (perhaps in D1 evaluation)?*

The course it teaches us how to develop a game completely from scratch and gives us complete freedom in doing so. From a practical viewpoint we could learn a lot from the GameLab.

From a theoretical perspective we found, that the lectures given during the semester did not contribute very much to the overall game development process, as most of the slides were self explanatory.

10. Did you like using MonoGame?

For 2D game development MonoGame worked like a charm. It supported the basic functionalities needed to implement all of the features we wanted to and content management was especially very easy. On the other hand MonoGame is not very well suited for the development of 3D games, since it is very difficult to get rigged animations to work and it takes a long time to get a basic game running.

Conclusion

In conclusion the gamelab was a great experience, since it taught us a lot things about developing a complete game. We are proud of our final game and maybe we will extend it in the future.