

7. Das Konzept des Prototyps

31 – marci_und_die_mitbewohner

Konsulent:

Kovács Márton

Mitglieder

Seben Domonkos András
Szapula László
Filip Krisztina
Golej Márton Marcell
Visy Tamás

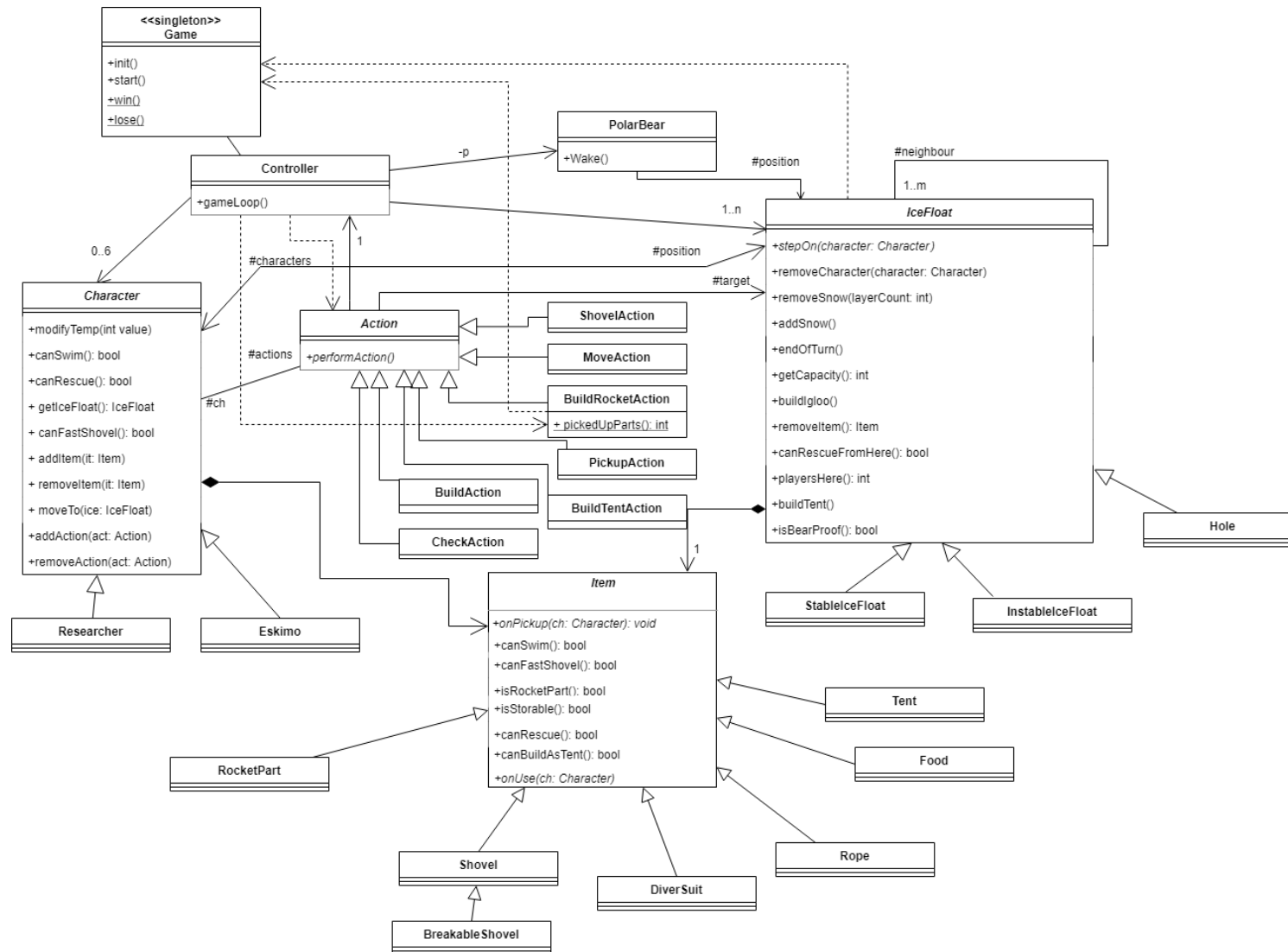
(ETBCNP)
(DJQOM9)
(QE4L0M)
(V1BYVS)
(CTSJ3H)

domi.seben@gmail.com
szapula.laszlo.99@gmail.com
fkriszta997@gmail.com
golejmarci@gmail.com
tamas.visy@gmail.com

7. Das Konzept des Prototyps

7.0 Wirkung der Änderungen auf das Modell

7.0.1 Geändertes Klassendiagramm



7.0.2 Beschreibungen der neuen oder geänderten Methoden

7.0.2.1 Item:

+**abstract void onUse(ch: Character)**: Sagt, was passiert, wenn ein Character diesen Gegenstand benutzt.

+**bool canBuildAsTent()**: Sagt, ob der Character ein Zelt auf seiner Eisscholle mit Hilfe von diesen Item aufstellen kann.

7.0.2.2 IceFloat:

+**abstract void buildTent()**: Baut ein Zelt auf der Eisscholle auf.

+**void addSnow()**: Eine Schicht Schnee wird auf die Eisscholle gelegt. Sie wird von dem Schneesturm gerufen. Wenn ein Zelt oder ein Iglu auf der Eisscholle ist, wird es zerstört, sonst verlieren alle Spieler, die hier stehen eine Einheit von Körpertemperatur.

+**void endOfTurn()**: Am Ende jeder Runde wird es aufgerufen, und dann können zB. die Zelten zerstört werden.

+**bool isBearProof()**: Wenn ein Eisbär auf die Eisscholle tritt, hängt davon ab, ob die Spieler das Spiel verlieren.

7.0.2.3 Controller:

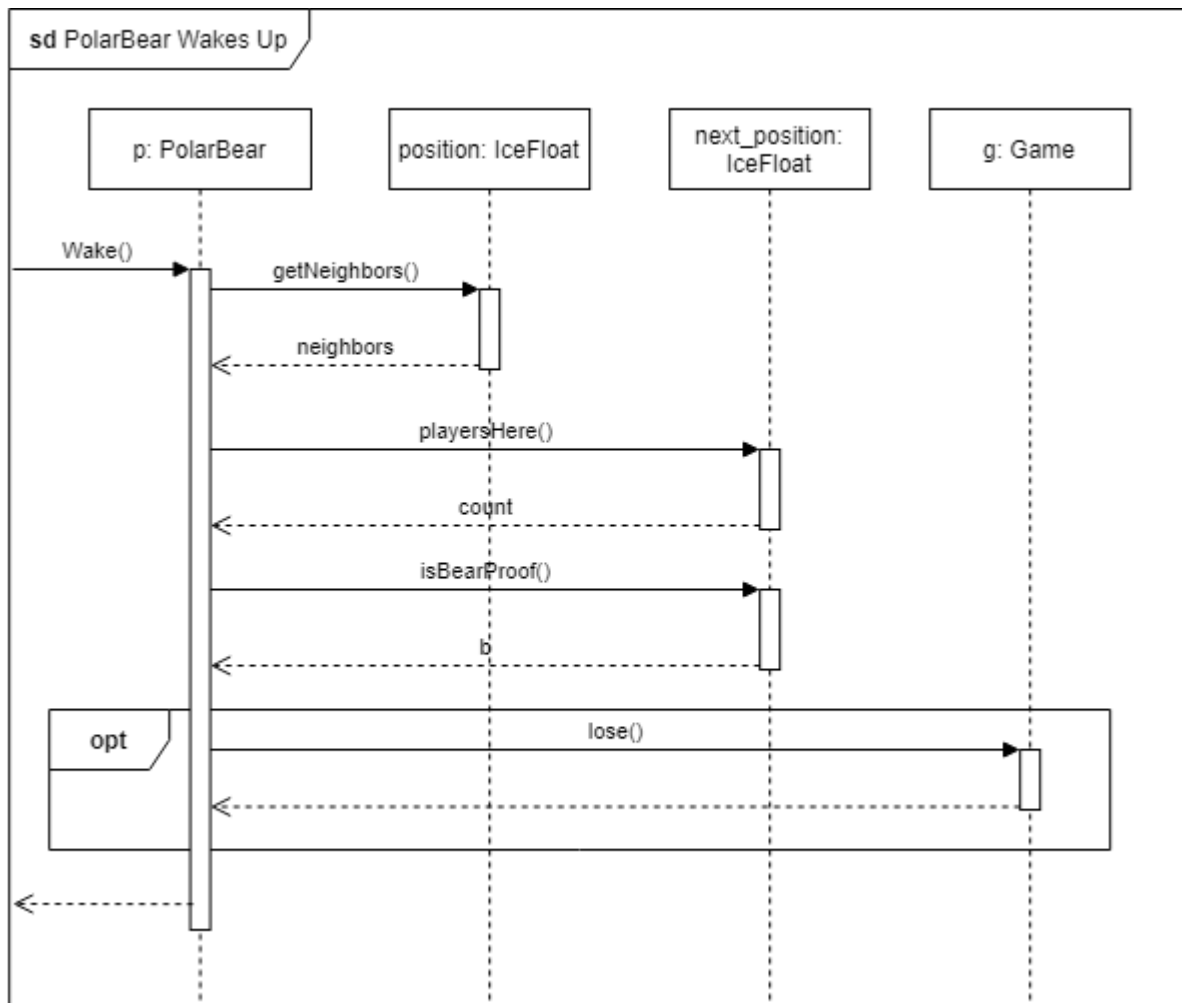
+ **void gameLoop()**: Diese Funktion ist verantwortlich für die Organisation der Zügen von Spielern, und nach jeder Runde ruft einige Funktionen, wie “snowStorm()” oder weckt den Eisbär.

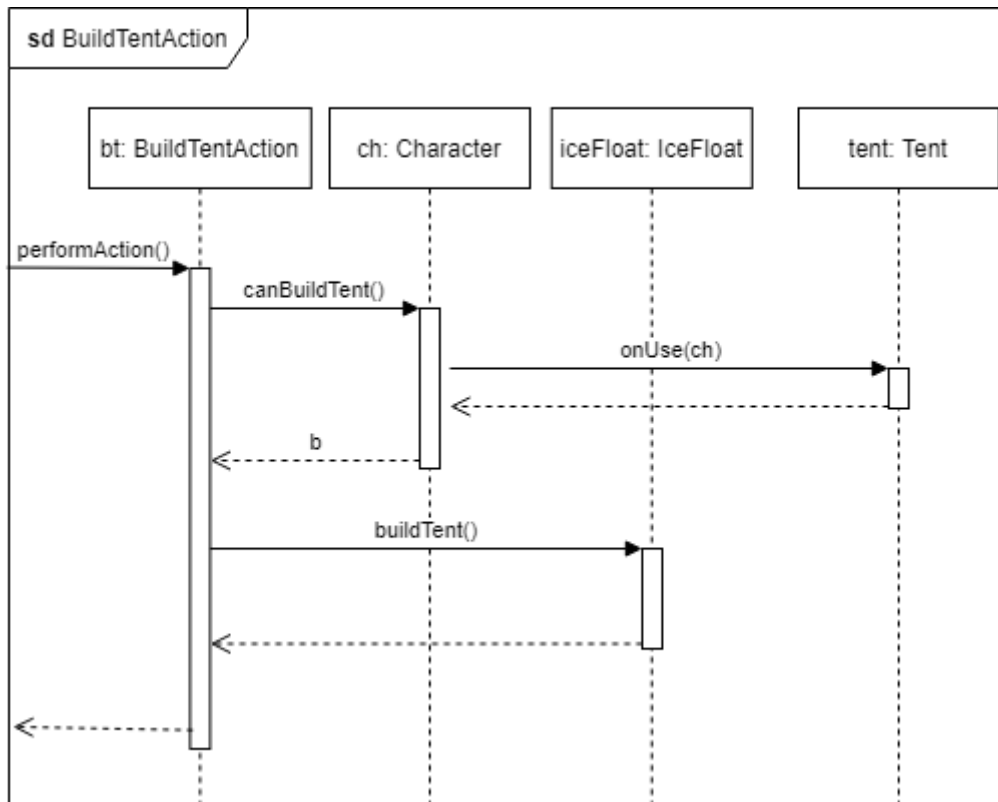
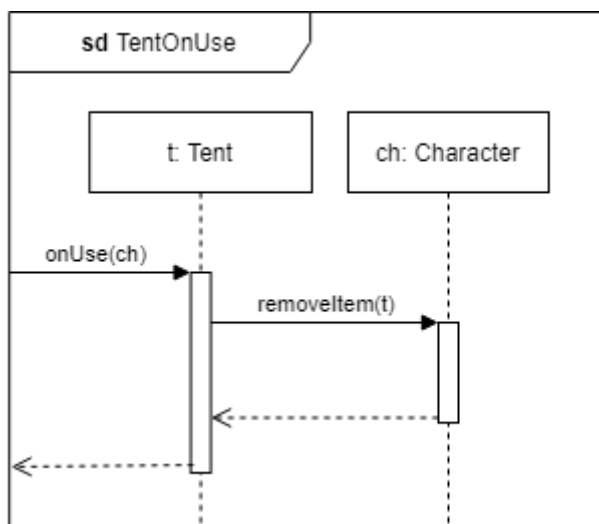
7.0.2.4 PolarBear:

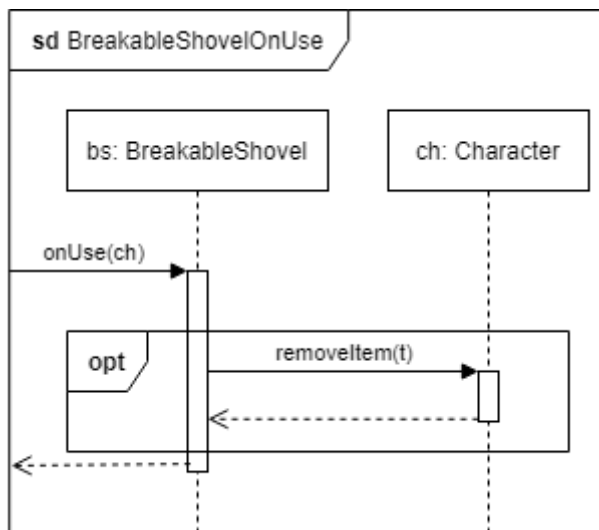
+**void Wake()**: Am Ende jeder Runde erwacht der Eisbär und bewegt sich in eine zufällige Richtung. Falls er mit Spielern auf einer Eisscholle landet, und sie nicht von einem Iglu geschützt sind, frisst er sie auf, und die Spieler sterben. Das Spiel wird verloren.

7.0.3 Sequenzdiagramme

7.0.3.1 PolarBear Wakes Up



7.0.3.2 Build Tent Action**7.0.3.3 Tent On Use**

7.0.3.4 BreakableShovel On Use

7.1 Definition der Schnittstelle des Prototyp

7.1.1 Allgemeine Beschreibung der Schnittstelle

Man hat zwei Möglichkeiten um das Spiel vorzubereiten. Man kann die Eisschollen, die Charakter und die Gegenstände durch Befehle in der Kommandozeile erzeugen, oder durch das Einlesen eines XML Dateies. Diese XML Datei kann auch die Schritte der Testfällen enthalten. Die Ausgang wird in einen einfachen Textdatei ausgeschrieben, falls diese Datei in der Kommandozeile gegeben wurde. Die Syntax der Testdatei sieht so aus:

7.1.2 Eingangssprache

loadTest <inputFile>

Beschreibung: Mit diesem Befehl kann man ein Testdatei einladen. Falls es kein <test> Tag im Datei gibt, wird nur der Anfangskonfiguration durchgeführt.

Optionen: Man muss den Namen der Eingangsdatei eingeben.

generateField <n> <m> <f11><f12>...<f1n> . . <fm1><fm2>...<fmn>

Beschreibung: Mit diesem Befehl kann man ein Eisfeld generieren. Falls es kein Parameter angegeben wurde, wird es random.

Optionen: <n> Anzahl der Säulen
 <m> Anzahl der Zeilen
 <fmn> Typ der Eisscholle
 -s : stabile
 -ix: instabile mit x Kapazität
 -h: Loch

setSnow <icefloat> <count>

Beschreibung: Legt Schnee auf der ausgewählte Eisscholle.

Optionen: <icefloat> : ID der Eisscholle
 <count> : Anzahl der Schneeschichten.

addItemToFloat <floatID> <item>

Beschreibung: Man kann Gegenstände zu einer Eisscholle geben.

Optionen: <floatID> : Nummer der Eisscholle
 <item>: Typ der Gegenstand.

addIgloo / addTent <floatID>

Beschreibung: Baut einen Igloo/ ein Zelt zur gegebenen Eisscholle.

Optionen: <floatID> : Nummer der Eisscholle

addCharacter <type> <name> <position> <temp>

Beschreibung: Addiert ein Charakter zum Spiel.

Optionen: <type>: Typ der Charakter (Forscher, Eskimo)
 <name>: Name der Charakter
 <position>: Nummer der Eisscholle, auf der der Charakter steht.
 <temp>: Temperature der Charakter

addItemToCharacter <item> <character>

Beschreibung: Addiert ein Gegenstand zum Charakter.

Optionen: <item>: Typ des Gegenstandes
<character>: Name der Charakter

setPolarBear <floatID> <randomize>

Beschreibung: Addiert ein Eisbär zum Spiel.

Optionen: <floatID>: Position des Bären.
<randomize>: Falls ja, der Controller steuert das Bär.

setSnowStorm <randomize>

Beschreibung: Sagt ob der Schneesturm bei Uns oder bei dem Controller gesteuert wird.

Optionen: <randomize>: Falls ja, der Controller steuert den Schneesturm.

startTest

Beschreibung: Startet den Test.

Optionen: -

moveAction <character> <icefloat>

Beschreibung: Bewegt den Charakter auf dem Eisscholle.

Optionen: <character>: Name der Charakter
<icefloat>: Nummer der Eisscholle.

shovelAction <character> <item>

Beschreibung: Der Charakter schaufelt Schnee.

Optionen: <character>: Name der Charakter.
<item>: normale oder fragile Schaufel (nicht obligatorisch)

buildAction <character>

Beschreibung: Der Charakter baut eine Igloo.

Optionen: <character>: Name der Charakter.

buildTentAction <character>

Beschreibung: Der Charakter baut ein Zelt.

Optionen: <character>: Name der Charakter.

checkAction <character> <float>

Beschreibung: Der Charakter überprüft die Kapazität der Eisscholle.

Optionen: <character>: Name der Charakter.
<float>: Nummer der Eisscholle.

buildRocketAction <character>

Beschreibung: Der Charakter baut den Rakete.

Optionen: <character>: Name der Charakter.

pickupAction <character> <float>

Beschreibung: Der Charakter nimmt den Gegenstand auf der Eisscholle.

Optionen: <character>: Name der Charakter.
<item>: normale oder fragile Schaufel (nicht obligatorisch)

snowstorm <n> <float2> ... <floatn>

Beschreibung: Startet den Schneesturm. Falls keine Parameter, wird es randomisiert.

Optionen: <n>: Anzahl der Eisschollen.

<floatn>: Nummer der ausgewählte Eisscholle.

polarBearMove <direction>

Beschreibung: Das Polarbär bewegt sich. Falls keine Parameter, dann in randomisierte Richtung.

Optionen: <direction>: Richtung der Bewegung.

saveOutput <filename>

Beschreibung: Speichert alle Ausgänge der Program.

Optionen: <filename>: Name der Datei.

Syntax der XML Datei:

```
<icefield>
<config>
  <icefloats>
    <icefloat>
      <type></type>
      <capacity></capacity>
      <item></item>
      <snowcount></snowcount>
      <igloo></igloo>
      <tent></tent>
    </icefloat>
  </icefloats>
  <characters>
    <character>
      <type></type>
      <name></name>
      <position></position>
      <bodytemp></bodytemp>
      <items>
        <item type="type"/>
      </items>
    </character>
  </characters>
  <polarbear position="#position" />
</config>
<test>
  <moveaction character="name of character" icefloat="#icefloat" />
  <buildaction character="Gyula" icefloat="1" />
  .
  .
  .
  <polarmove direction="dir/random" />
  <snowstorm>
    <ifloat id="id"/>
  </snowstorm>
</test>
</icefield>
```

7.1.3 Ausgangssprache

Nach jeder Befehl wird der Zustand des Charakters und auch der Eisscholle ausgeschrieben. Falls wir den Befehl nicht laufen lassen können, dann wird eine Fehlermeldung ausgeschrieben.

Syntax:

```
<Name der Charakter>: Body temperature: <Temperature>
                        Position: <Nummer der Eisscholle>
                        Items: <Liste der Gegenstände>
<Nummer der Eisscholle>: Type : <Typ der Eisscholle>
                        Capacity: <Kapazität der Eisscholle>
                        Item: <Typ der Gegenstand in der Eisscholle>
                        Snow: <Anzahl der Schneeschichten>
Polar bear: Position: <Position des Bären>
Error: <Cause>
```

7.2 Alle detaillierten Use-case Diagramme

Name der Use-case	loadTest
Kurze Beschreibung	Die Testdatei wird eingeladen, oder falls keine Testdatei angegeben ist, wird das Spiel einfach konfiguriert.
Aktoren	Player
Drehbuch	1.A Die Testdatei wird eingeladen. 1.B Anfangskonfiguration wird durchgeführt.

Name der Use-case	generateField
Kurze Beschreibung	Das Eisfeld wird generiert.
Aktoren	Controller (Player)
Drehbuch	1.A Eisfeld wird anhand gegebener Parametern generiert. 1.B Das Eisfeld wird so generiert, dass die einzelne Schollen random Typ bekommen werden.

Name der Use-case	setSnow
Kurze Beschreibung	Tut Schnee auf die gegeben Eisscholle. (Die ID der Eisscholle soll unbedingt gegeben werden.)
Aktoren	Controller (Player)
Drehbuch	Gegebene Menge Schnee erscheint auf die gegebene Eisscholle.

Name der Use-case	addItemToFloat
Kurze Beschreibung	Addieren einen Gegenstand zu einer Eisscholle.
Aktoren	Player
Drehbuch	Anhand der gegebenen Angaben (ID der Eisscholle, Typ der Gegenstand) wird ein Gegenstand zur Eisscholle addiert.

Name der Use-case	addIgloo (addTent)
Kurze Beschreibung	Ein Iglu (oder Zelt) zu einer Eisscholle addieren.
Aktoren	Player
Drehbuch	1.A Ein Iglu erscheint an der gegebene Eisscholle. 1.B Ein Zelt erscheint an der gegebene Eisscholle.

Name der Use-case	addCharacter
Kurze Beschreibung	Ein Charakter zum Spiel addieren.
Aktoren	Player
Drehbuch	Ein Charakter zum Spiel addieren. Gegebene angaben dazu: Typ, Name, Position, Körpertemperatur.

Name der Use-case	addItemToCharacter
Kurze Beschreibung	Der Charakter nimmt ein Gegenstand auf.
Aktoren	Researcher/Eskimo
Drehbuch	Ein Gegenstand wird zum Charakter Addiert.

Name der Use-case	setPolarBear
Kurze Beschreibung	Das Eisbär kommt ins Spiel an.
Aktoren	Controller
Drehbuch	1.A Das Bär wird auf das gegebene Position gestellt. 1.B Das Bär wird auf ein randome Position gestellt.

Name der Use-case	setSnowStorm
Kurze Beschreibung	Definition der Modellierung der Schneesturm.
Aktoren	Controller/Player
Drehbuch	1.A Der Schneesturm passiert random. 1.B Der Schneesturm passiert gesteuert.

Name der Use-case	startTest
Kurze Beschreibung	Der Test starten.
Aktoren	Player
Drehbuch	Der Test wird gestartet.

Name der Use-case	moveAction
Kurze Beschreibung	Der Charakter tritt auf dem Eisscholle weiter.
Aktoren	Researcher/Eskimo
Drehbuch	Anhand gegebener Angaben (Charactername, EisscholleID) den Charakter bewegen.

Name der Use-case	shovelAction
Kurze Beschreibung	Der Character Schnee schaufelt.
Aktoren	Researcher/Eskimo
Drehbuch	1.A Ein Einheit Schnee wird weggemacht mit der gegebene Typ von Schaufel. 1.B Zwei Einheit Schnee wird weggemacht.

Name der Use-case	buildAction
Kurze Beschreibung	Ein Character Igloo baut.
Aktoren	Researcher/Eskimo
Drehbuch	Ein Igloo erscheint auf der (gegebenen) Eisscholle.

Name der Use-case	buildTentAction
Kurze Beschreibung	Ein Zelt bauen.
Aktoren	Researcher/Eskimo
Drehbuch	Ein Zelt wird gebaut.

Name der Use-case	checkAction
Kurze Beschreibung	Ein Character überprüft die Kapazität der Eisscholle.
Aktoren	Researcher/Eskimo
Drehbuch	Der Charakter bekommt die Kapazität der Eisscholle.

Name der Use-case	buildRocketAction
Kurze Beschreibung	Die Rakete wird Zusammengebaut.
Aktoren	Researcher/Eskimo
Drehbuch	Der Charakter bildet die Rakete zusammen.

Name der Use-case	pickupAction
Kurze Beschreibung	Ein Gegenstand von der Eisscholle aufnehmen.
Aktoren	Researcher/Eskimo
Drehbuch	1.A Der Charakter nimmt ein Gegenstand auf. 1.B Der Charakter nimmt einen Schaufel auf. 1.B.a Der Schaufel ist normal. 1.B.b Der schaufel ist fragile.

Name der Use-case	snowstorm
Kurze Beschreibung	Ein Schneesturm wird durchgeführt.
Aktoren	Controller
Drehbuch	1.A Eine geleitete Schneesturm wird durchgeführt. 1.B Eine randome Schneesturm wird durchgeführt.

Name der Use-case	polarBearMove
Kurze Beschreibung	Das Eisbär bewegt sich.
Aktoren	Eisbär (Controller)
Drehbuch	1.A Das Eisbär bewegt sich in die gegebene Richtung. 1.B Das Eisbär bewegt sich random.

Name der Use-case	saveOutput
Kurze Beschreibung	Alle Ausgänge des Programms speichern.
Aktoren	Controller
Drehbuch	Alle Ausgänge des Programms wird in die gegebene File gespeichert.

7.3 Plan von Testen

Die Eingangssprache des Programms wurde in dem Punkt 7.1.2, und die Ausgangssprache des Programms wurde in dem Punkt 7.1.3 definiert. Durch diese zwei Sprachen wird das Testen realisiert. Wir werden einige spezifische Eingaben definieren, und zu jeder solchen Eingabe werden wir die Ausgabe kontrollieren, ob es den entsprechenden Inhalt hat.

Name des Testfalles	Player steps on stable Ice-float
Kurze Beschreibung	Die erste Test der Bewegungsarbeit von einem Spieler ist durch diesen Test realisiert. Der Fall wird getestet, in dem ein Spieler auf eine stabile Eisscholle zu treten probiert.
Ziel des Testens	Die Bewegung soll erfolgreich sein.

Name des Testfalles	Player steps on unstable Ice-float
Kurze Beschreibung	Diese Test realisiert die Bewegungsarbeit eines Spielers, wenn er auf eine instabile Eisscholle probiert zu treten.
Ziel des Testens	In diesem Test wird beide Fälle simuliert. Falls die Eisscholle umkippt, soll das Spiel beendet werden, andernfalls soll die Bewegung erfolgreich sein.

Name des Testfalles	Player steps on Hole
Kurze Beschreibung	Die dritte Art der Bewegungsarbeit soll durch diesen Test passieren. Diese Test simuliert die Ereignis wenn ein Spieler in einem Loch tritt.
Ziel des Testens	Drei Sachen können passieren, alle drei werden getestet. Erste ist das der Spieler ertrunken hat, in diesem Fall soll der Spiel beendet werden. Zweitens kommt die Möglichkeit, wenn der Spieler ein Taucheranzug hat. Am letzten kommt der Fall, wenn ihn ein anderer Spieler mit einem Seil rettet. In dem zweiten und dritten Fall soll der Spieler überleben.

Name des Testfalles	Player picks up food
Kurze Beschreibung	Der Spieler hebt ein Lebensmittel auf.
Ziel des Testens	Der Temperatur des Spielers soll mit Eins inkrementiert werden.

Name des Testfalles	Player picks up item other than food.
Kurze Beschreibung	Der Spieler hebt ein Gegenstand außer Lebensmittel auf.
Ziel des Testens	Der Gegenstand soll in dem Lager des Spielers gehen.

Name des Testfalles	Player shovels snow.
Kurze Beschreibung	Der Spieler entfernt Schnee von Eisscholle.
Ziel des Testens	Beide Fällen sollen getestet werden. Falls der Spieler ein Schaufel hat, dann soll er die Menge der Schnee mit Zwei dekrementieren, andernfalls soll sie mit eins dekrementiert werden.

Name des Testfalles	Player builds rocket
Kurze Beschreibung	Der Spieler basteln die Melder-rakete zusammen.
Ziel des Testens	Der Spiel soll gewonnen sein.

Name des Testfalles	Researcher examines Ice-float
Kurze Beschreibung	Der Polarforscher sieht an wie viele Personen die Eisscholle fördern kann.
Ziel des Testens	Mit der gesuchten Anzahl soll der Polarforscher informiert werden.

Name des Testfalles	Eskimo builds Igloo
Kurze Beschreibung	Ein Eskimo baut ein Iglu auf.
Ziel des Testens	Diese Arbeit soll erfolgreich sein.

Name des Testfalles	Snowstorm appears
Kurze Beschreibung	Ein Schneesturm taucht auf.
Ziel des Testens	Die Funktionen des Schneesturmes soll getestet werden. Diese Funktionen beinhalten: Schnee legen, Iglu zerstören, Temperatur niedrigen.

Name des Testfalles	Polar Bear moves
Kurze Beschreibung	Die Bewegung des Eisbären soll durch diesem Test simuliert werden.
Ziel des Testens	Die Bewegung soll erfolgreich sein. Die Funktionalität, wenn der Bär einen Spieler isst, soll auch getestet werden.

Name des Testfalles	Breakable Shovel breaks
Kurze Beschreibung	Ein zerbrechliche Schaufel wird zu viel benutzt, und dadurch zerstört.
Ziel des Testens	Der Schaufel soll zerstört werden.

Name des Testfalles	Player builds tent
Kurze Beschreibung	Ein Spieler, der schon ein Zelt aufgenommen hat, baut ein Zelt auf.
Ziel des Testens	Ein Zelt soll auf die Eisscholle gelegt werden, auf der der Spieler sich befindet.

7.4 Die Spezifikation von den Testen fördernde Hilfs- und Übersetzungsprogrammen

Ein Hilfsprogramm fürs Testen ist nicht nötig, da unser Prototyp es auch durch das Befehl “*loadTest*” ermöglicht. Man lässt nur ein Datei im Parameter übergeben und anhand dieses Dateies werden die spezifizierten Befehle ausgegeben und das Ergebnis wird mit der erwünschten Ausgabe verglichen. Der Erfolg des Testes kann in der Standardausgabe geprüft werden.